Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN
MECCANICA E SCIENZE AVANZATE DELL'INGEGNERIA

Ciclo 35

Settore Concorsuale: 09/C2 - FISICA TECNICA E INGEGNERIA NUCLEARE

Settore Scientifico Disciplinare: ING-IND/18 - FISICA DEI REATTORI NUCLEARI

DEVELOPMENT OF JADE, A NEW SOFTWARE FOR THE VERIFICATION AND VALIDATION OF NUCLEAR DATA LIBRARIES

Presentata da: Davide Laghi

Coordinatore Dottorato:

Lorenzo Donati

Supervisore:

Marco Sumini (UNIBO)

Co-Supervisore:

Stefano La Rovere (NIER
Ingegneria S.p.A.)

Co-Supervisore:

Marco Fabbri (F4E)

Esame finale anno 2023

Acknowledgements

I would like to express my deepest gratitude to professor Marco Sumini for his indispensable guidance and its strong passion. I'm also deeply indebted to Marco Fabbri, who has been the coordinator for the JADE project, a constant sponsor and a friend. Additionally, this endeavor would not have been possible without Stefano La Rovere, Benedetta Baldisserri and NIER as a whole. They sponsored this Ph.D. and helped me finding a (challenging) balance between work and research.

Special thanks to F4E and in particular to Raul Pampin and Alfredo Portone which have always supported this project and shared with me their knowledge and expertise. Similar gratitude is owed to my fellow Ph.D., Lorenzo Isolan, who shared with me part of this journey. Thanks should also go to the FENDL organization which believed in JADE and gave it a chance to prove its worth and to UNED for sharing their code with me, allowing an important part of JADE to be developed.

I would be remiss in not mentioning my family for the unconditional support they provided me all these years. Last but not least, I would like to thank Emma for putting up with me every day and for being my safe harbour.

Abstract

Le sezioni d'urto nucleari sono alla base della simulazione del trasporto di particelle e radiazioni. Poichè la catena produzione di librerie di dati nucleari è estremamente complessa e si compone di diversi passaggi, è opportuno prevedere stringenti procedure di verifica e validazione. Il lavoro di tesi qui presentato si è concentrato sullo sviluppo in python di un nuovo software, JADE, il cui obbiettivo è quello di aiutare significativamente nella automatizzazione e standardizzazione di queste procedure in modo tale da diminuire il tempo necessario per il rilascio di nuove versioni di queste librerie e allo stesso tempo aumentarne la qualità.

Dopo un' introduzione alla fusione nucleare (ambito sul quale si è concentrata per il momento l'attività di V&V) e alla simulazione del trasporto di particelle e radiazioni, vengono discusse le motivazioni che hanno portato allo sviluppo di JADE. Successivamente, sono descritti l'architettura generale del codice ed i benchmark, sia sperimentali che computazionali, che sono stati implementati. Inoltre, vengono riportati i risultati ottenuti dalle principali applicazioni di JADE effettuate durante gli anni di ricerca. Infine, dopo un ultima discussione conclusiva sugli obbiettivi raggiunti da JADE, vengono esaminati i possibili sviluppi a breve, medio e lungo periodo per il progetto.

Abstract

Nuclear cross sections are the pillars onto which the transport simulation of particles and radiations is built on. Since the nuclear data libraries production chain is extremely complex and made of different steps, it is mandatory to foresee stringent verification and validation procedures to be applied to it. The work here presented has been focused on the development of a new python based software called JADE, whose objective is to give a significant help in increasing the level of automation and standardization of these procedures in order to reduce the time passing between new libraries releases and, at the same time, increasing their quality.

After an introduction to nuclear fusion (which is the field where the majority of the V&V action was concentrated for the time being) and to the simulation of particles and radiations transport, the motivations leading to JADE development are discussed. Subsequently, the code general architecture and the implemented benchmarks (both experimental and computational) are described. After that, the results coming from the major application of JADE during the research years are presented. At last, after a final discussion on the objective reached by JADE, the possible brief, mid and long time developments for the project are discussed.

Executive Summary

The subject of this dissertation is JADE, a novel python-based and open source software which has been developed to help with the automation and standardization of nuclear data libraries Verification and Validation (V&V). The presented work was conducted in the framework of an industrial Ph.D. shared between University of Bologna and NIER ingegneria. A close collaboration was also established with the neutronics team of Fusion For Energy (F4E), the domestic agency that is responsible for the European contribute to the ITER project. JADE source code can be found at [1], while a detailed documentation is hosted at [2].

At first, some basic concepts of particle transport theory and isotope activations are introduced. Then, Monte Carlo (MC) simulations are presented, together with the two MC codes that JADE can take advantage of, which are MCNP and its modified version D1S-UNED. It is explained how cross sections and collections of cross sections (nuclear data libraries) are fundamental in the transport field and how, due to their complex production chain, they require extensive V&V procedures in order to ensure reliable simulation results. The advantages to automate and standardize such procedures are discussed and it is illustrated how the nuclear fusion field would benefit the most from these improvements. In fact, the majority of V&V efforts in the last 70 years has been focused on nuclear fission, which, among other things, has different cross sections, energy ranges and material needs with respect to fusion. Moreover, fission-dedicated cross sections can count on the definition of integral parameters to simplify their V&V workflow, while fusion must tally a variety of quantities discretized in energy and in space to have a clear estimate of the impact that cross section modifications may have on the transport simulation results. Once the main motivations at the base of JADE development are summarized, the tool conceptual scheme and requirements are presented. JADE is able to automatically generate and run a great number of benchmarks inputs and to automatically post-process the obtained raw data in order to present them in a more meaningful and efficient way. This is achieved through the use of formatted excel files and plots. JADE architecture is described both on a high level, where the main functionalities are illustrated, and on a low level, where it is explained how these functionalities have been abstracted according to the Object Oriented Programming (OOP) paradigm used during JADE development. Finally, a brief description of all benchmarks that have been implemented in JADE' suite is provided.

After that, the focus is shifted on how the software was used during the Ph.D. period. The first presented application is a proof of concept, where JADE was tested on commonly used libraries to demonstrate its potential and to show that it was able to re-spot known libraries inconsistencies. Then, it is illustrated how JADE played a role on the different stages of FENDL v3.2 release (a real production case). Additionally, other case studies are presented, such as a comparison of the performance of the MCNP theoretical model versus the use of cross section data, some benchmarking done on the newer JEFF library release and the application of JADE to D1S-UNED activation libraries, which have been tested on Shut Down Dose Rate (SDDR) benchmarks.

To conclude, it is argued how JADE reached its goals and demonstrated to have the capability to be an important tool in the field of nuclear data libraries V&V (especially for fusion related ones). The hope is that the automation and standardization provided by the tool could lead to a reduction of effort fragmentation and speed up the release cycle of nuclear data libraries, while, at the same time, improving their quality. As a very last thing, the brief, mid and long term developments for the project are discussed.

Objectives

The main objective of the JADE project are to:

- improving the standardization of nuclear data libraries Verification & Validation (V&V);
- bring heavy automation to nuclear data libraries V&V;
- provide a focus on testing data related to nuclear fusion applications.

During this first three-year cycle of development the following goals were set (and reached):

- finalize JADE general architecture;
- implement a minimum set of both computational and experimental benchmarks;
- prove JADE utility on production applications.

Scientific production

List of international publications (related to JADE):

- Davide Laghi et al. “JADE, a new software tool for nuclear fusion data libraries verification & validation”. In: *Fusion Engineering and Design* 161 (Dec. 2020), p. 112075. DOI: `10.1016/j.fusengdes.2020.112075`
- Davide Laghi et al. “Application of JADE V&V capabilities to the new FENDL v3.2 beta release”. In: *Nuclear Fusion* 61 (2021), p. 116073
- Laghi D. et al. “Status of JADE, an open-source software for nuclear data libraries V&V”. in: *Fusion Engineering and Design* 187 (2022), p. 113380

List of international conferences and workshops contributions (related to JADE):

- Laghi D. et al. “Progress report on JADE, a new nuclear data libraries V&V test suite”. In: XIVth ITER Neutronics Meeting. Barcelona, Spain, 2020
- Laghi D. et al. “A new V&V philosophy for fusion nuclear data libraries”. In: 27th International Conference on Transport Theory - ICTT27. Bertinoro, Italy, 2022
- Laghi D. et al. “JADE, a novel V&V software for fusion nuclear data libraries”. In: XVth ITER Neutronics Meeting. Barcelona, Spain, 2022
- Laghi D. et al. “Status of JADE, an open-source software for nuclear data libraries V&V”. in: 32nd Symposium on Fusion Technology - SOFT. Dubrovnik, Croatia, 2022
- Laghi D. et al. “Status of JADE, an open-source software for nuclear data libraries V&V”. in: IAEA - Technical Meeting on the Compilation of Nuclear Data Experiments for Radiation Characterization. Vienna, Austria, 2022

List of Acronyms

Acronym	Description
ACE	A Compact ENDF
API	Application Programming Interface
CD	Continuous Deployment
CI	Continuous Integration
CPU	Central Processing Unit
DPA	Displacement Per Atom
EAF	European Activation File
ENDF	Evaluated Nuclear Data File
FENDL	Fusion Evaluated Nuclear Data Library
FNG	Frascati Neutron Generator
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HCPB	Helium Cooled Pebbled Bed
IAEA	International Atomic Energy Agency
IG	Iter Grade
ITER	International Thermonuclear Experimental Reactor
JEFF	Joint Evaluated Fission and Fusion
MC	Monte Carlo
ML	Machine Learning
OOP	Object Oriented Programming
OS	Operative System
PEP	python Enhancement Proposal
SDDR	Shut Down Dose Rate
SINBAD	Shielding Integral Benchmark Archive and Database
SS	Stainless Steel
TBM	Test Blanket Module
UNED	UNiversidad Education a Distancia
UNIBO	UNIversity of Bologna
V&V	Verification and Validation
VV	Vacuum Vessel
WCLL	Water Cooled Lithium Lead

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The work presented in this thesis has been performed in the framework of an industrial Ph.D. programme shared between the University of Bologna and NIER Ingegneria. A close collaboration was also established with the neutronics team of Fusion For Energy (F4E), the domestic agency that is responsible for the European contribute to the ITER project (see Section 1.2.3 for additional details on it).

The main goal has been the development of the JADE platform, a novel python-based and open source software that has been developed to automatize and standardize the Verification and Validation (V&V) procedures of nuclear data libraries, with a special focus on fusion applications. As it will be discussed in Section 1.1, nuclear data libraries are at the core of the particles and radiations transport simulations. In Section 1.1.3 it will be explained how, due to their production chain being particularly complex and composed by many steps, strict V&V procedures are necessary in order to obtain reliable results from simulations. This is one of the main reasons, together with the ones presented in Section 1.3, why the development of JADE was started, with its main objective being to help with the standardization and automation of such V&V procedures. As it is well known, particles and radiation transport simulations are of great importance in the development of many highly technological fields, and, among them, the nuclear fusion field (which is introduced in Section 1.2) was chosen as the focus of the benchmarks implemented so far in JADE's suite.

After this introduction, Chapter 2 will illustrate what actually JADE is, how it was developed and what is its structure. All the benchmarks that have been implemented will also be described. Then, in Chapter 3 all the major application of the code will be presented. These applications span from simple proof of concepts up to real world production cases and aim to demonstrate the value that JADE can bring to the nuclear data libraries V&V and fusion fields. Finally, Chapter 4 summarizes the work that has been done and discuss future exciting possible developments for the project.

1.1 Particles transport simulation

The first thing that is introduced in this work is what it means to simulate particles (and radiations) transport. At first, Monte Carlo transport, cross sections and nuclear data libraries are introduced, being them the core topic of this thesis. Then a standard tool, MCNP, and one of its variants devoted to activation/dose calculations, D1S-UNED, are presented. These are the Monte Carlo transport codes that JADE is currently able to interact with.

1.1.1 Monte Carlo transport

There are a few ways of simulating the transport of particles through matter. The activities described in this work are all based on Monte Carlo simulations. Monte Carlo method is a statistical approach that can be applied to many fields like stock price evolution prediction, transport phenomena, maintainability analyses and so on [11][12]. The core of the method is the Central Limit Theorem [13], a very famous probability theorem that establish that when independent and identically distributed random variables (with mean and variance defined) are summed up, their sum is a random variable itself that tends towards a normal distribution even if the original variables were not normally distributed themselves. Monte Carlo methods apply this concept as follows. Firstly, random inputs are sampled from a domain of possible ones defined accordingly to the problem under study, then, some kind of computation based on these inputs is performed and finally, the obtained results are aggregated.

When this methodology is applied to particles transport simulations [14], the sampled inputs may include for example information on where the particle is originated, its initial energy, position, etc. The computation part consists of simulating the transport of the particle in a medium where all possible interactions of interest with matter are considered. The probabilities related to specific interaction between particles and matter are also to be considered as sampled inputs and are derived from the nuclear cross sections, which are discussed in better detail in Section 1.1.2. The simulation of the transport is considered complete when either the particle exits the energy or spatial domain set up by the user or if it is absorbed/destroyed. During its journey, the initial particle may give birth to what are defined as secondary particles and their transport can be equally simulated. The simulation of one particle and, sequentially, all the secondary particles that it has generated is commonly referred to as particle history and a schematic example of it is displayed in Fig. 1.1. If a high enough number of histories is simulated (the higher the better) the Central Limit Theory can be applied and general particle flux results (or a generic reaction rate) can be inferred from the simulations. Fig. 1.2 gives an idea of the effect that increasing the number of simulated histories can have in a simple model.

1.1.2 Nuclear cross sections

In the field of nuclear analyses, cross sections occupy a fundamental role since they represent the only way to properly describe the interactions of particles with matter. A cross section is a measure of the likelihood that some particular interaction (e.g. absorption, fission, scattering, etc.) will occur after the collision of a particle with a target nucleus. This likelihood is usually measured in barns (10^{-24} cm^2) which is a surface area measure. Indeed, an analogy is made with target shooting: the bigger

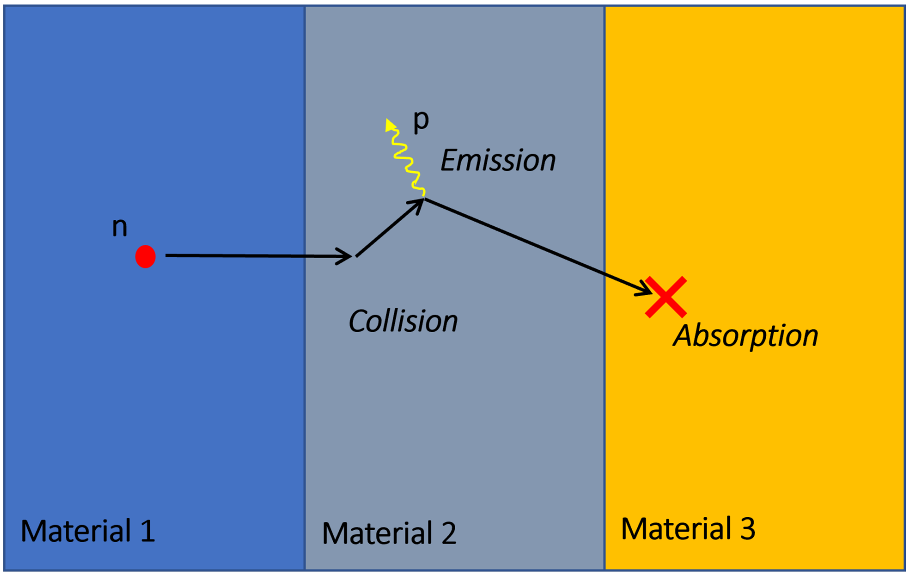


Figure 1.1: Schematic example of particle transport in a Monte Carlo simulation

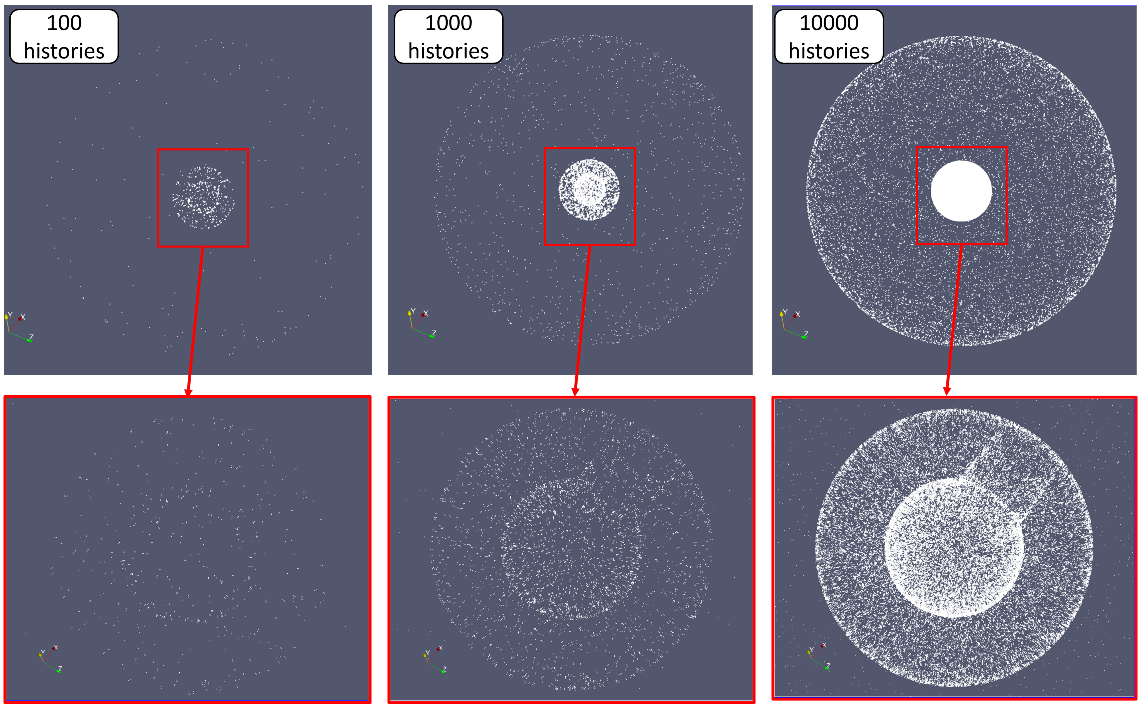


Figure 1.2: Visualization of the effects of an increase of simulated particle histories, each point represents an interaction with matter

the target area, the more likely the interaction to happen. To be more precise, this is the definition of microscopic cross section σ, but is possible to define also a macroscopic cross section $\Sigma = N\sigma$ where N is the the atomic density of the target. Σ $[m^{-1}]$ can then be interpreted as the "equivalent area" of all target particles per unit volume. This is usually done to obtain a simple expression able to link the reaction rate of certain interaction (R) $[m^{-3}s^{-1}]$ with the flux (Φ) $[m^{-2}s^{-1}]$ of the particles or radiations crossing the target:

$$R = \Sigma\Phi \tag{1.1}$$

Cross sections can depend on several factors such as:

- the type of particle under study;
- the energy or velocity of the particle;
- the direction of the particle;
- the medium into which the transport is being carried out;
- the temperature of the medium.

Additional details on the cross sections theory can be found in [15] and [16].

1.1.3 Nuclear data libraries production chain

Cross sections, as described in Section 1.1.2, are complex and continuous functions of both projectile and target cinematic conditions. Unfortunately, complete analytic models able to reproduce them have never been identified apart from really simple cases (like f.i. neutron-deuteron intraction) and their generation heavily rely on experimental data and simplified interaction models. For various technical reasons, it is impossible to measure many of the physical parameters required for nuclear data applications, which leads to the necessity to combine experiments with modelling techniques. This causes cross sections to follow a quite complex production chain which is extensively described in [17] and [18]. To summarize, the procedure starts with experimental data points that are interpreted with the help of theoretical models in order to extrapolate evaluated continuous data. A collection of evaluated nuclear cross sections for all available interactions of a set of isotopes is referred to as a *nuclear data library.* This kind of data can be stored in different formats, although the most common one is ENDF-6 [19], developed and maintained by the Cross Section Evaluation Working Group (CSEWG) at Los Alamos Laboratories. Finally, the libraries need to be post-processed in order to obtain point-wise or multi-group cross sections. This happens because computer codes cannot run efficiently using continuous data and can be way faster with discrete ones. For neutron-gamma interactions with matter, this averaging process is done using software like NJOY [20], which produces files in the so-called ACE (A Compact ENDF) format for each isotope. These files are ultimately adjusted to serve as inputs for standard Monte Carlo transport simulation codes. Fig. 1.3 shows a simplified scheme of the nuclear data libraries production chain.

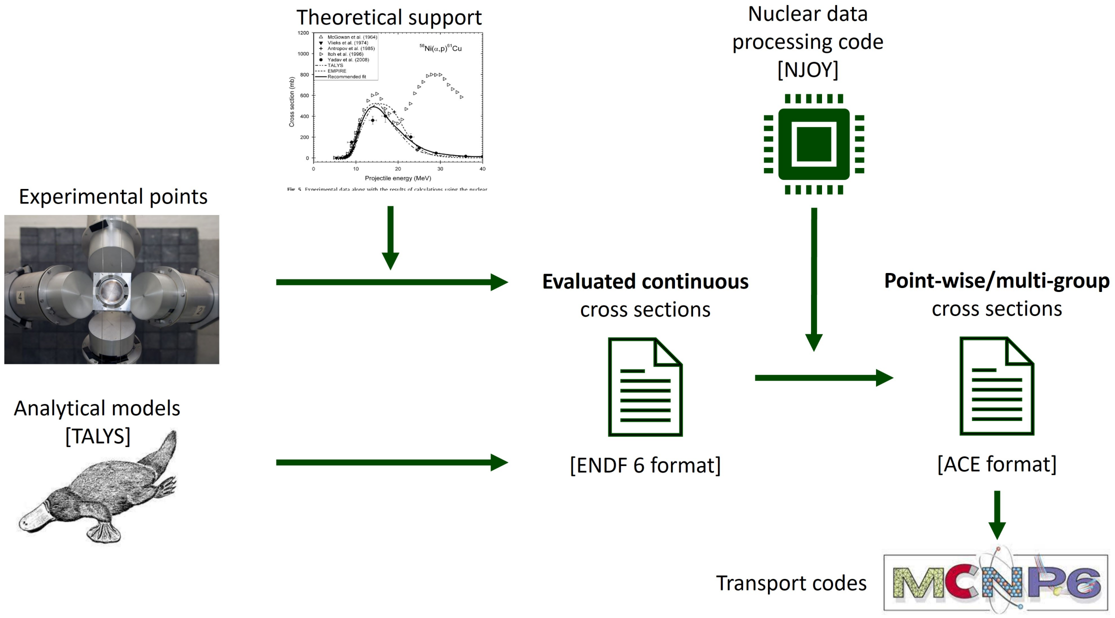


Figure 1.3: Production chain scheme for a single isotope nuclear cross section file

1.1.4 MCNP

Many Monte Carlo transport codes are available, but one of the most widely used is MCNP [21], which is also one of the reference transport codes employed in the ITER project. MCNP is an originally FORTRAN based code which has been developed and is currently maintained and moved towards new programming standards (like C++, etc.) by the Los Alamos Laboratories in New Mexico, USA.

An exhaustive description of MCNP functioning would be redundant, as all information can be read in its manual [21].Nevertheless, it is worth to discuss a few concepts related to the use of MCNP (or Monte Carlo codes in general) which are important to understand the work here presented.

Brief introduction to Constructive Solid Geometry

In Monte Carlo codes, the geometrical domain into which the transport takes place is defined through the use of Constructive Solid Geometry [22], also known with the acronym CSG. As it can be visualized in fig Fig. 1.4, Constructive Solid Geometry means that every complex shape has to be built starting from primitive solids like cubes, spheres, cones, etc. that are combined using boolean operators. The reason for this is that when the code is simulating a particle crossing a surface, it needs to compute the normal to that surface using its defining mathematical equation.

In an MCNP input, everything is defined through the use of *cards*. In order to build the geometrical model, a number of surface cards need to be defined such as planes, spheres or cylinders. Then, these surfaces can be combined in a boolean way in order to create the so-called *cells* through the use of specific cards. A cell is an elemental volume in an MCNP model and a single material can be assigned to it. A set of cells constitutes the medium where the particle transport is effectuated.

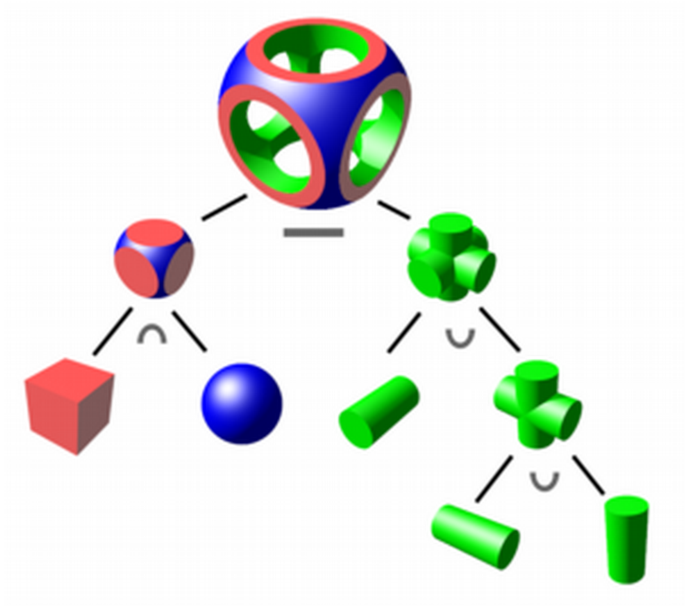


Figure 1.4: Constructive Solid Geometry (CSG) visualization

Brief introduction to tally definition

Another important type of card in MCNP is the one which defines a *tally*, which simply consists in an output that the user is able to request to MCNP scoring some relevant quantities. Common tallies may be:

F1 which tallies the current of a specified particle through a surface;

F2 which tallies the flux of a specified particle on a surface;

F4 which tallies the flux of a specified particle over a cell;

F6 which tallies the heating generated by a specified particle over a cell;

F7 which tallies the fission energy deposition over a cell;

F8 which tallies the energy distribution of pulses created in a detector.

Another thing to point out is that, apart from the explicit F6 tally, there is an alternative way to measure the heating generated in a cell. Indeed, the resulting flux coming from an F4 tally can be multiplied by the heating number (i.e., KERMA, Kinetic Energy Release in Material [23]) using the *multiply card* (FM), where the heating number estimates the energy deposited per unit track length by a particle in a specific material. The two heating computation methods should yield exactly the same result.

Brief introduction to material cards

Material cards are used to specify the isotopic composition of a material in MCNP which can then be associated to one or more cells. For each isotope the following quantities are specified:

- the *zaid number*, which uniquely identifies the isotope (e.g. 92235 for U_{92}^{235});
- the *nuclear data library* to be used, which is specified using its suffix tag (e.g. 31c);
- the *fraction* of that isotope in the material; if positive it is interpreted as an atom fraction, if negative as a mass fraction.

As an example, the string "1001.31c 2   8016.31c 1" defines water where hydrogen isotopes have an atom fraction equal to 66% and oxygen 33% (the fractions are automatically normalized to 1). For both isotopes, the FENDL-3.1 nuclear data library will be used (library tags can be customized by the user).

Brief Introduction to the STOP card

There are different ways to specify when a simulation should stop. The STOP card allows to specify one or more of this limits simultaneously, that is:

- total computer time (i.e. the sum of computer time used by all CPUs);
- number of simulated particle histories;
- statistical precision reached on a specific tally.

1.1.5 D1S-UNED

The estimation of the shut-down dose rate (SDDR) induced by neutron activation is a major safety task for fusion reactors [24]. Indeed, all kind of intervention and maintenance operations must be planned in such a way to guarantee that the dose limits to which the operators and the public can be exposed are not exceeded.

The neutron activation process

The neutron activation process, better described in [25], consists of a nucleus capturing an incoming neutron. This causes the nucleus to become heavier and to enter in an excited state. The excited nucleus can then decay emitting either:

- photons, often referred to as "secondary photons";
- beta particles (i.e., either electrons or positrons);
- alpha particles (i.e., helium nuclei).

For some isotopes more than one neutron capture is needed to make them unstable. Each radioactive isotope will decay according to specific decay reaction(s) and a key parameter that characterize their decay is the half-life time, which indicates what is the time needed for half of the initial quantity of radioactive material to decay. The decay time is defined in this way because decay process is not a deterministic phenomenon: there is only a measurable probability that a specific radioactive isotope will decay in a certain time interval. When a great number of radioactive isotopes is considered though, the half life time can be considered constant and is independent of the initial amount of isotopes.

Computing the material activation is the first step in calculating the SDDR as it will be better discussed in the following paragraph.

D1S Vs R2S

The Direct One Step (D1S) and the Rigorous Two Steps (R2S) methods are two of the most used techniques to compute SDDR [26].

R2S is actually composed by three different steps:

1. a spatial map of the neutron flux of the reactor is computed with the help of a transport code (e.g. MCNP);

2. the neutron flux activates the materials and the resulting gamma decay source can be computed through activation codes like FISPACT [27];

3. a new transport simulation is performed using the decay photons source in order to tally the SDDR at specific locations.

On the contrary, D1S allows to evaluate SDDR with a single transport calculation. To achieve this, the generation of photons by the neutrons (i.e. prompt photons) is deactivated and substituted with the "decay" photons ones. This is done using ad-hoc built libraries where the photon production cross section do not represent anymore the probability that an interaction of a neutron with the specific isotope will lead to the production of a (prompt) photon, but, instead, with the cumulative probability that the isotope will get transmuted into a radioactive one (activation) which will decay producing a (decay) photon. This implies that the temporal dimension is now part of the equation since the probability of decay depends on the unstable isotope half-life constant and for how much time the original isotope was irradiated by neutrons. In order to account for that, some factors need to be computed that adjust this probability based on the irradiation history and on the cool-down time, i.e., the time interval after the end of the irradiation campaign at which SDDR should be computed. These factors are usually computed analytically or through activation codes.

A more detailed discussion on the differences between the two methods, their requirements and history can be found in [28]. Even if both techniques present advantages and disadvantages, the higher speed and the direct coupling between the decay gammas and the neutron flux have led D1S to be currently the preferred method for SDDR calculations for the ITER project. More specifically, D1S-UNED [28] appears to have established itself as the primary tool for this kind of computations. D1S-UNED consists of a modification of the MCNP source code that, among many other additional features, allows evaluating SDDR values with the D1S method using ad-hoc generated nuclear data libraries.

The following paragraphs give a high level description of two additional files that are necessary (together with a classic MCNP input) in order to run a transport simulation with the D1S-UNED code.

Reactions file

D1S method needs as input data the list of reactions that produce radioactive nuclides. The information of all reactions to be considered in the calculation is given in a ‘reaction list’ file. For each reaction the following is defined:

- *Parent nuclide*, representing the stable isotope to be activated;
- *MT*, integer defining the reaction type according to the ENDF definition;
- *Daughter nuclide*, representing the isotope resulting from the reaction

During the D1S simulation only those photons produced by a reaction included in the list are forced to be transported using the "PIKMT" MCNP card. Therefore, no other photons are produced by any other reactions (parent nuclide or MT) than those listed in "reaction list" file. For a correct result of the D1S simulation, parent nuclides listed in reaction list file should have a specific modified (decay library) neutron transport library associated with it.

Irradiation file

In D1S methodology the decay photon production rate (induced by nuclear reaction) is related to the reaction rate through the time correction factor. This time factor, which depends on the radioactive nuclide and irradiation scenario, has to be provided for each isotope as input data. In order to provide photon flux at different cooling times in the same run, the disintegration constant of each isotope has to be provided also in the irradiation file. If several irradiation schedules are defined for different components (cells), all time correction factors, corresponding to each irradiation scenario, should be indicated in this file.

1.2 Nuclear fusion and ITER

Nuclear fusion is the process that naturally occurs in all stars where, thanks to the massive gravity forces involved, smaller nuclei are fused together in order to produce heavier elements. During such process, a part of the nuclei mass is converted into energy, ultimately allowing for the birth and evolution of life as we know it. Fusion is an important topic for JADE, since, as it will better discussed later on, the focus of its V&V has been directed exactly on this field of application. An overview of what is a nuclear fusion reaction is given and the main fusion reactor technologies are discussed. A specific sub-section is dedicated to the ITER project since it has had a significant influence in JADE benchmarks definition.

1.2.1 A brief introduction to nuclear fusion reactions

As anticipated, nuclear fusion [29] is a reaction involving two or more nuclei which combine together to form one or more heavier atomic nuclei with a concurrent release of subatomic particles such as neutrons or protons. This process is quite different from nuclear fission [30], where, instead, heavier nuclei split in lighter ones. Both processes release significant quantities of energy which comes from the difference in mass between the reactants and products. This difference in mass can be associated with the difference in binding energy between neutrons and protons composing the reactants nuclei and the ones composing the products. This energy can be calculated subtracting the mass of the nucleus from the sum of all the individual masses of the nucleons that form it (which always differ). The higher the binding energy is, the more stable is the nucleus, and if the nucleons rearrange from a lower binding energy configuration to a higher one, they will release an amount of energy equal to the difference in binding energy. Fig. 1.5 displays the binding energy of nuclei versus the number of nucleons which leads to two main observations. The first is that Fe-56 (or Ni-62) can be individuated as the most stable elements in nature which means that all lighter isotopes will tend to be naturally fusible, while heavier nuclei will be naturally fissionable. The second observation is that the amount of energy that can be released from hydrogen fusion (per isotope) is significantly higher then the one coming from uranium fission.

Nuclear fusion is the reaction that constantly takes place in stars. The energy generated by this reaction is what allows them to avoid to collapse under the gravity force exerted from their immense masses and for life as we know it to exists. Many different fusion reactions can occur in stars but, among them, it is worth to better discuss the hydrogen cycle [31] since it consists of the reactions that have been most studied in an attempt to reproduce them on earth. Everything starts with the **H-H** reaction where two protons fuse together to generate deuterium:

$$^{1}\mathrm{H} + {}^{1}\mathrm{H} \rightarrow {}^{2}\mathrm{H} + \beta^{+} + \nu$$

The by-products are a positron (which will at some point be annihilated by an electron) and a neutrino generating a total energy of 1.44 MeV. When some deuterium has been generated in the star, even if **D-D** reactions are technically possible, the vast amount of hydrogen available will lead more often to **H-D** reactions:

$$^{1}\mathrm{H} + {}^{2}\mathrm{H} \rightarrow {}^{3}\mathrm{He} + \gamma$$

where the majority of 5.49 MeV produced will be carried by the produced gamma (kinetic energy). Finally, the helium-3 is burned:

$$^{3}\mathrm{He} + {}^{3}\mathrm{He} \rightarrow {}^{4}\mathrm{He} + 2{}^{1}\mathrm{H}$$

liberating two hydrogen nuclei and 12.86 MeV.

Obtaining nuclear fusion though is not a simple task. First of all, the hydrogen gas needs to be heated up to plasma, which means that the particles kinetic energy is so high that the bond between electrons and the nuclei has been broken. At this point, two “naked" nuclei will feel a substantial electrostatic repelling force since they are positively charged. A significant force (or energy) is necessary to bring the nuclei close enough in order for the strong nuclear force to ramp up and activate the fusion reaction. In stars, this is achieved with a combination of extremely high temperature and density. On earth, different solutions are under study as it will be investigated in Section 1.2.2. For this reason, scientists individuated two different hydrogen fusion reactions that could give better results which are the **D-D** and **D-T**. Thanks to the additional neutrons, it is much easier to overcome the Coulomb

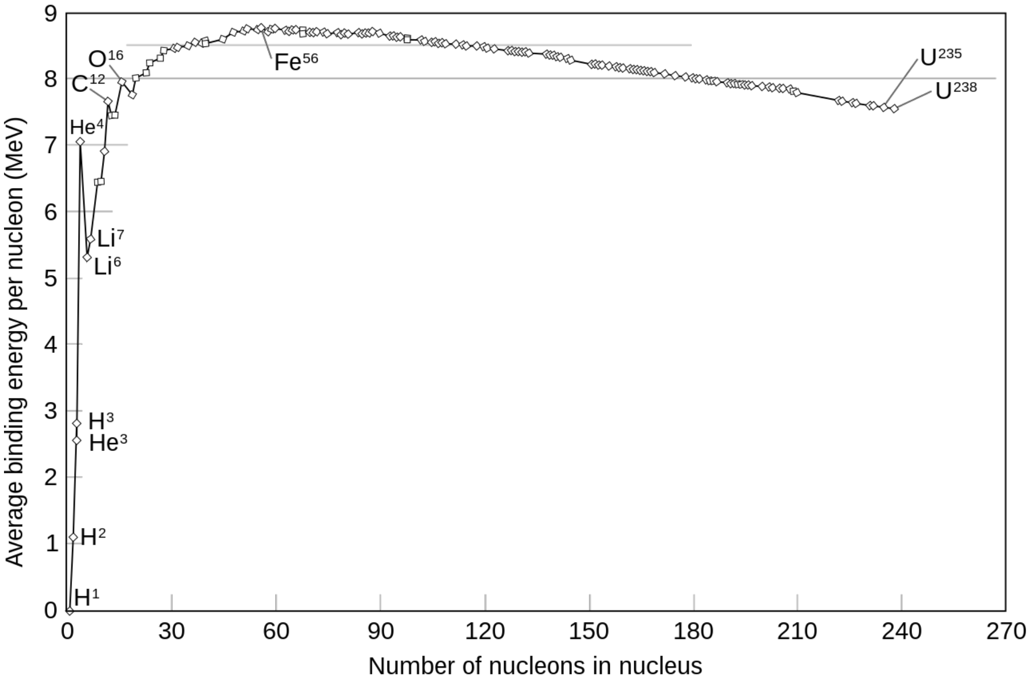


Figure 1.5: The nuclear binding energy curve.

barrier of deuterium or tritium isotopes with respect to the hydrogen one (which is a single proton). Additionally, the likelihood of these reactions to happen is higher (trivially, their size is bigger and it is easier for the nuclei to collide) and they release more energy:

$$\textbf{D-D:}\ {}^{2}\text{H} + {}^{2}\text{H} \rightarrow ({}^{3}\text{He} + 1\ \text{MeV}) + (p + 3.03\ \text{MeV})$$

$$\textbf{D-D:}\ {}^{2}\text{H} + {}^{2}\text{H} \rightarrow ({}^{3}\text{He} + 0.82\ \text{MeV}) + (n + 2.45\ \text{MeV})$$

$$\textbf{D-T:}\ {}^{2}\text{H} + {}^{3}\text{H} \rightarrow ({}^{4}\text{He} + 3.5\ \text{MeV}) + (n + 14.1\ \text{MeV})$$

In particular, the **D-T** reaction is considered to be the most promising one due to its high energy yield, lower Coulomb barrier and higher rate of interactions between deuterium and tritium (if compared with hydrogen) [32]. It should be noted, though, that hydrogen and deuterium are readily available in nature, while tritium is radioactive and naturally decays with a half-life time of 12.32 years. This means that, to use tritium, nuclear fusion reactors will also need to be able to breed it in some way.

1.2.2 Overview on nuclear fusion reactors technology

Around the 1950s scientists started to look for ways to replicate fusion on earth. The following is a brief description of some of the main concept that were individuated through the years.

TOKAMAK

One of the oldest (and more explored) concepts in the fusion field is certainly the TOKAMAK [33], whose schematic design is illustrated in Fig. 1.6. These machines

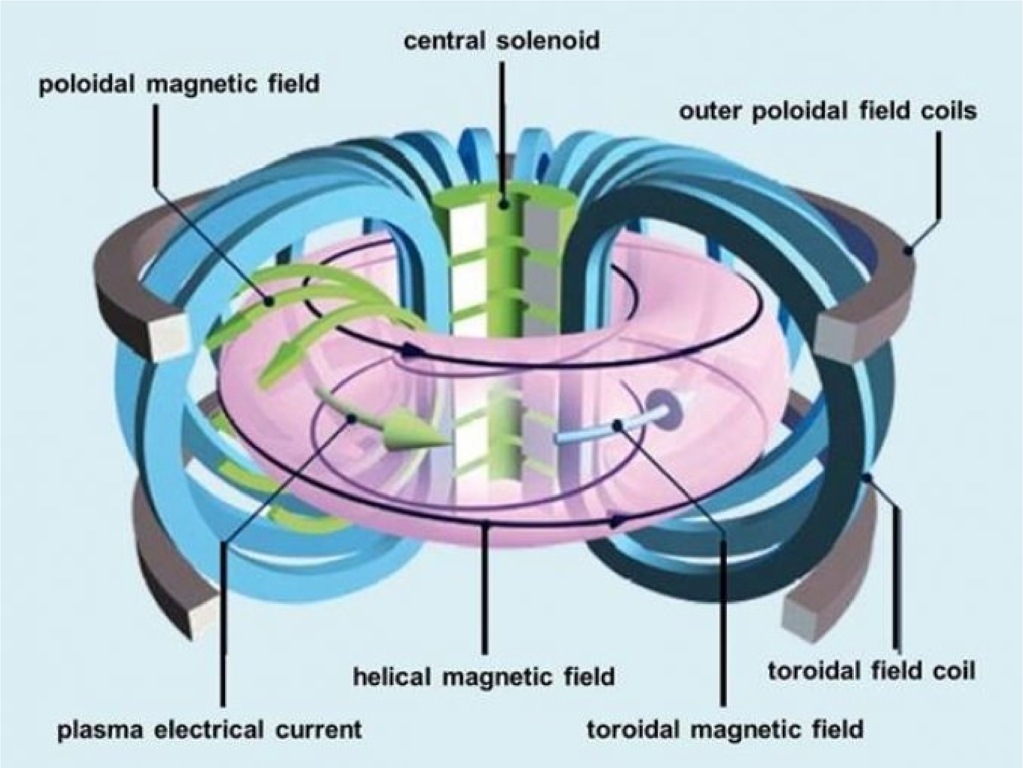


Figure 1.6: A schematic example of TOKAMAK design

consist of an array of D-shaped super-conductive magnets arranged around a torus vacuum chamber. The extremely strong electro-magnetic field generated by them and by other solenoidal and poloidal coils has the objective to confine the plasma generated in a vessel were vacuum is previously obtained. The plasma is then heated up until it reaches the temperature and the density needed to initiate a fusion chain reaction, according to the Lawson criterion [34]. The biggest TOKAMAK currently active is the JET (Joint European Torus) [35] located in Culham, UK, where experimental campaigns started in 1983. JET was instrumental to the design of ITER and DEMO which will be both better described in the following section.

Stellarator

Building on the TOKAMAK idea, only one year later, the Stellarator design was conceptualized [36] in order to solve some of the issues afflicting the TOKAMAK design. Thanks to its magnets twisted shape (see Fig. 1.7), it does not require plasma current to work, allowing for a better resistance to instabilities and, most notably, for steady-state operation. In spite of its intrinsic advantages, the majority of the efforts related to magnetic confinement fusion have been historically directed towards the TOKAMAK solution, considered simpler to build and generally more efficient. The most important example of Stellarator is the Weldenstein 7-X [37] located in Greifswald, Germany.

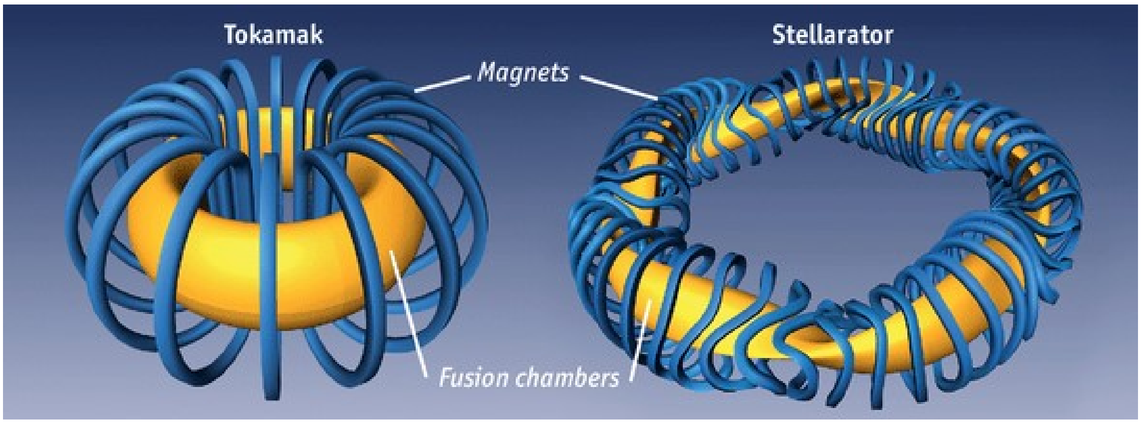


Figure 1.7: Comparison between TOKAMAK and STELLARATOR magnets shape

Inertial confinement fusion

A radically different approach with respect to magnetic confinement is what is called Inertial Fusion Confinement or IFC [38]. Here, to compress and heat the fuel, powerful lasers deposit energy on the outer layer of a target which explodes producing a compressive force against the remainder of the target. This process creates shock waves that travel inward through the target and, if they are powerful enough, the compression and heat reached at the center are high enough for fusion to occur. The largest active IFC reactor is the National Ignition Facility (NIF) [39] located in Lawrence Livermore, California.

Magnetized target fusion

Magnetized target fusion [40] builds on the IFC concept but with significant differences. Fusion is still reached thanks to compression, but, instead of lasers, off-the-shelf steam pistons are used. Inside the fusion chamber, a liquid metal wall can be found which has the double function of protecting the outer solid wall from damage and also of being the energy vector. The metal is heated up and goes through a heat exchanger that feeds a conventional water-steam cycle to produce electricity. This spinning liquid metal wall is shaped into a sphere by the steam pistons. Some fusion fuel is injected in the chamber and the wall is compressed until a pulse of fusion is reached. In this way, no targets need to be manufactured. This design concept has been proposed and currently developed by General Fusion inc. [41] which will build its demonstration plant at the UKAEA Culham Campus.

1.2.3 The ITER Project

As anticipated before, ITER (International Thermonuclear Experimental Reactor, but also *the way* in latin) is arguably the most important project in the field of experimental fusion reactors [32]. When completed, it will be the biggest and most advanced prototype of a TOKAMAK and it is currently under construction in Cadarache, France. The partners participating to the ITER project are Europe, USA, India, South Korea, Japan and Russia, each one represented by its domestic agency. F4E, in particular, is responsible for the biggest contribution to the project (around 50%) because Europe will be the party physically hosting the reactor. Parties contributions are provided in cash, in services and reactor components. To achieve this, preparation and coordination of the design, research and development (R&D) and the fabrication of very high-technology components that are required to construct ITER is done both internally to the various domestic agencies and involving skilful external companies through public tenders.

ITER aim is to achieve Q = 10, that is, to be able to produce 10 times the energy used for its functioning. However, since ITER will still be considered a prototype, the reactor will not include any electricity generator group and will not actively breed the tritium needed for the fusion reactions. The solution of these two open issues has been remanded to DEMO, an even bigger TOKAMAK that will be the first real prototype of a commercially viable power plant. DEMO objective is to reach Q = 45 and it is envisioned to start operations somewhere in the 2050s. The main objectives of ITER are:

- produce 500 MW of thermal fusion power with Q=10, where, currently, the record is hold by the JET experiment in 1997 with Q=0.62 at 16 MW [42];
- demonstrate the integrated operation of various fusion power plant technologies such as heating, control, diagnostic, cryogenics and remote handling;
- achieve self-sustained D-T fusion (i.e., through internal heating);
- test tritium breeding concepts to help DEMO design;
- demonstrate the safety characteristics of a fusion device.

1.3 Motivation for JADE development

The aim of this section is to present the main motivations that led to the development of JADE, which is the subject of the work described in this dissertation. At first, the state of the art of nuclear data libraries V&V is presented, from which emerges the concept of fission bias. After a better description of such bias, the FENDL project is introduced and it is explained how JADE aims to help solving some of the issues individuated by the FENDL consultants.

It should be clarified that when discussing V&V of nuclear data libraries through Monte Carlo calculation, the algorithms, statistical estimator, transport methodologies should be kept fixed. That is, the transport simulation method is not under investigation, but only the nuclear data is.

1.3.1 Nuclear data libraries V&V

Verification and validation [43] are two independent procedures that aim to check that a model, a service, a product or a system meets a series of requirements and specifications that allow it to fulfil its intended purpose. In the nuclear data libraries context, verification is the process that checks that the libraries are produced in the right way (quality assessment) and that the results coming from transport simulations are coherent with the physical modelling they try to implement (computational benchmarks). Validation, instead, consists in checking that the simulation results actually reflect the real world physics and it is usually achieved through the definition of experimental benchmarks. At this point, it is worth to clarify that in this dissertation the term "computational benchmark" will be used in a broader sense with respect to its strictly grammatical definition. In fact, a computational benchmark, should only consist of comparing some calculations with a "bench", like an analytical solution of a transport problem. Here, the definition will be extended (accordingly to what is often done in the nuclear fusion field) to comparisons of calculations where different codes or different nuclear data libraries are used on the same computational problem. In this case the "bench" is one of these results, to which all other are compared to, meaning that it can change depending on the comparisons that are performed.

As showed in section Section 1.1.3, the production chain of a nuclear data library is quite long and complex, thus, rigorous V&V procedures and quality assessment must be applied at every production stage and should envisage at least:

- the use of systematic quality and consistency checks to find faults in the inputs reference files/channels or other general errors in the library production;
- visual data inspection and comparison;
- verification against computational benchmarks;
- validation against integral experimental benchmarks, as the ones available in the SINBAD database [44].

In a single library, there are usually hundreds or even a thousand of isotopes and for each one of them there are hundreds of cross sections referred to into the ACE file [45] (Fig. 1.8). This means that the number of cross sections to be tested are in the order of 10^5 for a single library, and this number does not take into account the fact that each cross section may be dependent on the particle energy or angle of incidence. As it will be discussed in the following paragraphs, a good portion of V&V is currently carried out manually, with a focus on a few key benchmarks in order to evaluate the impact of changes made in the library. It is clear though that heavy automation is needed if an extensive V&V procedure is to be applied in order to verify as many cross sections as possible. Moreover, automation would also be a great opportunity for standardization, since tests would be always conducted in the same way on the same set of selected benchmarks.

As far as the Quality Assurance (QA) and covariance is concerned, a number of utility tools have been developed by the nuclear community, each one of them with their specific perks and limitations. The most widely used are arguably NJOY [20], maintained by the Los Alamos Laboratories, and PREPRO [46], maintained by the IAEA nuclear data section. NJOY is one of the major tool for the generation of point-wise or multi-group cross sections starting from the evaluated ones and, for this reason, includes a series of tools to verify the quality of the libraries it

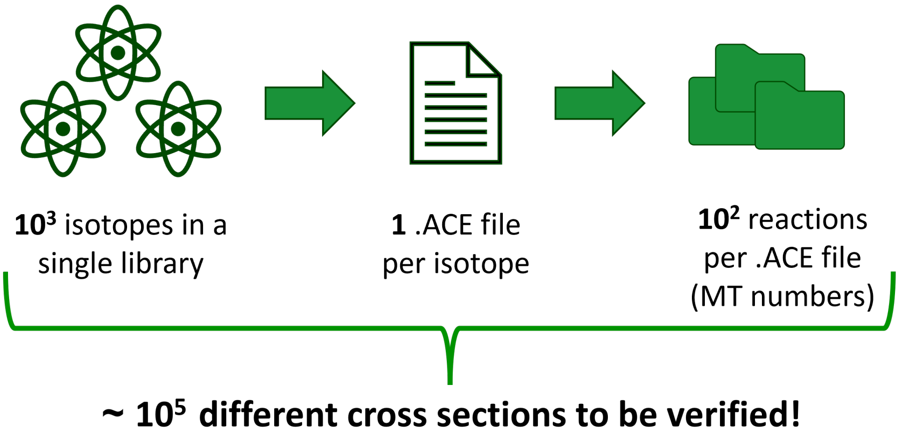
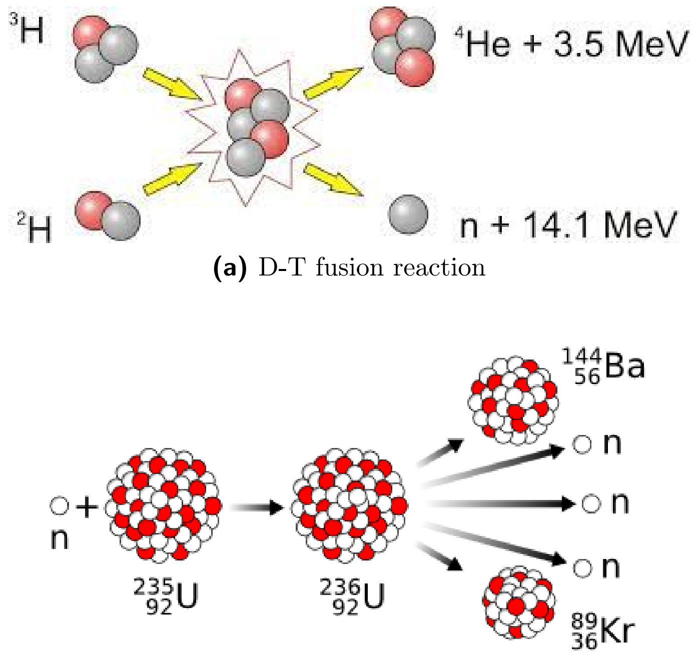

10^3 isotopes in a single library
1 .ACE file per isotope
10^2 reactions per .ACE file (MT numbers)
~ 10^5 different cross sections to be verified!

Figure 1.8: Order of magnitude of cross sections to be tested in a single nuclear data library

produces. Consistency checks are performed to ensure that no unreasonable value is found on the reactions thresholds, angular distribution, energy distribution and particle production. Additionally, an extensive set of plots is produced that can be examined for possible issues. PREPRO is a collection of 18 computer codes that can be used to extensively check and correct evaluated data prior to use them in applications and to pre-process the data into a form that makes their subsequent use much easier (e.g. linearization, adding resonance contribution, Doppler broadening, etc.). In terms of QA, it can be used to check that all ENDF formatted output is actually FORTRAN, C and C++ compatible, to check that each cross section is consistently equal to the sum of its parts (when linearized) or to ensure that MT indexes are correct. Moreover, it provides means to plot cross sections, angular distributions and energy distributions in addition to the possibility to compare cross sections coming from different evaluations. There are many other codes focused on cross-sections processing and QA, a few examples are:

- FRENDY [47], another cross-section pre-processor developed by the Japan Atomic Energy Agency;
- FUDGE [48], developed by the Lawrence Livermore National Laboratory, is written in python with some C++ routines and can be used for data manipulation of all kinds;
- RULER [49], available through the IAEA Nuclear Data section, is similar to NJOY, faster, but with less features.

To simplify the use of these tools and to reunite in a single place their different capabilities, higher level software such as ADVANCE [50], MyENDF [51] or the NDEC platform [52], have been developed in recent years. These are able to make use of the lower level tools above described in order to conduct extensive V&V procedures that are focused on the quality assessment of the cross section evaluation process. This mostly consists in checking that the data files format is correct and that the main physical constraints are fulfilled.



(a) D-T fusion reaction

(b) U^{235} fission reaction

Figure 1.9: Comparison between fission and fusion reaction

Finally, there is also an on-going effort aimed at automating the pre-processing, run and post-processing of extensive suites of benchmarks. The NDEC platform implements this kind of benchmarking, even if its primary focus remains quality assessment, while a specific suites of automated benchmarks have been developed at Los Alamos Laboratories [53]. Both of these projects, though, present a quite strong nuclear fissile applications bias while a fusion-oriented V&V is missing.

1.3.2 The Fission Bias

The commitment to develop a V&V software on nuclear data libraries which is focused on fusion application is significant, since there are many differences that distinguish fusion from fission. The most relevant ones are discussed in the following paragraphs.

Energy range

Even if there are many possible fusion reactions, the one involving deuterium and tritium isotopes is arguably the most important one, at least for magnetic confinement reactors. As shown in Fig. 1.9, the two hydrogen nuclei isotopes fuse together giving birth to an alpha particle and to a high energy neutron (>14 MeV). These neutrons are at the base of the fusion technology, since they are the main carriers of the energy that later will be converted into electric power. For fission application, instead, the energy range that is typically considered is much lower. Indeed, for classic U^{235} fission reaction, the neutron are produced on average at 2 MeV and they need to be slow down up to the thermal range in order to properly continue the fission chain reaction. For many years in the past (and today), the majority of the effort on V&V of nuclear data libraries has been conducted with a bias for fission application and extensive data and experiments can be found in literature regarding fission energy range neutrons, while high energy nuclear data is still quite scarce. This makes it a very active field of research.

Different materials and cross sections

Since the applications are different, the focus on certain materials or cross sections can be very different between fission and fusion applications. For instance, Uranium has almost no use in fusion while is widely studied in fission. At the same time, innovative materials are currently being developed for fusion applications such as the EUROFER-97 [54], which is a new type of steel that presents reduced activation and improved mechanical resistance. With regards to cross sections there is the obvious difference between fission and fusion rates but that is not all. As an example, tritium production is of paramount importance in TOKAMAK applications since, in the future, it will need to be bred in order to have self-sustaining fusion reactions. Another example may be the nuclear heating, which is an important quantity for both fusion and fission applications, but at different scales. Indeed, since superconductive magnets need to be maintained at cryogenic temperature in order to avoid thermal quench and destruction [55], even very small amount of heating can make a difference, meaning that fusion calculations are much more sensible to these cross sections variations. Another important difference may be found between the focus on fission products chains (typical of fission applications) or on materials activation (typical of fusion applications).

These are just a few examples but they are sufficient to hint why the V&V work conducted on fission oriented data for all these years should be integrated in order to ensure that fusion calculations can have a similar reliability.

Integral parameters

Last but not least, a discussion should be made on the integral parameters. In fission applications, there are a number of parameters such as the multiplication coefficient (k), Nubar or the Beta-effective, that are representative of the entire reactor, or at least of a key behaviour of the system analyzed. This can become very important for cross sections data validation process, since they can be used to quickly evaluate how new modifications impact the calculations results. Unfortunately, the same is not true for fusion applications, where suitable integral parameters are yet to be identified. Fusion reactors often need a much more detailed and complex geometry to be modelled, with many regions that present different needs. As an example, one can think how different are the requirements and the environments for a face plasma component in terms of temperature and neutron fluence with respect to superconductive magnets at cryogenic temperature. To proper verify that results are in accordance with expectations, there is no single integral parameter to be checked, but, instead, many tallies must be investigated which will be heavily discretized in space and energy.

This does not mean that spot checks are not needed in fission applications. By definition, parameters integration causes loss of information. The gain is a more simple to use and quick indicator. A proper V&V procedures though, cannot neglect sensitivity calculations or analysis of the distribution of the different computed reaction rates of interest. Nevertheless, having a consolidated definition of these integral parameters is undoubtedly an advantage in terms of V&V for nuclear data specific for fission applications.

1.3.3 The FENDL libraries

Many nuclear data libraries have been developed through the years to address fusion related needs. The Fusion Evaluation Nuclear Data Library (FENDL) was the response of the IAEA to the need for a data library specifically designed for fusion applications. FENDL libraries [56], which are developed under the auspices of the IAEA Nuclear Data Section, provide dedicated data libraries for many kinds of fusion applications and, in particular, they are the reference libraries for the ITER project. FENDL consists of General Purpose and Activation parts, both of which contain neutron, proton and deuteron-induced files with a maximum energy of the incident particle at least equal to 60 MeV and typically extended up to 150 MeV. FENDL does not directly evaluate the cross sections that it includes in the library but it collects evaluations coming from other libraries such as ENDF [57], JEFF [58], JENDL [59] TENDL [60] and BROND [61] . The FENDL library is periodically updated in order to improve and correct data and to expand the range of applications. These updates require systematic and extensive V&V process.

Development started at the IAEA/NDS back in 1987 [62]. A first version, FENDL-1, was released in 1994 [63] with 57 data files selected in a careful evaluation procedure from the regional nuclear data libraries developed in the US, Japan, the EU and the Russian Federation (i.e., ENDF/B-VI, JENDL, JEFF and BROND). This first version was used for early ITER design calculations and qualified thanks to international in-kind benchmark effort. A further update based on the data evaluations from the then state-of-the-art nuclear data libraries has led to the improved FENDL-2 library [64] which were improved in 2003 following a recommendation coming from a IAEA Consultant's Meeting. This update included up-to-date data evaluations to remove apparent deficiencies and replace obsolete evaluations [65]. This led to the release of a new FENDL-2 sub-version, FENDL-2.1 which served as a reference data library for ITER neutronics calculations despite some drawbacks and deficiencies revealed, among others, in the course of the experimental FENDL-2.1 validation activities. This exercise highlighted the need for further data improvements and provided recommendations for FENDL-3, the next major library release that was developed over the years from 2008 to 2012. The new library includes major extensions and updates with regard to the covered neutron energy range (up to 150 MeV to serve also the needs of the IFMIF [66] fusion neutron source), the library content that reached 180 isotopes and the quality (improved evaluations for many isotopes and reactions, including gas production data and secondary energy-angle distributions). FENDL-3 became also more comprehensive through the use of evaluations from the TENDL project (theoretical models) for those major isotopes data that was missing evaluations. After extensive benchmarking, FENDL-3 was recommended and formally adopted as new reference data library for ITER. Nevertheless, a few issues were discovered through the years, leading to the release of additional versions up to the current one, FENDL-3.2, in January 2022, which contains data for 191 isotopes.

As it can be observed, FENDL is essentially a community-driven tool, with a fairly long release cycle and with the majority of V&V historically conducted by hand and in-kind. This is one of the reasons why an IAEA-FENDL consultants' meeting in 2018 [56], highlighted the need for a standardized, automated and exhaustive V&V procedure covering all the nuclear data responses of interest (e.g. flux, heating, gas production, displacement per atom, shut-down dose rate, etc.). Such a test suite would have helped anticipating possible problems, spotting missing and/or inconsistent data, and estimating the impact of new releases over the different responses. Moreover, since contributions to the FENDL project are provided in kind, the development of a common open-source tool where contributions coming from many parties can converge would help with the issue of effort fragmentation.

Chapter 2

JADE Description

This chapter is focused on describing in detail JADE, the subject of this dissertation. In particular, Section 2.1 will present JADE scope and the development strategy that were adopted for the project. Section 2.2 will provide both a high and low level description of JADE architecture implementation, while Section 2.3 will describe in detail the benchmarks that are already part of JADE test suite.

As it will be repeated later on, JADE is an open source project and is available on GitHub [1]. A complete and detailed documentation of the code and its usage has also been produced and it is hosted online on ReadTheDocs [2].

An order of magnitude of JADE dimension can be derived by the number of its code lines, which surpasses twelve thousands.

2.1 Scope and development strategy

This section is dedicated to explore in more detail JADE scope and present the development strategy that has been adopted for the project.

2.1.1 JADE scope

As discussed in Section 1.3.1, JADE is not alone in the nuclear data V&V software space. Nevertheless, the majority of its "competitors" either are mainly focused on quality aspects of the data or are strongly biased towards nuclear fission applications and the importance of such bias has been discussed in Section 1.3.2. In addition to this, it has been highlighted in Section 1.3.3 how there is a lack of standardization in the V&V procedure of nuclear data libraries, especially for fusion application. Moreover, a significant portion of nuclear data V&V is currently conducted "by hand", in a field where manpower is chronically scarce. This translates in effort fragmentation and in a reduction in scope of the V&V process combined with longer time for new releases and bug correction.

For all these reasons, in early 2019, once the stakeholders requirements were collected, prioritized and listed, the development of a new and open source software called JADE started as a collaborative effort between Fusion For Energy (F4E), NIER Ingegneria s.p.a. and University of Bologna (UNIBO). The suite of automated tests carried out by the tool are focused on the nuclear data obtained at the end of

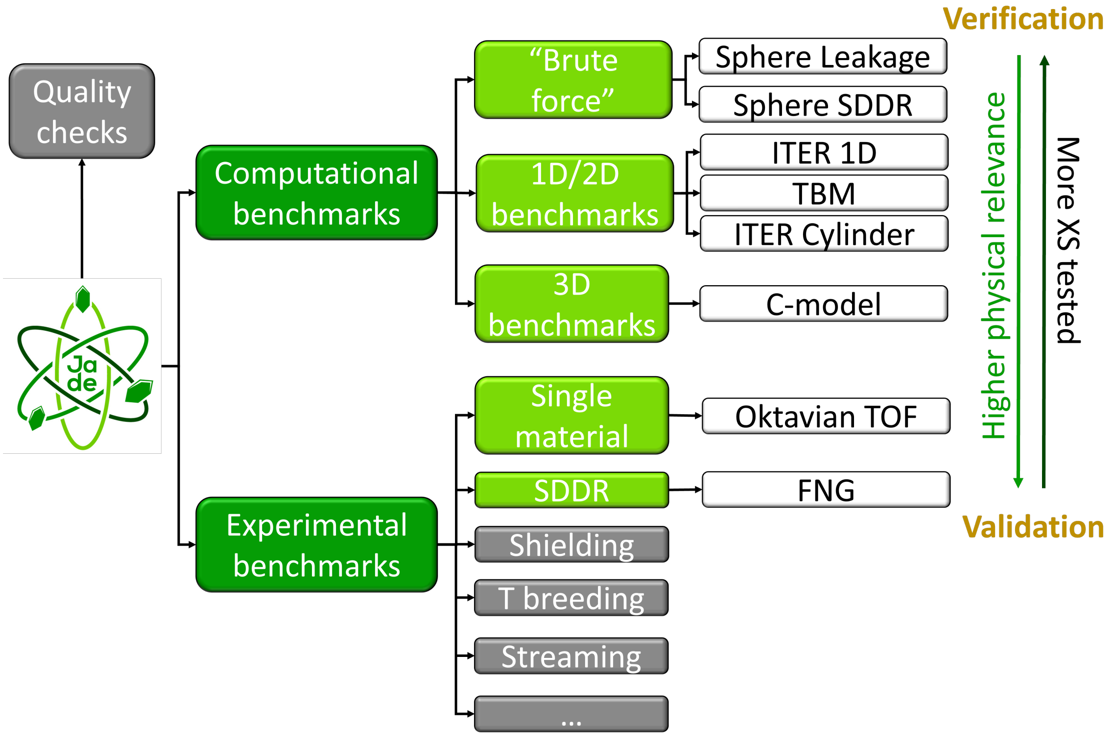


Figure 2.1: JADE conceptual scheme. Tree leaves that have not yet been implemented are greyed out

the production chain, which is the data contained in ACE files. The hope is that the level of standardization and automation offered by JADE may allow for much shorter time passing between new nuclear data libraries releases and, at the same time, improve their quality.

As shown in Fig. 2.1, three main branches were identified when conceptualizing JADE: quality checks, computational benchmarks and experimental benchmarks. At early stage of development, it was realized that much had already been done in the fusion community with regards to quality assurance routines. For this reason, the development of that branch was dropped in order to avoid duplication of work and to avoid scope creep. Nevertheless, it is not excluded that in future the capability to interact with these already existing tools will be introduced in JADE. Experimental benchmarks are JADE validation branch and, in an ideal world, this would be all that is needed. Indeed, what is mostly important for nuclear data libraries (and transport codes that use them) is to allow to simulate with accuracy the physical reality. Unfortunately, the number of available experiments will never be sufficient to have enough data to test all cross sections, let alone to explore with sufficient detail the entire energy spectrum. This is why the verification branch, which is composed by computational benchmarks, is needed. Here, a much bigger portion of cross sections can be explored, although an actual physical reference to compare the obtained results is missing. For this reason, verification in JADE consists of either to perform consistency checks (e.g. controlling that no negative values are tallied in the quantities of interest like particle flux or heat generation) or com-

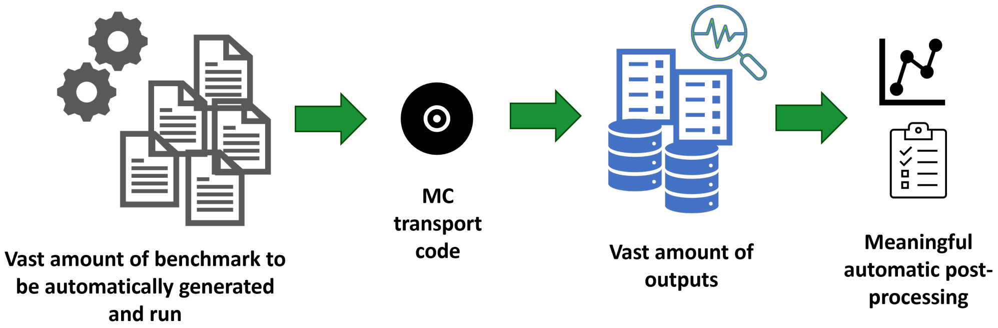

Vast amount of benchmark to be automatically generated and run
MC transport code
Vast amount of outputs
Meaningful automatic post-processing

Figure 2.2: JADE main requirements

pare the results obtained using different libraries. Combining computational and experimental benchmarks forges a spectrum where at one end are the more 'brute force' computational benchmarks which test every single isotope of a library, but with a very low physical relevance, while, at the other end, there are experimental benchmarks with very high physical relevance but a reduced number of isotopes and cross sections that are validated. In between, there are all those benchmarks that aim to model some real application, such as a fusion reactor for example, but are computational benchmarks nonetheless, since the application has not been realized yet or the model is too simplified to be used for an experimental campaign. The aim of these benchmarks is to compensate as best as possible for the lack of experimental data while trying to get results that have some degree of physical reference. Considering this topic, Fig. 2.1 also displays a classification of all benchmarks that have been already implemented in JADE and that will be better presented in Section 2.3.

In order to reach its goal, JADE has to satisfy a few requirements which are visualized in Fig. 2.2. First of all, JADE must be able to automatically generate vast amount of benchmarks inputs, especially for the benchmarks that have been categorized as 'brute force'. As discussed in Section 1.3.1, the total number of cross sections that would need to be potentially tested is in the order of 10^5 with up to thousands of isotopes in a single nuclear data library. An example of a brute force benchmark is the Sphere Leakage, described in Section 2.3.1, which runs a distinct simulation for each isotope in a library. The variety and number of benchmarks implemented in JADE are necessary in order to have extensive V&V of the nuclear data libraries and in order to deal with the absence of integral parameters in fusion applications. These benchmarks then need to be automatically run, meaning that JADE has to be able to interact with a Monte Carlo transport code. The capability to run many different benchmarks containing a great number of tallies would be useless though if JADE was not able to efficiently deal with the large amount of outputs that they can generate. For this reason, another key requirement for the code is the capability to automatically parse all the raw output coming from the benchmarks simulations and apply an automatic post-processing which is able to extract meaningful results. The final goal is to obtain comparative data that can be easily examined by cross sections evaluators, helping them to understand the effects that the cross section modifications they introduced have on the transport simulations. How all of this was achieved will be discussed in Section 2.2.

2.1.2 Strategic choices and tools

In this section, the main tools that were used during development are presented and the motivations that led to their usage are discussed.

The python choice

python is a high-level, interpreted and general purpose programming language and it supports multiple programming paradigms, including structured, functional and Object Oriented Programming (OOP). It was conceived in the late 1980s while python 3.0, the last major release, was published in 2008.

Rather then building the majority of its functionalities into its core, python has been designed to be highly modular, which is one of the reasons that make it so popular as a means of adding programmable interfaces to existing applications. Every text file containing python code and with a .py extension can be run by a python interpreter. If the code contained in the file is meant to be run stand-alone, the file is often referred as a python *script.* If, instead, it contains code to be imported and used on other scripts, it is referred to as a *module.* A collection of hierarchically organized modules that are developed to fulfill a common objective is called a *package.* A package, in turn, may be a stand-alone application (such as JADE) or a *library* to be imported by other scripts and modules for them to build on it.

Python is arguably the fastest-growing major programming language. To get an idea of the progression of its adoption, Fig. 2.3 reports the number of questions asked on the popular programming support website Stack Overflow per month from 2009 to 2022. This kind of plot can be reproduced using Stack Overflow Trends at [67]. Even if popularity is not necessarily a meaning of a language quality, the wide adoption brings massive benefits to users, which can count on vast collection of libraries and great support online.

The programming paradigm chosen for JADE development is OOP which is based on the abstraction concept of ‘objects’. This programming paradigm shares with other paradigms the use of variable and functions, but introduces containers that can include either data, which in python is defined as *attributes* or code in the form of functions, referred to as *methods.* A ‘class’ defines all the attributes and methods of a given type of object, while with the term ‘object’ one refers to a specific instance of a class. A program may create different instances of the same class which are independent and that are created with a different set of input parameters. Languages supporting OOP, like python, typically make extensive use of composition and inheritance for code reuse and extensibility. Compositions simply means that an object could contain another object and it is used to express the fact that the two objects have a relationship. For instance, a Document object will probably contain at least an Author object and a Content one. Inheritance, instead, is used to express the fact that a particular object should be thought as if it was a particular type of another one. Continuing with the previous example, Dissertation could be a child of the Document class: this means that Dissertation will inherit all methods and attributes of Document allowing at the same time for other things to be added to it, such as a Supervisor object. Sub-classes may override

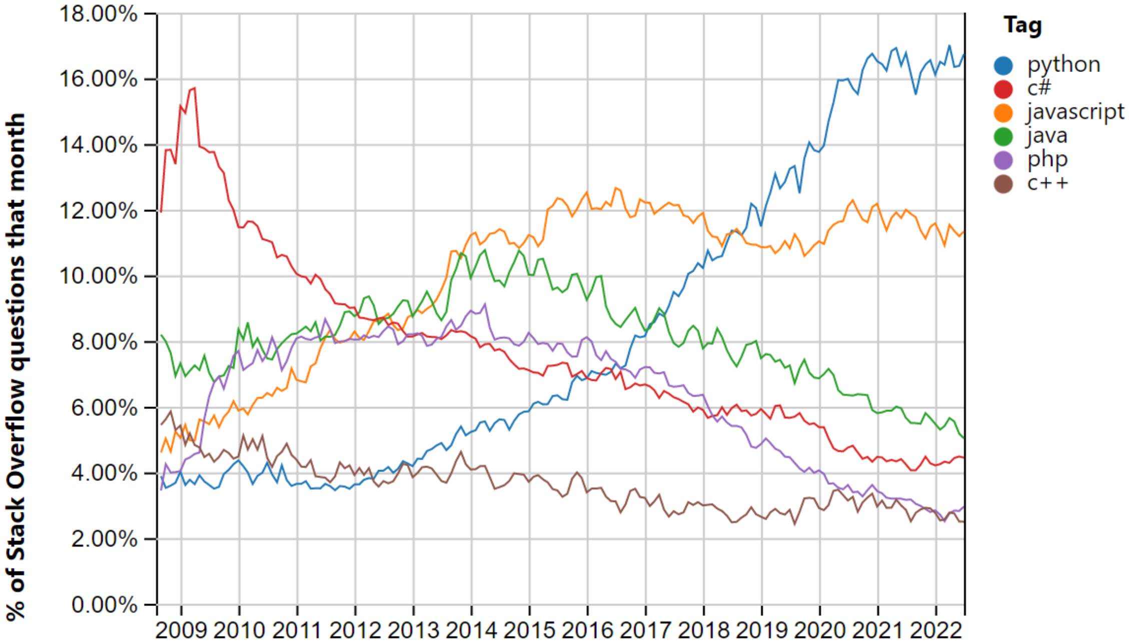


Figure 2.3: Questions per month on Stack Overflow on popular programming languages during the years

methods or attributes defined by the super-classes and python also support multiple inheritance. Composition and inheritance can be combined, or used as alternatives to one another. Who favors the use of composition over inheritance, for example, would advocate that the correct way to define the Dissertation class is not for it to be a child of Document, but instead having a Document as internal object. This would guarantee for the Document object to remain private (i.e. inaccessible from external code) even if the Dissertation class was to have many other public methods and attributes.

To summarize, the decision to select python as the programming language for the project was made based on the following considerations:

- **open-source**: python is an open-source language, with a wide community support and tons of libraries;
- **easy to learn**: python is fairly easier to learn (and to use) with respect to other competitors such as C++ or JAVA, which is one of the reasons why its use in the engineering field has been constantly growing in the last years;
- **general purpose**: python is often compared to MATLAB in the engineering field, since it has very similar capabilities at handling matrices or producing plots. Nevertheless, this is just a very small fraction of python potential. In fact, python is a full general purpose language;
- **portability**: python is portable through all operative systems, meaning that the same code can run in Windows, Linux ans IOS without modifications. Unfortunately, this is not true anymore if OS-specific libraries are used.

python also provides the possibility to generate on the same machine various ‘virtual environments’ which can contain a combination of different installed packages and packages versions. It is also possible to ‘freeze’ such an environment and copy it in a different machine in order to ensure that applications can be run in the same conditions were they were developed and tested.

The MCNP choice

MCNP has been introduced in Section 1.1.4. The choice to use it as the go to transport code for JADE was mainly driven by two factors. The first one is that MCNP is one of the most used and validated transport code, making it a perfect candidate for a project focused on V&V. It is maintained by a reliable institution like the Los Alamos Laboratories which guarantees for the code quality and reliability also thanks to a strict control of the source code distribution. The second reason is that MCNP is one of the official codes which may be used for transport calculation in the ITER project. Since F4E is one of the three parties involved in the development of JADE and one of the first applications of the tool has been done on the FENDL libraries (reference nuclear data libraries for ITER), MCNP was almost a mandatory choice.

The main drawback of this is that MCNP is not an open-source code (on the contrary its policy distribution is fairly restrictive) and that its inputs and outputs are not particularly user-friendly if compared with more recently developed transport code. Nevertheless, these new codes cannot claim the same long history of validation that MCNP has.

The Windows OS choice

JADE development and testing has been conducted exclusively on Windows operative systems. The main reason for this derives from the choice of using Microsoft Excel as the principal tool for the visualization and formatting of the post-processed simulation outputs. This choice allowed to considerably speed up the development of the project, since the coding of a visualization tool from scratch was avoided. In fact, post-processed data could be simply dumped in Excel files and the powerful formatting features of Excel could be used to improve the readability and usability of the results. Moreover, there are at least a couple of robust python libraries such as xlwings [68] and openxl [69] that allow to easily interact with the Microsoft application.

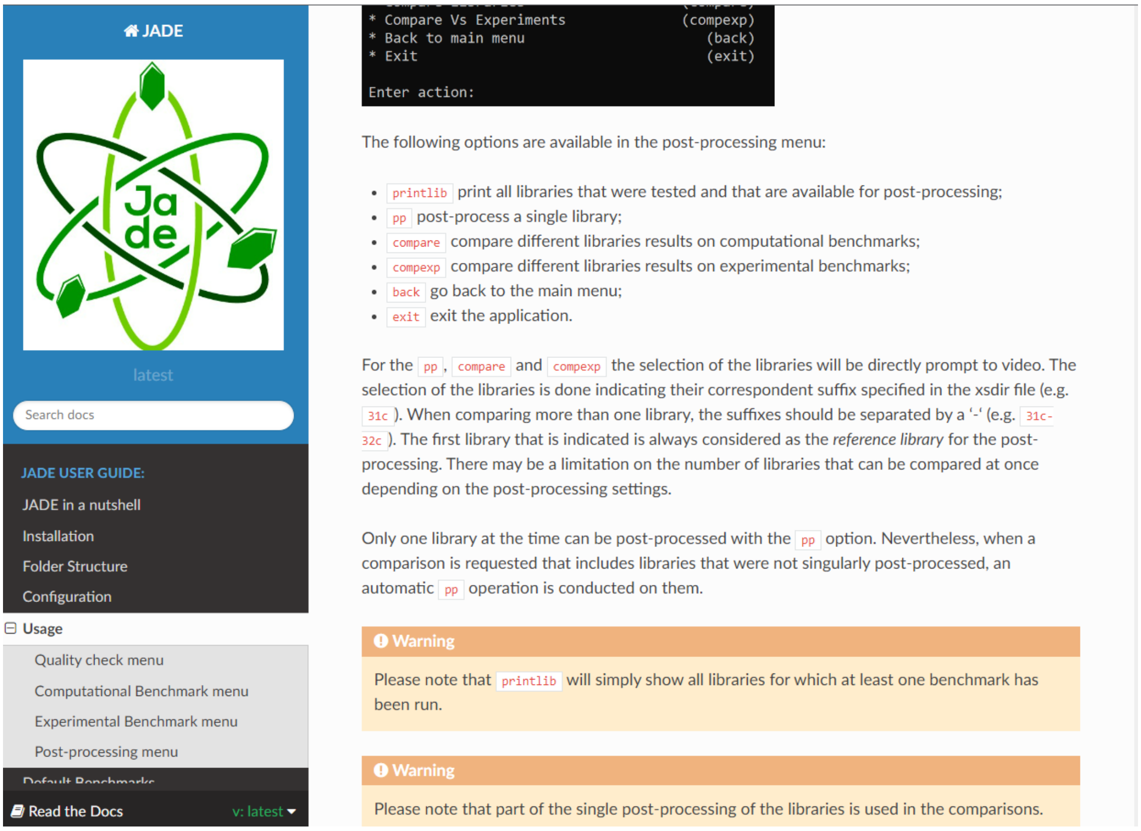
GitHub and ReadTheDocs

GitHub has been one of the most important tools during the development of JADE. Essentially, it is an online repository that integrates the Git version control system [70] with various additional feature. Originally created by Linus Torvalds to help with the development of the Linux operative system, Git has quickly became the standard for version control and collaborative development in the open source world. Its main feature is that it allows different programmers to work on the same core code but in an independent way thanks to a tree structure, where, from a single root, many branches and sub-branches can span out, each one representing the work of a single developer or of a specific feature. Git is able to track all changes that are made in the different branches which can then be merged together at any time. Developers can also simply update their code from the root or a parent branch without having to commit their modifications and this guarantees that they are always up to date with the main code. Moreover, Git handles also the code version control, i.e., it allows to freeze in time a specific status of the code labelling it with a tag. GitHub is a special online repository that allows to use Git in a very simple way and, at the same time, provides a number of useful features like the possibility to open and track issues and to implement Continuous Integration (which will be better discussed in Section 2.2.6). An "issue" can represent different things, from an actual bug, to a request of a new feature or enhancement in the code or in the documentation. For each of these issues, a thread is open that allows to discuss it and to track its resolution. To give an idea of how important this feature has been to the project, at the moment of writing, a total of 151 issues have been opened on JADE since its inception, of which 131 have been closed.

The other major web-tool that has been used during the development of JADE is ReadTheDocs. This is an online repository which helps automatizing the creation, versioning and hosting of (open source) technical documentation. For python-based projects, the power of sphinx [71] is leveraged in order to compile the source documentation written using the ReStructured Text (RST) language in different formats such as html or LaTeX. All files needed for the compilation of the documentation are stored inside a dedicated folder inside JADE repository, to which ReadTheDocs is linked. In this way, every time a push is made to JADE main branch on GitHub (i.e., when the local code version is synchronized with the cloud one), the online documentation is automatically recompiled by ReadTheDocs. This navigable html documentation contains information on:

- how to install JADE;
- JADE folder structure;
- A thorough description of JADE configuration files and how to modify them;
- how to use JADE;
- a description of all implemented benchmarks;
- examples of post-processing;
- a Tips&Tricks and a Troubleshooting section;
- how to implement new benchmarks in JADE suite;
- the API description of some of his classes.

Fig. 2.4 shows a screenshot of the documentation.



```
 * Compare Vs Experiments             (compexp)
 * Back to main menu                     (back)
 * Exit                                  (exit)

Enter action:
```

The following options are available in the post-processing menu:

- `printlib` print all libraries that were tested and that are available for post-processing;
- `pp` post-process a single library;
- `compare` compare different libraries results on computational benchmarks;
- `compexp` compare different libraries results on experimental benchmarks;
- `back` go back to the main menu;
- `exit` exit the application.

For the `pp` , `compare` and `compexp` the selection of the libraries will be directly prompt to video. The selection of the libraries is done indicating their correspondent suffix specified in the xsdir file (e.g. `31c`). When comparing more than one library, the suffixes should be separated by a '-' (e.g. `31c-32c`). The first library that is indicated is always considered as the *reference library* for the post-processing. There may be a limitation on the number of libraries that can be compared at once depending on the post-processing settings.

Only one library at the time can be post-processed with the `pp` option. Nevertheless, when a comparison is requested that includes libraries that were not singularly post-processed, an automatic `pp` operation is conducted on them.

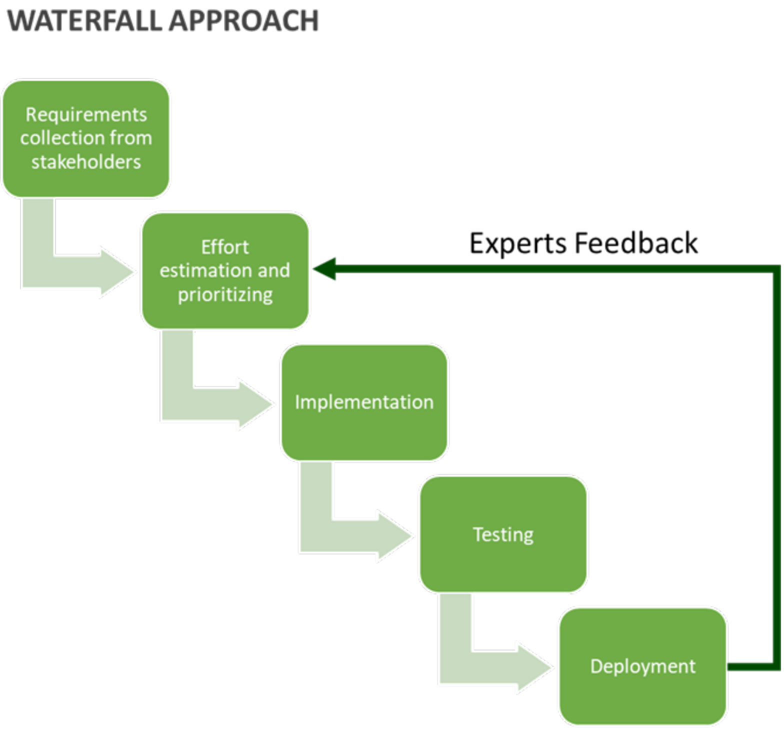
Warning

Please note that `printlib` will simply show all libraries for which at least one benchmark has been run.

Warning

Please note that part of the single post-processing of the libraries is used in the comparisons.

Figure 2.4: Screenshot of JADE html online documentation



WATERFALL APPROACH
Requirements collection from stakeholders
Effort estimation and prioritizing
Implementation
Testing
Deployment
Experts Feedback

Figure 2.5: Illustration of the waterfall approach used during JADE development

Development approach

To develop JADE, the Waterfall methodology was followed. This is a project management approach that emphasizes a linear progression from beginning to end of a project. The waterfall methodology is a sequential development process that flows like a waterfall through all phases of a project, with each phase completely wrapping up before the next phase begins. This means that the work is based on fixed dates, requirements, and outcomes. With this method, the individual execution teams aren't required to be in constant communication and, unless specific integrations are required, are usually self-contained. It is said that the Waterfall methodology follows the adage to "measure twice, cut once." and it depends on the belief that all project requirements can be gathered and understood upfront.

The Waterfall methodology was chosen as it is a straightforward, well-defined project management methodology with a proven track record. Additionally, since the available manpower was limited, this methodology was deemed to be the most effective one in terms of avoiding scope creep. As described in Fig. 2.5, at first, the requirements coming from the main stakeholders were collected and the necessary effort of the different work packages was estimated. Then, an implementation and testing phase followed. Finally, the tool has been deployed and used for production cases, feedback has been received by the experts and new work packages were defined.

Other benefits of the Waterfall method include:

- the total cost (in terms of ppy) of the project can be accurately estimated, as the timeline can, after the requirements have been defined;
- it is easier to measure progress according to clearly defined milestones;
- stakeholders aren't able to constantly add new requirements to the project, delaying production.

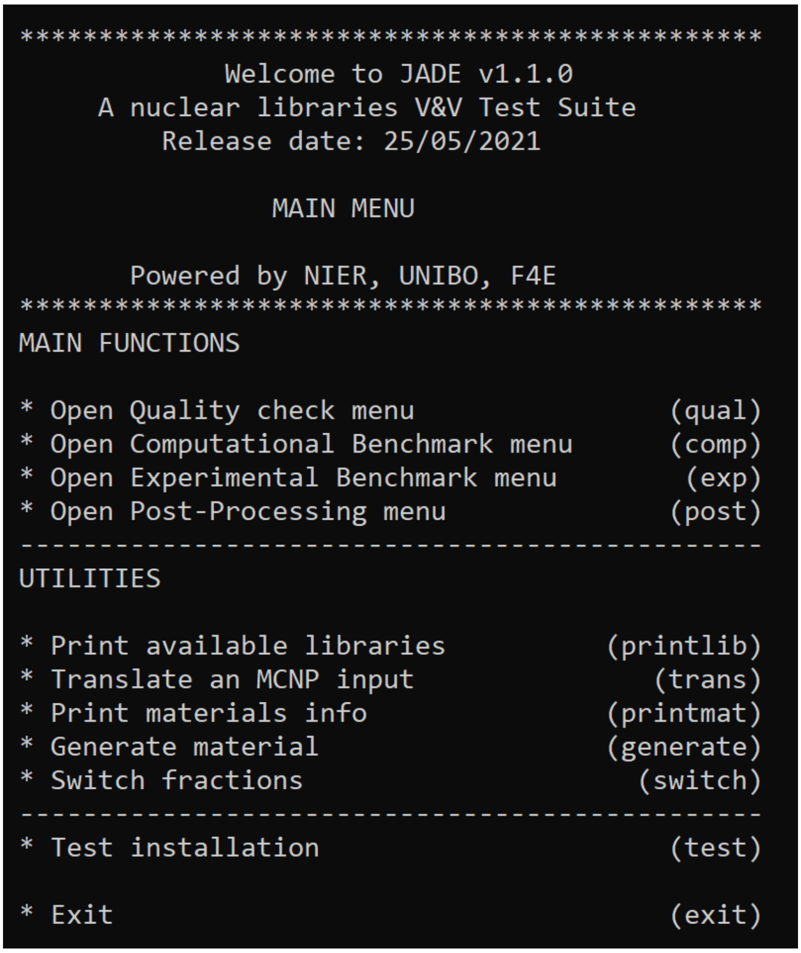
2.2 Implementation

In the following sections, JADE structure and capabilities will be described. Each section will be divided in two parts, one describing the structure and functions of the code at a higher level, while the other will be focused on how that was implemented in JADE Object Oriented Programming (OOP) paradigm.

2.2.1 General architecture

High-level description

One of the first thing to specify is that, for the time being, no Graphical User Interface (GUI) has been developed for JADE. This is simply because building a GUI can be an extremely time-consuming task which adds very little in this case to the scientific value of the project. Nevertheless, a surrogate GUI is used in JADE and the user can interact with the code via menus that are directly loaded on the command prompt. An example of such menus is displayed in Fig. 2.6. The drawback of not having a GUI is that all settings customization need to be implemented



```
***********************************************
             Welcome to JADE v1.1.0
      A nuclear libraries V&V Test Suite
          Release date: 25/05/2021

                MAIN MENU

       Powered by NIER, UNIBO, F4E
***********************************************
MAIN FUNCTIONS

* Open Quality check menu                (qual)
* Open Computational Benchmark menu      (comp)
* Open Experimental Benchmark menu        (exp)
* Open Post-Processing menu              (post)
-----------------------------------------------
UTILITIES

* Print available libraries          (printlib)
* Translate an MCNP input               (trans)
* Print materials info               (printmat)
* Generate material                  (generate)
* Switch fractions                     (switch)
-----------------------------------------------
* Test installation                      (test)

* Exit                                   (exit)
```

Figure 2.6: JADE main menu

in external files for better usability, that is, working with extensive setups on a command prompt would give a very bad user experience. Since the great majority of these settings can be expressed as tables, the natural choice would have fallen on CSV (Comma Separated Values) file. Once again though, it was decided that this file format would have given a bad user experience in case of long tables, hence the decision to store JADE settings in Microsoft (MS) Excel files. As discussed before, Excel was already introduced as a requirement for the current development of JADE as one of the primary output format, so it made sense to use it also to store settings and take advantage of its more user friendly way to organize and visualize data. Additional details on JADE configuration can be found in Section 2.2.2.

Before starting the application for the first time, the user should edit the main configuration file that allows to set a few ambient variables and to customize the selection of benchmarks to be run or post-processed by the code. Then, from the command prompt, the user can request the assessment of a specified nuclear data library. At this point all selected benchmark inputs will be automatically generated and then run using MCNP or D1S-UNED, depending on the specific benchmark. Once the simulations have been completed, the automatic post-processing can be run. It is worth to notice that, the post-processing can be customized through benchmark-specific Excel configuration files. The output of the post-processing includes:

- raw data in .csv format containing the entire tallied output of the simulations (see Section 2.2.4 for additional details);
- formatted Excel recap files that, through color codes, help to spot inconsistencies among the results of the simulations (see Section 2.2.4 for additional details);
- Word and .PDF atlas collecting the plots generated during the post-processing (see Section 2.2.4 for additional details).

Fig. 2.7 displays a scheme summarizing JADE general architecture. In addition to run benchmarks, JADE provides also a series of stand-alone utilities. This is because during development it often happened to realize that a specific piece of JADE could be used outside the sole scope of nuclear data V&V. These utilities are better described in Section 2.2.5.

OOP implementation

Fig. 2.8 presents the general OOP scheme implemented in JADE. As it can be observed, the user directly interacts only with the *gui.py* module. This module does not define any class but only specifies the loops that are to be executed during a JADE session. In short, this loop is responsible for all command menus that are prompted to video and for the triggering of the actions requested by the user. If, in the future, a proper GUI will be implemented in JADE, it would be fairly trivial to substitute this code with the GUI one having the GUI buttons linked to exactly the same functions that are now called by *gui.py*.

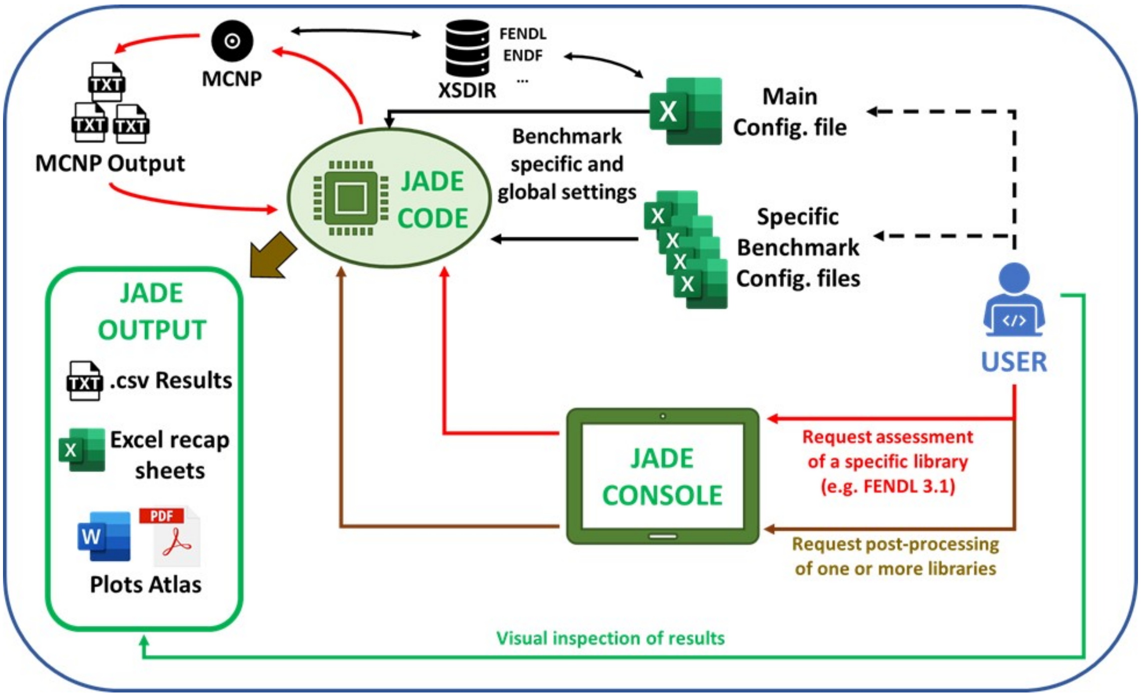


Figure 2.7: Scheme of JADE general architecture

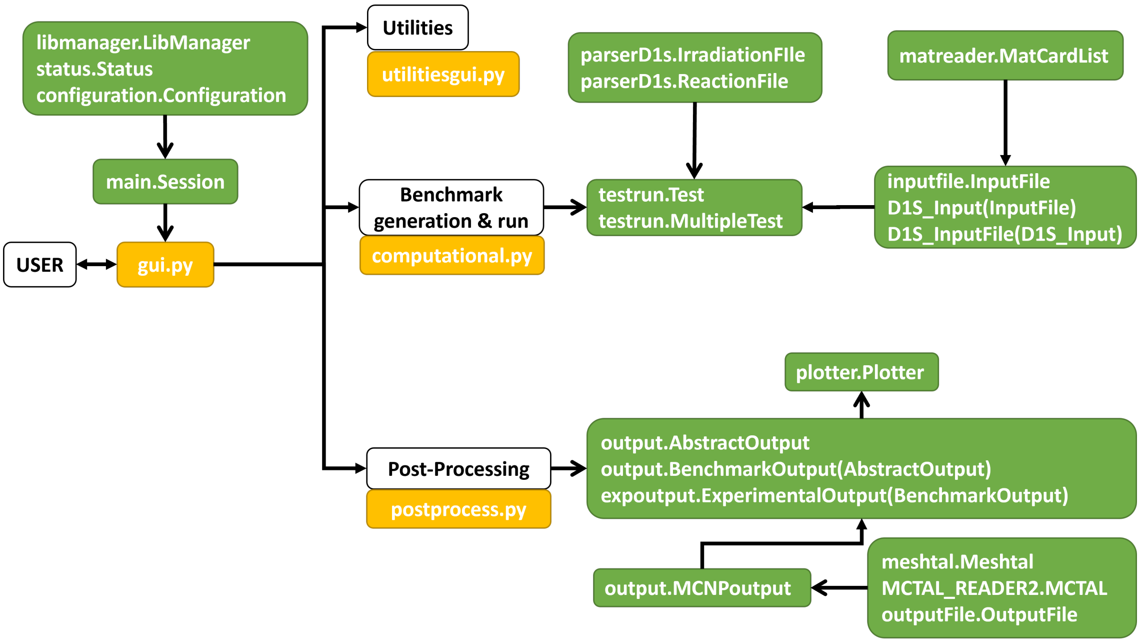


Figure 2.8: Scheme of JADE OOP implementation

When JADE is started, a Session object is created (from *Session.py*) which is a container for a series of information and tools that many part of the code may need to access. In particular it contains:

- all the paths to the different folders that constitutes the JADE folders tree;
- the Status object which reads and stores information on which libraries have already been assessed or post-processed (and that contains the methods to update this kind of information);
- the Configuration object, which handles the parsing of the Main Configuration file where the environment variable are set together with the selection of active benchmarks;
- the LibManager object, which is responsible for all the operations related to nuclear data libraries like checking their availability or handling the translation of single isotopes.

Generally speaking the user can request one of three things. The first is to use one of the utilities, triggering a call to *utilitiesgui.py*. The second one is to assess a library. This will run all benchmarks that are selected in the main configuration file on a specific nuclear data library through the *computational.py* module. The last one is to perform the post-processing of the results coming from one or more libraries assessments through to the *postprocess.py* module. These three modules do not define any classes, but only contain additional logic to parse and address the different requests that may come from the users. Further details on how these actions are performed are provided in the following sections where the main functionalities of JADE code are pointed out and described.

2.2.2 Configuration

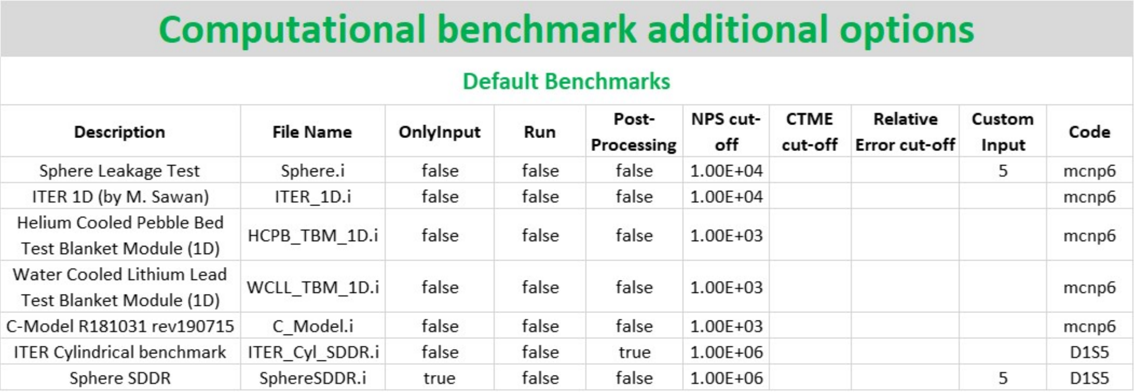
High-level description

Even if JADE has been built with the idea of standardization in mind, users are still able to customize its execution up to some extent. The customisation allow users to extend JADE benchmark suite, adjust benchmark execution and change post-processing parameters without the need to modify the software code.

JADE configuration file can be subdivided in three groups. The first one includes only the main configuration file, which is the only file whose editing is mandatory for the user since it contains a couple of environment variables that must be set up. These are:

- the path to the XSDIR file, which is the MCNP file that points to all the different ACE files that actually contain the cross sections data are located;
- the number of CPUs that MCNP should use to run the transport simulations

Moreover, the file also controls which benchmarks should be run or post-processed during a JADE session.Fig. 2.9 shows an extract of such selection sheet. As it can be seen, in addition to the selection of the active benchmarks, it is also possible to



Computational benchmark additional options

Default Benchmarks

Description	File Name	OnlyInput	Run	Post-Processing	NPS cut-off	CTME cut-off	Relative Error cut-off	Custom Input	Code
Sphere Leakage Test	Sphere.i	false	false	false	1.00E+04			5	mcnp6
ITER 1D (by M. Sawan)	ITER_1D.i	false	false	false	1.00E+04				mcnp6
Helium Cooled Pebble Bed Test Blanket Module (1D)	HCPB_TBM_1D.i	false	false	false	1.00E+03				mcnp6
Water Cooled Lithium Lead Test Blanket Module (1D)	WCLL_TBM_1D.i	false	false	false	1.00E+03				mcnp6
C-Model R181031 rev190715	C_Model.i	false	false	false	1.00E+03				mcnp6
ITER Cylindrical benchmark	ITER_Cyl_SDDR.i	false	false	true	1.00E+06				D1S5
Sphere SDDR	SphereSDDR.i	true	false	false	1.00E+06			5	D1S5

Figure 2.9: Extract of the main configuration file, computational benchmarks selection

specify which parameters should be controlling the termination of the simulation, that is, either the number of simulated histories, the total computer time or the statistical precision reached in a specific tally. Additionally, it allows to select which executable (i.e. transport code) should be used to run each benchmark. A common use case for this file may be the need to run a particularly heavy benchmark on a cluster. In this case, JADE can be asked to only generate the necessary input files for the simulation without running it. The input files can then be copy pasted and run in the external HPC solution of choice. Once the simulations are completed, the outputs can be copied back into JADE folders and the automatic post-process can be run locally.

In addition to the main configuration file, there are the computational benchmarks files, which are all based on the same structure and control the computational benchmarks post-processing. Great effort was spent in order to build an architecture general enough that it would allow users to define new computational benchmarks to be added to the JADE test suite without the need of additional code and the standardization of the computational benchmarks configuration files is part of this effort.

There are also customized configuration files required by experimental benchmarks. These benchmarks are more difficult to standardize, mainly due to the fact that the output data coming from the transport simulations cannot be directly used for comparisons with the experimental results, but often needs to be processed first. This is why, in addition to specific configuration file, to include an experimental benchmark to the JADE suite will always require some additional coding from the user.

The ensemble of configuration files that are shipped together with JADE constitute the default settings. If any of the parameters contained in this files are modified, they can be easily restored at any time by the user in order to guarantee a standard JADE execution.

OOP implementation

In terms of python implementation, the module *configuration.py* and, more specifically, the Configuration object is responsible for the parsing and storing of all parameters contained in the Main Configuration file. Benchmark-specific configuration files are not parsed by a dedicated object due to their simple structure and are directly read in the post-processing phase.

2.2.3 Input generation and run

High-level description

An important JADE feature is its ability to automatically generate and run all the inputs of the benchmarks composing the test suite which have been selected through the main configuration file. In order to do so, for each benchmark added to JADE, an MCNP input template must be provided. During a library assessment, the templates are completed with the parameters responsible for the stop of the simulations and, most importantly, they are "translated" to the nuclear data library under assessment. That is, JADE is able to convert a material section of an MCNP input to a whatever nuclear data library available in the XSDIR file, which is the native MCNP file that contains information on the installed nuclear data libraries available for transport simulations.

A material in an MCNP input simply consists of a list of all the isotopes (or zaids) that compose the material expressed in the format "ZZAAA.XXp +/-DDD" where:

- ZZ is the atomic number (i.e. the number of protons) which uniquely identify an element;
- AAA is the mass number (i.e. the total number of protons plus neutrons in the atom) which in combination with ZZ uniquely identify an isotope;
- XXp is the tag that identifies a specific nuclear data library (Xs are numbers and p is a letter codifying the type of particle);
- DDD is a floating number expressing the fraction correspondent to the zaid in the material composition; if positive, the fraction is to be intended as an atomic one, if negative, as a mass one.

The following is a simple example of an MCNP material card representing water using atomic fractions:

```
m1    1001.31c 0.66
      8016.31c 0.33
```

At first, it may appear that a translation operation would consist of a trivial substitution of the library tag for all zaids in the input, but this is slightly more complicated than that due to the use in MCNP of natural elements, that were often used in older nuclear data library releases. These elements, instead of pointing to the cross sections of a specific isotope, allow sampling on an 'equivalent natural zaid' which averages the responses of all the different isotopes that are part of that element weighted on their natural abundance percentage.

When the translation of a single zaid is requested, three different scenarios can occur:

1. the zaid is not available in the requested new library, hence a default one will be used;
2. the zaid is available in the new library and a 1:1 conversion can be performed;
3. a natural element is used instead of a zaid, i.e., AAA = 000.

During scenario (3), at first, the selected library is checked for exact correspondence, i.e., it is checked if also the new library includes the natural element. In this case, the behavior is identical to scenario (2). If this is not true, the natural element needs to be expanded: all the isotopes available in the new library that have the same element are recovered together with their atomic mass (m) and natural abundance (NA). At this point, if the original natural element fraction is expressed in atomic format (x_N^A), the new zaids deriving from the expansion will have as fraction their natural abundance (NA) multiplied for the original natural zaid fraction:

$$x_{zaid}^A = \mathrm{NA}_{zaid} \cdot x_N^A \tag{2.1}$$

If, instead, the original natural element fraction is expressed in a mass format (x_N^M), the "equivalent mass" m_N of the natural element can be calculated as:

$$m_N = \sum_{zaids} \mathrm{NA}_{zaid} \cdot m_{zaid} \tag{2.2}$$

and then the mass fraction of each expanded new zaid x_{zaid}^M can be computed as:

$$x_{\mathrm{zaid}}^M = x_N^M \cdot (\mathrm{NA}_{zaid} \cdot m_{zaid})/M_N \tag{2.3}$$

where $(\mathrm{NA}_{zaid} \cdot m_{zaid})/M_N$ is basically the natural abundance in mass of the zaid.

The scheme reported in Fig. 2.10 summarizes the JADE translation logic.

OOP implementation

Fig. 2.11 highlights the portion of JADE general OOP scheme that is related to input generation and run. As previously discussed, it is the *computational.py* module that contains all the logic to initiate the benchmark input generation and run, based on user requests. The generation and input of a specific benchmark is handled by a Test object or one of its children. That is, the Test object can handle all benchmarks implemented in JADE, with the notable exception of the sphere tests for reasons that will be clearer once the section dedicated to benchmarks description is read (Section 2.3). If the benchmark is actually composed by more than one geometry, for instance a series of experiments conducted in the same way but on different materials, a MultipleTest object can be created which will simply create a Test object for each specific experiment. The Test object do not directly handle the benchmark input. This is done by the InputFile object (developed for MCNP v6.2) or one of

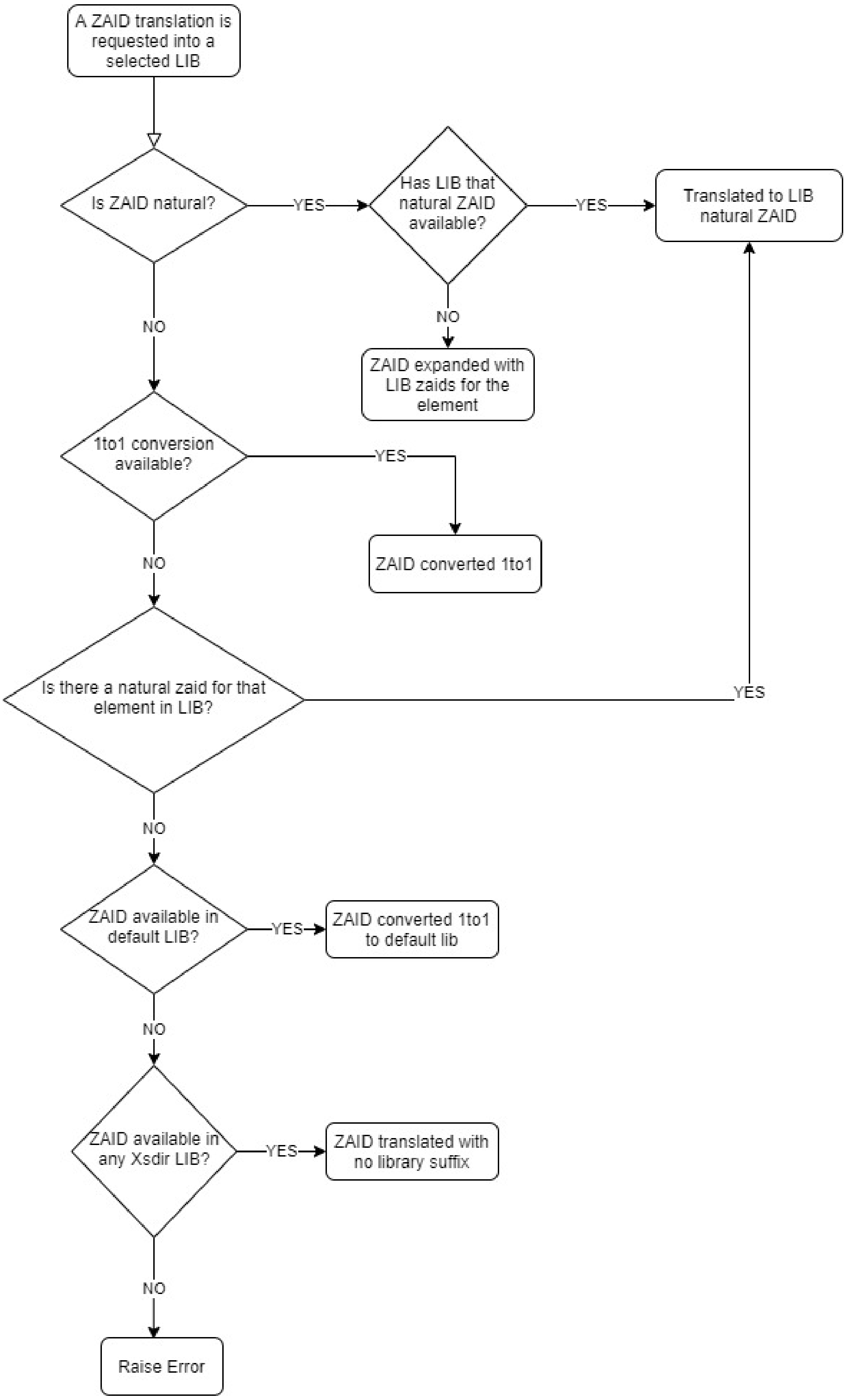

A ZAID translation is requested into a selected LIB
Is ZAID natural?
YES
Has LIB that natural ZAID available?
YES
Translated to LIB natural ZAID
NO
ZAID expanded with LIB zaids for the element
NO
1to1 conversion available?
YES
ZAID converted 1to1
NO
Is there a natural zaid for that element in LIB?
YES
NO
ZAID available in default LIB?
YES
ZAID converted 1to1 to default lib
NO
ZAID available in any Xsdir LIB?
YES
ZAID translated with no library suffix
NO
Raise Error

Figure 2.10: JADE translation logic for a single zaid

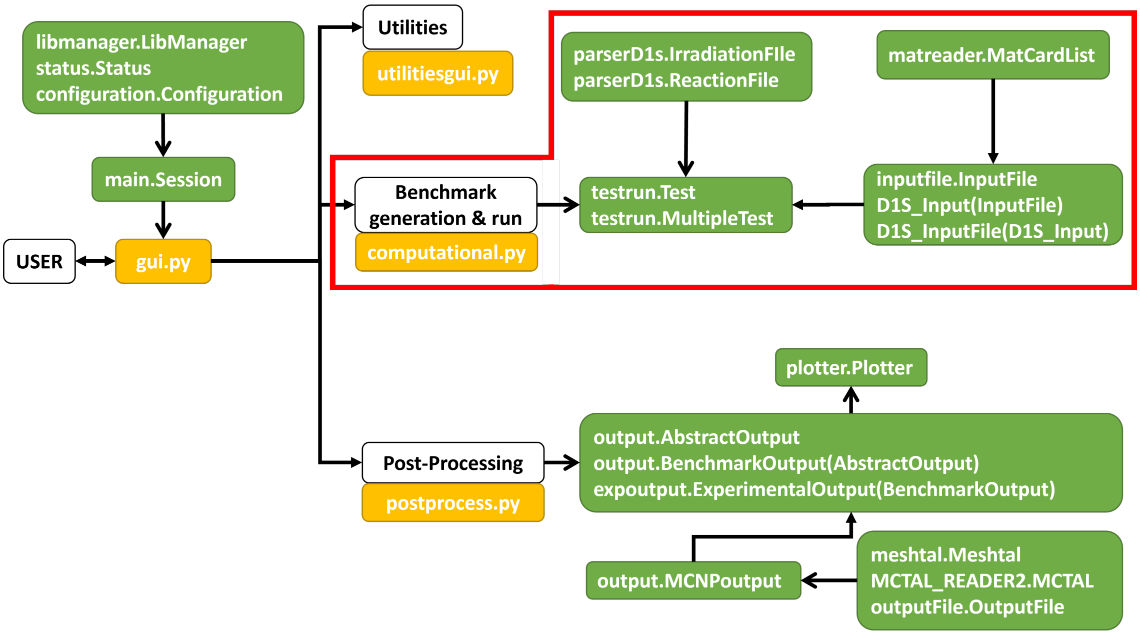


Figure 2.11: Scheme of JADE OOP implementation. Focus on input generation and run

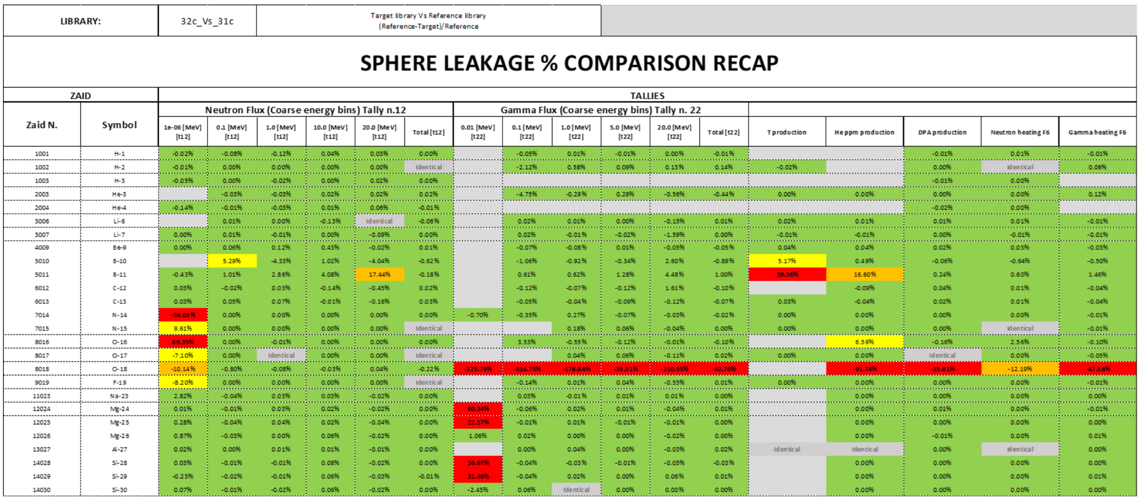
its children (for D1S versions). The InputFile object is responsible for parsing the inputs templates and modify them adding for instance the STOP card parameters and, more importantly, translating them. The translation logic though is not directly contained in this object but on a lower level parser that only deal with the material section of an MCNP input: the MatCardList object. This is the object that directly interacts with the LibManager one and applies the translation algorithm discussed in the high level description. Finally, in case of D1S inputs, additional files are necessary for a correct simulation as better detailed in Section 1.1.5. For these simple files, fully-fledged python API have been developed which are constituted by the IrradiationFile and ReactionFile objects. Their generation is handled by the Test object too.

To summarize, each time a benchmark needs to be generated, a Test object is created. A Test is composed by an InputFile and all other files that are needed for the MCNP simulations. The Test object is also the one responsible for calling MCNP and run the benchmark. A fundamental part of the InputFile is the MaterialCardList, wich is a complete API that have been developed to handle the material cards in an MCNP input.

2.2.4 Post-processing

High-level description

Once the benchmarks have been run, JADE automated post-processing can be requested. Post-processing operation are divided between single library post-processing and comparison of the results obtained from different libraries. The first type performs consistency checks (e.g. ensure non-negative responses) and collects the statis-



LIBRARY: 32c_Vs_31c

Target library Vs Reference library
(Reference-Target)/Reference

SPHERE LEAKAGE % COMPARISON RECAP

ZAID		TALLIES																
		Neutron Flux (Coarse energy bins) Tally n.12						Gamma Flux (Coarse energy bins) Tally n. 22										
Zaid N.	Symbol	1e-06 [MeV] [t12]	0.1 [MeV] [t12]	1.0 [MeV] [t12]	10.0 [MeV] [t12]	20.0 [MeV] [t12]	Total [t12]	0.01 [MeV] [t22]	0.1 [MeV] [t22]	1.0 [MeV] [t22]	5.0 [MeV] [t22]	20.0 [MeV] [t22]	Total [t22]	T production	He ppm production	DPA production	Neutron heating F6	Gamma heating F6
1001	H-1	-0.02%	-0.08%	-0.12%	0.04%	0.03%	0.00%		-0.05%	0.01%	-0.01%	0.00%	-0.01%			-0.01%	0.01%	-0.01%
1002	H-2	-0.01%	0.00%	0.00%	0.00%	0.00%	Identical		-2.12%	0.58%	0.09%	0.13%	0.14%	-0.02%		0.00%	Identical	0.06%
1003	H-3	-0.05%	0.00%	-0.02%	0.00%	0.02%	0.00%									-0.01%	0.00%	
2003	He-3		-0.03%	-0.03%	0.02%	0.02%	0.02%		-4.75%	-0.29%	0.29%	-0.56%	-0.44%	0.00%	0.00%	0.00%	0.00%	0.12%
2004	He-4	-0.14%	-0.01%	-0.03%	0.01%	0.06%	-0.01%									-0.02%	0.00%	
3006	Li-6		0.01%	0.00%	-0.13%	Identical	-0.06%		0.02%	0.01%	0.00%	-0.15%	0.01%	0.02%	0.01%	0.01%	0.01%	-0.01%
3007	Li-7	0.00%	0.01%	-0.01%	0.00%	-0.09%	0.00%		0.02%	-0.01%	-0.02%	-1.59%	0.00%	-0.01%	-0.01%	0.00%	-0.01%	-0.01%
4009	Be-9	0.00%	0.06%	0.12%	0.43%	-0.02%	0.01%		-0.07%	-0.08%	0.01%	-0.05%	-0.05%	0.04%	0.04%	0.02%	0.03%	-0.03%
5010	B-10		5.29%	-4.33%	1.02%	-4.04%	-0.62%		-1.06%	-0.92%	-0.34%	2.60%	-0.89%	5.17%	0.49%	-0.06%	-0.64%	-0.50%
5011	B-11	-0.43%	1.01%	2.86%	4.08%	17.44%	-0.18%		0.61%	0.62%	1.28%	4.48%	1.00%	39.56%	16.60%	0.24%	0.60%	1.46%
6012	C-12	0.03%	-0.02%	0.03%	-0.14%	-0.45%	0.02%		-0.12%	-0.07%	-0.12%	1.61%	-0.10%		-0.09%	0.04%	0.01%	-0.04%
6013	C-13	0.03%	0.05%	0.07%	-0.01%	-0.16%	0.03%		-0.05%	-0.04%	-0.09%	-0.12%	-0.07%	0.03%	-0.04%	0.02%	0.01%	-0.04%
7014	N-14	-39.05%	0.00%	0.00%	0.00%	0.00%	0.00%	-0.70%	-0.35%	0.27%	-0.07%	-0.03%	-0.02%	0.00%	0.00%	0.00%	0.00%	-0.01%
7015	N-15	9.61%	0.00%	0.00%	0.00%	0.00%	Identical			0.18%	0.06%	-0.04%	0.00%	0.00%	0.00%	0.00%	Identical	-0.01%
8016	O-16	69.35%	0.00%	-0.01%	0.00%	0.00%	0.00%		3.53%	-0.55%	-0.12%	-0.01%	-0.10%		6.59%	-0.16%	2.56%	-0.10%
8017	O-17	-7.10%	0.00%	Identical	0.00%	0.00%	Identical			0.04%	0.06%	-0.11%	0.02%	0.00%	0.00%	Identical	0.00%	-0.05%
8018	O-18	-10.14%	-0.80%	-0.08%	-0.03%	0.04%	-0.22%	-325.79%	-514.73%	-178.84%	-35.31%	-210.55%	-42.76%		-91.74%	-35.81%	-12.19%	-47.58%
9019	F-19	-8.20%	0.00%	0.00%	0.00%	0.00%	Identical		-0.14%	0.01%	0.04%	-0.55%	0.01%	0.00%	0.00%	0.00%	0.00%	-0.01%
11023	Na-23	2.82%	-0.04%	0.03%	0.03%	-0.02%	0.00%		0.03%	-0.01%	0.01%	0.01%	0.00%		0.00%	0.00%	0.00%	0.00%
12024	Mg-24	0.01%	-0.01%	0.03%	0.02%	-0.02%	0.00%	60.34%	-0.06%	0.02%	0.01%	-0.04%	0.01%		0.00%	0.01%	0.00%	-0.01%
12025	Mg-25	0.28%	-0.04%	0.04%	0.02%	-0.04%	0.00%	22.57%	-0.01%	0.01%	-0.01%	-0.01%	0.00%		0.00%	0.00%	0.00%	0.00%
12026	Mg-26	0.87%	-0.03%	0.00%	0.06%	-0.02%	0.00%	1.06%	0.02%	0.00%	0.00%	-0.02%	0.00%		0.00%	-0.01%	0.00%	0.01%
13027	Al-27	0.02%	0.00%	0.01%	0.01%	-0.01%	0.00%		0.00%	0.04%	0.00%	-0.03%	0.02%	Identical	Identical	0.00%	Identical	0.00%
14028	Si-28	0.03%	-0.01%	-0.01%	0.08%	-0.02%	0.00%	36.97%	-0.04%	-0.03%	-0.01%	-0.05%	-0.03%		0.00%	0.00%	0.00%	0.00%
14029	Si-29	-0.25%	-0.02%	-0.01%	0.06%	-0.03%	-0.01%	31.49%	-0.04%	0.02%	0.00%	0.06%	0.01%		0.00%	0.00%	0.00%	0.01%
14030	Si-30	0.07%	-0.01%	-0.02%	0.06%	-0.02%	0.00%	-2.45%	0.06%	Identical	0.00%	0.00%	0.00%		0.00%	0.00%	0.00%	0.00%

Figure 2.12: Example of MS Excel output

tical error associated with each tally. The second one compares the results obtained from the same benchmark using different libraries. In case of experimental benchmarks, comparisons with the experimental data are always provided.

Three main types of outputs are generated during post-processing operations: Raw Data, MS Excel files and Plot Atlas.

The Raw Data files are .csv (comma separated values) files containing the raw data coming from the parsing of the MCNP output in a table format. Very little manipulation is done on the data at this stage so that these files can be useful for the definition of additional post-processing operations outside JADE without having to re-extract the data from MCNP legacy output files.

The Excel files usually contain the most extended set of manipulated outputs. The results of each tally selected in the benchmark-specific configuration file are reported here and divided in three sheets: values, errors and statistical checks. The values sheet will contain the consistency checks or the comparison between results depending on the number of libraries that are being post-processed. The errors sheet is applicable only when a single set of results (i.e. coming from a single library) is being post-processed and reports the statistical error associated with the results presented in the values sheet. Finally, the statistical checks sheet recaps, for each tally, if the 10 statistical checks performed by MCNP have been passed or not. Thanks to Excel conditional formatting capabilities, inconsistencies or mistakes are highlighted using a color code. This allows to examine a great amount of data, while still being able to spot possible errors or inconsistencies rather easily. Examples of extracts of these MS Excel outputs can be found in Appendix D and a sample is shown in Fig. 2.12.

The last type of JADE output are the plots atlas. These consist of collections of indexed plots that allow for a more visual representation and comparison of the results. Such atlas is produced both in MS Word and PDF formats. A gallery of example plots can be found in Appendix E.

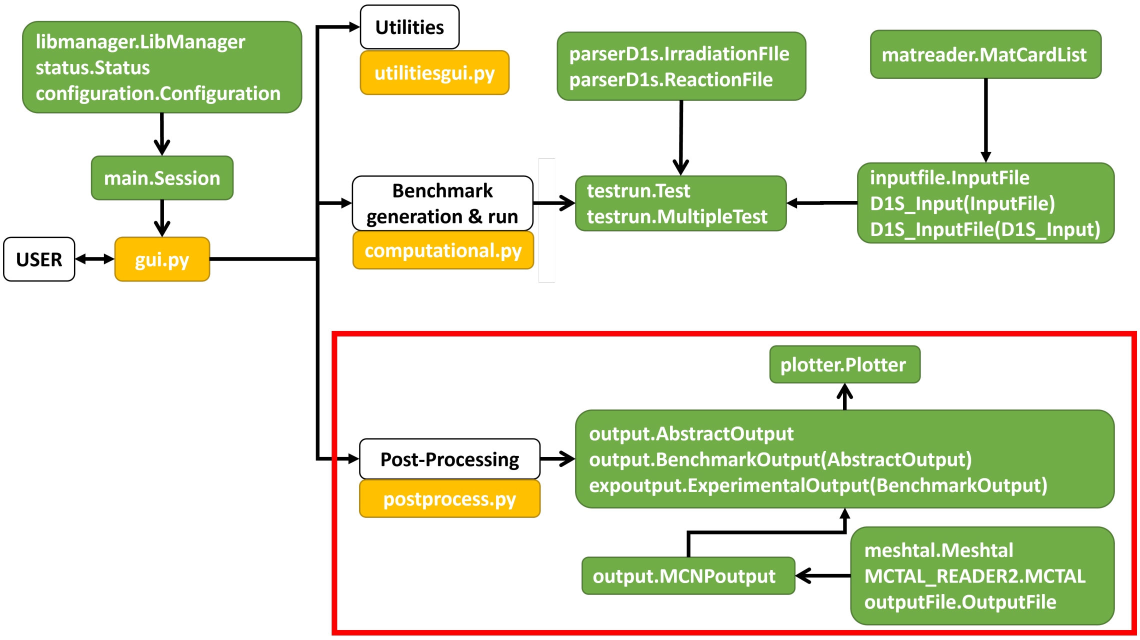


Figure 2.13: Scheme of JADE OOP implementation. Focus on post-processing

OOP implementation

Fig. 2.13 highlights the portion of JADE OOP scheme which is related to post-processing. The *postprocess.py* module contains the functions called by the *gui.py* one which are responsible for carrying out the post-processing (single and comparison) of the libraries requested by the user. These functions have the objective of correctly initialize a BenchmarkOutput (or one of its children) for each one of the benchmarks that need to be post-processed. It could be said that these objects are the core of JADE functioning. Indeed, here is defined all the logic that transform the vast amount of raw data coming from the simulations into a more meaningful and accessible format. The raw data is parsed by the Meshtal, MCTAL and OutputFile objects which are parsers for respectively the *.meshtal*, *.m* and *.o* MCNP output files. These are coordinated by the MCNPOutput object, which also have the duty to re-organize the parsed raw data in a table-like structure.

The BenchmarkOutput object can handle all possible computational benchmarks with some degree of customization coming from the specific benchmark configuration files. All experimental benchmarks instead, need a dedicated child of the ExperimentalOutput (which is itself a child of the BenchmarkOutput) to be defined. This has been already done for all experimental benchmarks that are part of JADE suite, while more detailed instructions on how to proceed to define a new one can be found in JADE documentation [2]. The dumping of the post-processed data in Excel is done through the ExcelOutputSheet helper object (contained in the *output.py* module), while the plots are produced through the Plotter object.

2.2.5 Utilities

High-level description

As anticipated before, during the developing of JADE, some stand-alone tools were identified and exposed for direct usage. These additional utilities include:

- the capabilty to print to video the available libraries included in the XSDIR file of the user;
- the capability to modify the suffix associated to a specific library modifying the ACE files;
- the capability of creating material mixtures in MCNP material card format starting from materials defined in an existing MCNP input;
- the capability to 'translate' existing MCNP material cards to different nuclear data libraries;
- the capability to print a detailed summary of the material compositions included in an existing MCNP input;
- the capability to switch an existing MCNP material card between atom or mass fraction format;
- the capability to produce D1S reaction files where all the possible reactions included in a specific D1S activation libraries are listed.

OOP implementation

All these utilities are handled from a dedicated function contained in the *utilitiesgui.py* module which is called by the user through the 'GUI'.

2.2.6 Automatic testing

There are many ways to test a software application. The best way to do it may depend on a number of factor such as the specific application, the dimension of the software, the number of users, the number of developers and so on. The most common way to test software is a functional and manual one, i.e., the developer introduces a new features, runs the code and checks that the obtained output is the expected one. Subsequently, another feature is added and the process is repeated. This simple way to approach the problem of code testing tends to work better when:

- only one or a few developers are involved;
- the complexity of the code is fairly low;
- the software outputs are limited and can be easily explored;
- the software is used mostly "internally" by people that have some knowledge about it and are less likely to break it with weird requests.

The further the development of a new tool moves away from these assumptions, the higher is the need to complement this functional testing with something more structured and automated, which is exactly what happened for JADE following the open-source release. The choice fell upon "unit testing" which is commonly used in software development and consists of stress-testing single units of code, i.e., the minimal portion of the software that have an autonomous operation like single classes or methods. These set of tests are usually defined with the help of specific libraries and are automated in order to be executed frequently. This approach guarantees a series of advantages:

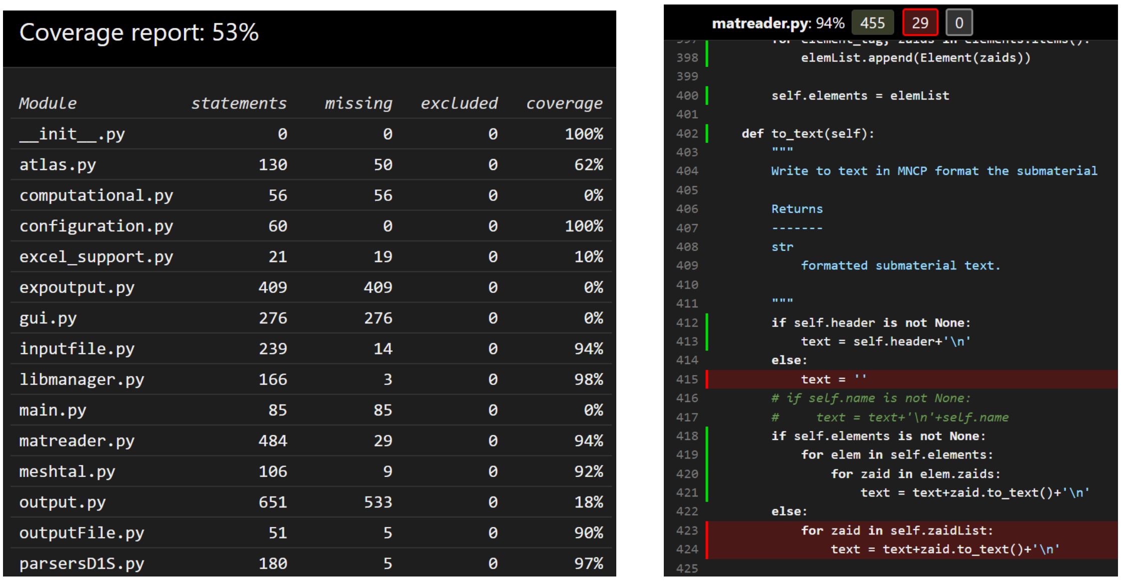
- it is easier to accept new features from new developers because the risk of them breaking the code will be way lower if it passed the tests;
- it generally improves the reliability of the code;
- it enhances the stability of the code through the years, since, when a bug is found, it is good practice to add a specific test able to catch it and ensure that it will never present itself again in the future;
- it generally improves the quality of the code since unit testing forces developers to write code in a more schematic and clear way (retrofitting unit testing to code already written, on the contrary, can cause a bit of an headache);
- if done correctly, it can actually speed up the development of new features when the complexity of the code is high due to a more easy identification of the issues (i.e. the failed tests).

Unit testing is also a key component of the so-called Continuous Integration (CI) which is needed when many developers start to collaborate to the same project. In order to do so, each of them will work on a local copy of the code which will gradually become less representative of the master repository due to the constant modifications applied by others. If a single developer waits too much before committing his modifications to the master repository, the main code could have become so different that trying to merge a commitment with success (i.e. without conflicts) could become more time consuming than just restart the development of the new feature from scratch. This situation is usually referred to as "integration hell" or "merge hell". In short, continuous integration consists of committing your code modifications fast and often, trying to avoid at all costs integration hell and save effort and time. Tools like GitHub are instrumental to the diffusion of the CI philosophy thanks to its many collaborative features and robust version control. Nevertheless, key principles of the process are considered to be the automatic build and automatic testing of the code which need to be performed each time the master code is updated.

Pytest and coverage

Two python packages have been used for the implementation of unit testing in JADE:

- pytest [72], which is the actual package that helps with the definition and automatic run of tests;



Coverage report: 53%

Module	statements	missing	excluded	coverage
__init__.py	0	0	0	100%
atlas.py	130	50	0	62%
computational.py	56	56	0	0%
configuration.py	60	0	0	100%
excel_support.py	21	19	0	10%
expoutput.py	409	409	0	0%
gui.py	276	276	0	0%
inputfile.py	239	14	0	94%
libmanager.py	166	3	0	98%
main.py	85	85	0	0%
matreader.py	484	29	0	94%
meshtal.py	106	9	0	92%
output.py	651	533	0	18%
outputFile.py	51	5	0	90%
parsersD1S.py	180	5	0	97%

(a) Index

matreader.py: 94% 455 29 0

```
            for element_tag, zaids in elements.items():
                elemList.append(Element(zaids))

        self.elements = elemList

    def to_text(self):
        """
        Write to text in MNCP format the submaterial

        Returns
        -------
        str
            formatted submaterial text.

        """
        if self.header is not None:
            text = self.header+'\n'
        else:
            text = ''
        # if self.name is not None:
        #     text = text+'\n'+self.name
        if self.elements is not None:
            for elem in self.elements:
                for zaid in elem.zaids:
                    text = text+zaid.to_text()+'\n'
        else:
            for zaid in self.zaidList:
                text = text+zaid.to_text()+'\n'
```

(b) Single module detail

Figure 2.14: Example of a coverage html report

- coverage [73], which is a helper package that register which lines of code are actually executed during the tests executions in order to estimate their coverage, i.e., the ratio between the lines executed with respect to the total number of lines.

Commercial-grade software usually never goes below 100% coverage. Nevertheless, full coverage of a code should always be seen as a necessary condition and not a sufficient one. Indeed, one thing is to execute a line of code during a test, another is to guarantee that this will never break whatever input is provided. It is virtually impossible to foresee in advance (hence test) all the possible scenarios in which a method or a class can be used. Unfortunately, to reach 100% coverage in JADE was not feasible due to the lack of manpower and to the Microsoft Office dependencies. The test suite defined through pytest can be run at any time by users and it is automatically run every time there is a push to the main branch of JADE in GitHub thanks to GitHub workflows. This grants Continuous Integration (CI) and prevents future JADE contributors to push changes to the code that would break its core functionalities.

If pytest defines and runs the test, it is coverage that tracks which part of the code is executed and then creates an html tree that summarizes a number of useful information for the developer. A main index is produced where a summary of the coverage results is reported together with links that allow to access the coverage results for each single python module. Fig. 2.14 displays screenshots of such coverage reports. One of the best feature is that, in the single module report, all lines that were not executed during the tests are highlighted in red in order for the developer to quickly individuate them.

Appendix A contains an extract of the coverage output for JADE complemented with brief description of the modules and eventual notes on their coverage. As it can be seen, the majority of the core modules of JADE has been covered for more than 90% with the notable exception of the three modules related to the generation of post-processing which are “expoutput.py”, “output.py” and “sphereoutput.py” that have not been covered at all. The result is a total coverage of only 53% of JADE lines of code. For the modules that are actually tested, 100% was not reached due to the fact that covering all possible exceptions that the code is programmed to raise during quite rare events would have been inefficient in terms of ratio between increased reliability of the code and manpower spent. With regard to the untested modules, instead, the issue relies on the fact that they are highly dependent from the Microsoft Office suite dependency (i.e., Excel and Word). This generates an issue because these specific modules could not run in the virtual environment created by GitHub to run the tests on. Theoretically, it would still be possible to define tests able to run only on local installations, but this would deviate from the objective of having CI. Moreover, it is expected for JADE to eliminate this dependency in the mid-future and to significantly change how the post-processing is produced. When this will happen, it is strongly recommended to develop suitable testing in parallel for this part of the code.

2.3 Benchmarks description

JADE’s heart is clearly its benchmarks suite. A number of computational and experimental benchmarks have been implemented and are part of the default JADE V&V process. For users that have access to D1S-UNED (or equivalent MCNP modifications), ‘activation’ benchmarks are also provided. This kind of benchmarks allow to test the capabilities of the libraries to correctly evaluate dose rates, especially Shut-Down Dose Rates (SDDR), which are key quantities for fusion reactors like ITER. The following sections describe in more detail all the default benchmarks included in JADE’s suite. It can be noticed how in all currently implemented benchmarks only neutrons and secondary photons are transported, while charged particles are not considered. This is surely a line of work where JADE could expand in the future but it was not considered a priority. In fact, the main drivers for commercial nuclear fusion power plant designs (in terms of what can be computed through particles transport analyses) are DPA (caused by neutrons), nuclear heating, photon heating and SDDR.

2.3.1 Sphere Leakage

The Sphere Leakage benchmark is arguably the most important benchmark included in JADE. Indeed, it allows to individually test each isotope of a nuclear data library under assessment plus some example materials that are commonly found in the ITER reactor. These are:

- Water;
- Ordinary Concrete;
- Boron Carbide;

- SS316L(N)-IG;
- Natural Silicon;
- Polyethylene (non-borated).

A different MCNP input will be generated and run for every zaid in the library and every material and many quantities of interest such as neutron flux, secondary photons flux, gas production and nuclear heating are tallied, often in fine discrete energy ranges.

Geometry and run parameters

The Sphere Leakage geometry consists of three concentric spheres. The inner one is void and has a radius of 5 cm. At the center of this sphere is located a neutron point source which is isotropic and has a uniform probability of energy emission between 0 and 14 MeV. 14 MeV (more precisely 14.1 MeV) is the typical neutron emission energy observed in the D-T fusion reaction as discussed in Section 1.2.1, which is of great interest for the fusion community. The second sphere has a radius of 50 cm and it is composed entirely by the single isotope or the typical ITER material under assessment. Finally, the last 60 cm radius sphere acts as a graveyard where particles importance is set to zero and the boundary of the model is defined. A representation of such geometry is shown in Fig. 2.15.

During the development of this benchmark, two key parameters needed to be defined: the sphere density and the MCNP STOP card (see Section 1.1.4). At first the possibility to impose a single density of the sphere equal for all materials and isotopes was considered, but quickly discarded. Instead, in order to keep some kind of physical meaning of the results, it was decided to compute the default densities to NTP (Normal Temperature and Pressure) conditions, which are defined at 20 °C and 101325 Pa (1 atm). Even if these values work quite well with solids, they cause gases to perform poorly in terms of tally scoring. This happens due to the substantially lower density in NTP conditions for gases when compared to solids, resulting in too few interactions of the neutrons and secondary photons with the material. This has been proven to be especially true for hydrogen and helium, leading to the choice of selecting their liquid phase density instead of the gas one. Another issue was encountered when simulating fissile isotopes like U235. Indeed, a 1 m diameter sphere containing a pure fissile isotopes at NTP density is very much super-critical and the high number of secondary particles (i.e. other neutrons) that are produced in this way were causing the simulations to fail due to memory limitations. For this reason, the density of these fissile isotopes was imposed equal to 1 g/cc as if an aerosol phase was considered.

Finally, the STOP card parameters for each isotope were optimized. MCNP allows to stop a simulation based either on:

- the precision reached in a specified tally;
- the number of histories (NPS);

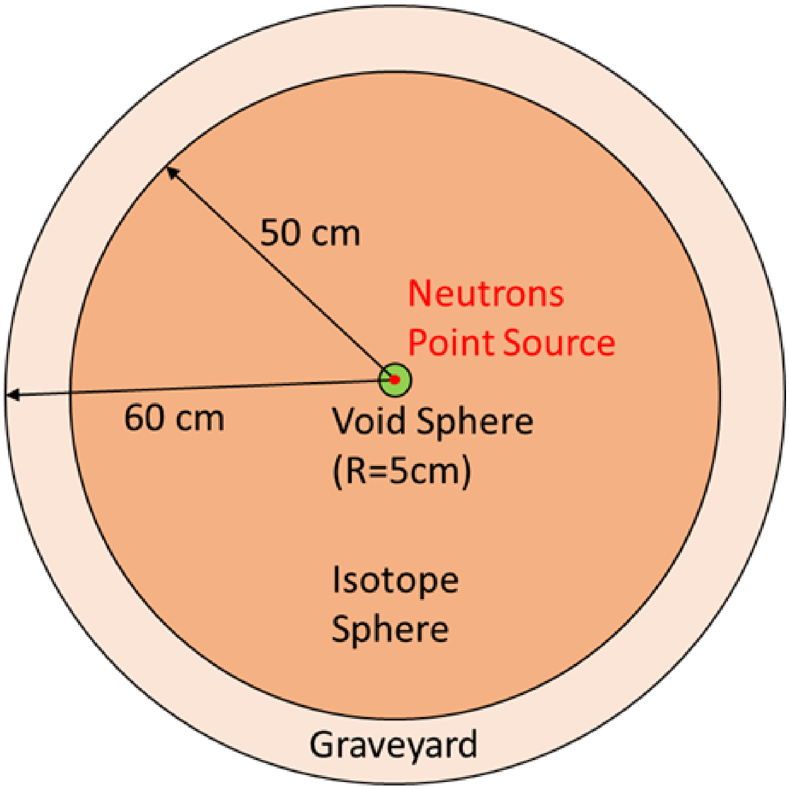


Figure 2.15: Sphere Leakage benchmark geometry representation

- the total elapsed computer time (i.e. the sum of computer time used by all CPUs).

The optimization of such parameters for each element was done through trial and error with the aim of finding a good balance between computational cost and precise enough results. These parameters are provided by default in JADE, but the user may modify them, as well as density values, through the benchmark-specific configuration files.

Tallies

Both the transport of neutrons and of secondary photons are active and photons cut-off energy is left to the default value of 1 KeV. The following MCNP tallies are defined:

Tally n. 2 Fine neutron flux at the external surface of the filled sphere. The flux is binned in energy using the Vitamin-J [74] 175 energy group structure.

Tally n. 12 Coarse neutron flux at the external surface of the filled sphere. The flux is binned in 5 energy groups: 1e-6, 0.1, 1, 10 and 20 MeV.

Tally n. 32 Fine photon flux at the external surface of the filled sphere. The flux is binned in energy using the 24 group structure described in the FISPACT manual [27].

Tally n. 22 Coarse photon flux at the external surface of the filled sphere. The flux is binned in 5 energy groups: 0.01, 0.1, 1, 10 and 20 MeV.

Tally n. 4 Neutron heating computed in the filled sphere (F4+FM strategy).

Tally n. 44 Photon heating computed in the filled sphere (F4+FM strategy).

Tally n. 6 Neutron heating computed in the filled sphere (F6 strategy).

Tally n. 16 Photon heating computed in the filled sphere (F6 strategy).

Tally n. 14 Helium (He) ppm production in the filled sphere.

Tally n. 24 Tritium (T) ppm production in the filled sphere.

Tally n. 34 Displacement Per Atom (DPA) production in the filled sphere.

More details on the MCNP tally definition and especially on the difference between the heating computation using the F6 or the F4+FM strategy can be found in Section 1.1.4.

2.3.2 ITER 1D

The ITER 1D benchmark [75] developed by Sawan M. is a popular 1-Dimensional neutronic model used for nuclear data benchmarking in the fusion community. This consists of a simple but realistic model of the ITER TOKAMAK where the inboard and outboard portion of the machine and the plasma region are modelled by means of simple concentric cylindrical surfaces.

Geometry and run parameters

As shown in Fig. 2.16 the benchmark geometry is uniquely composed by concentric cylindrical surfaces. A detailed description of the different layers is reported in Fig. 2.17. The plasma region includes a 14.1 MeV isotropic neutron source (characteristic of Deuterium-Tritium fusion reaction).

Tallies

Many quantities are tallied in the ITER 1D benchmark, the following is a thorough description of them.

Tally n. 4 Neutron flux [$\#/cm^2$] (binned in Vitamin-J 175 energy groups) in 97 different MCNP cells located across the radial direction.

Tally n. 204 Total neutron flux [$\#/cm^2$] at the same locations as Tally n. 4.

Tally n. 14 Photon flux [$\#/cm^2$] (binned in energy) at the same locations as Tally n. 4. The energy bins limits are 0.1, 1, 5, 10 and 20.

Tally n. 214 Total photon flux [$\#/cm^2$] at the same locations as Tally n. 4.

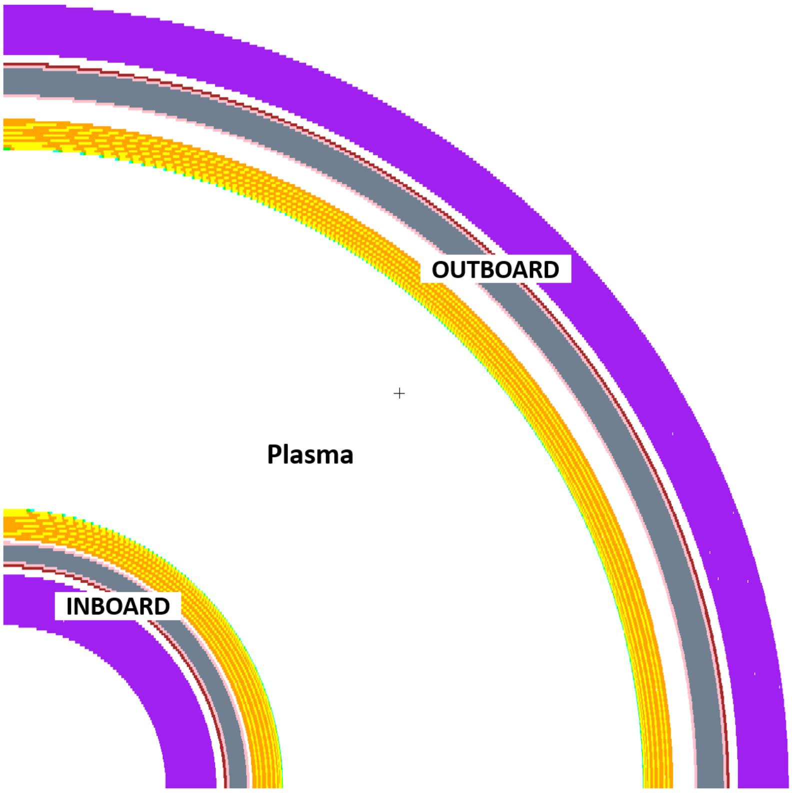


Figure 2.16: Normal section view of the ITER 1D benchmark MCNP geometry (one quarter)

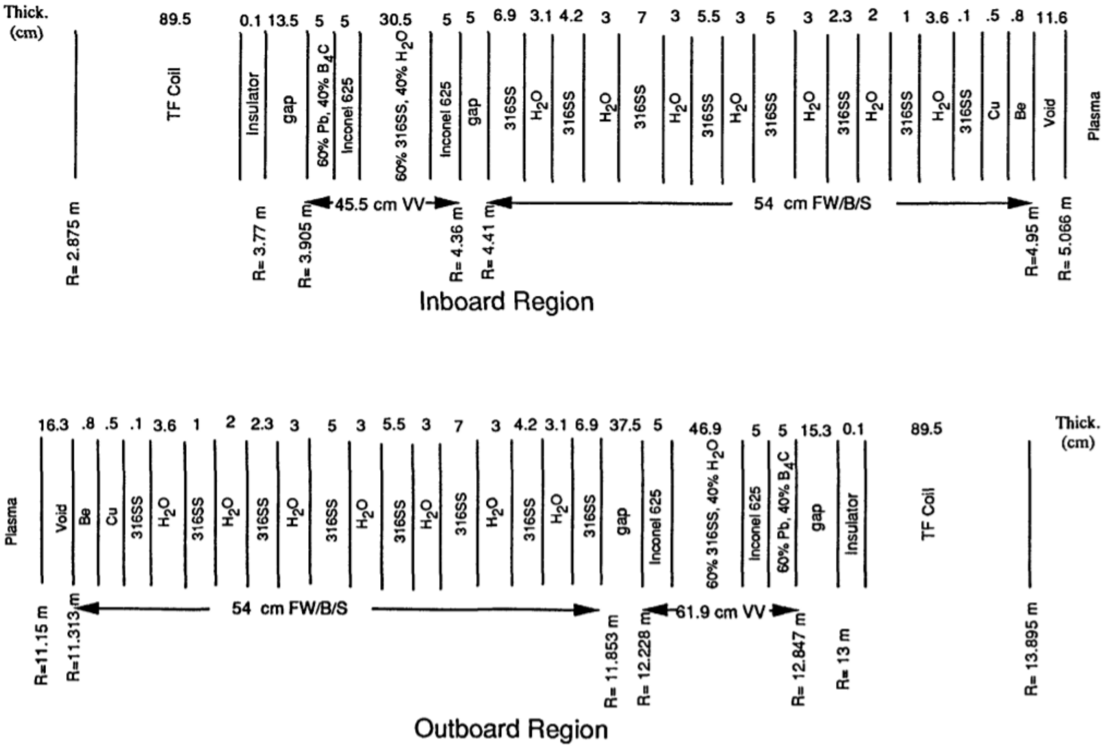


Figure 2.17: Description of the layers composing the ITER 1D benchmark

Tally n.6 Total nuclear heating [W/g], i.e., neutron plus photon heating at the same locations as Tally n. 4.

Tally n. 16 Neutron heating [W/g] at the same locations as Tally n. 4.

Tally n. 26 Photon heating [W/g] at the same locations as Tally n. 4.

Tally n. 34 Helium production in steel.

Tally n. 44 Hydrogen production in steel.

Tally n. 54 Tritium production in steel.

Tally n. 64 Displacement per atom (DPA) in Cu.

Tally n. 74 Helium production in CuBeNi.

Tally n. 84 Hydrogen production in CuBeNi.

Tally n. 94 Tritium production in CuBeNi.

Tally n. 104 DPA in Nickel.

Tally n. 114 Helium production in Inconel.

Tally n. 124 Hydrogen production in Inconel.

Tally n. 134 Tritium production in Inconel.

Tally n. 144 Helium production in Be.

Tally n. 154 Hydrogen production in Inconel.

Tally n. 164 Tritium production in Inconel.

Tally n. 174 Fast (E>0.1 MeV) neutron fluence at magnets.

2.3.3 Test Blanket Module

Tritium and Deuterium are the two main ingredients of the fusion reaction that are foreseen to be tested in ITER and that should guarantee sustainable energy production in DEMO. Deuterium is abundant in normal sea water and can be extracted relatively easily. Unfortunately, the same cannot be said for Tritium, whose short half life means that is not readily available in nature. For this reason, in order to be sustainable, TOKAMAKs will need to be able to produce, or "breed" all the tritium needed for their fusion reactions. Tritium breeding is not foreseen for the ITER power plant and the open point will need to be closed by DEMO instead. Nevertheless, ITER is a unique opportunity to test various breeding concepts and find out what will be the best solution to implement in DEMO, leading to the creation of the Test Blanket Module (TBM) project. These are prototypes of blanket sections (and their ancillaries) which have the capability to breed and store tritium for later use.

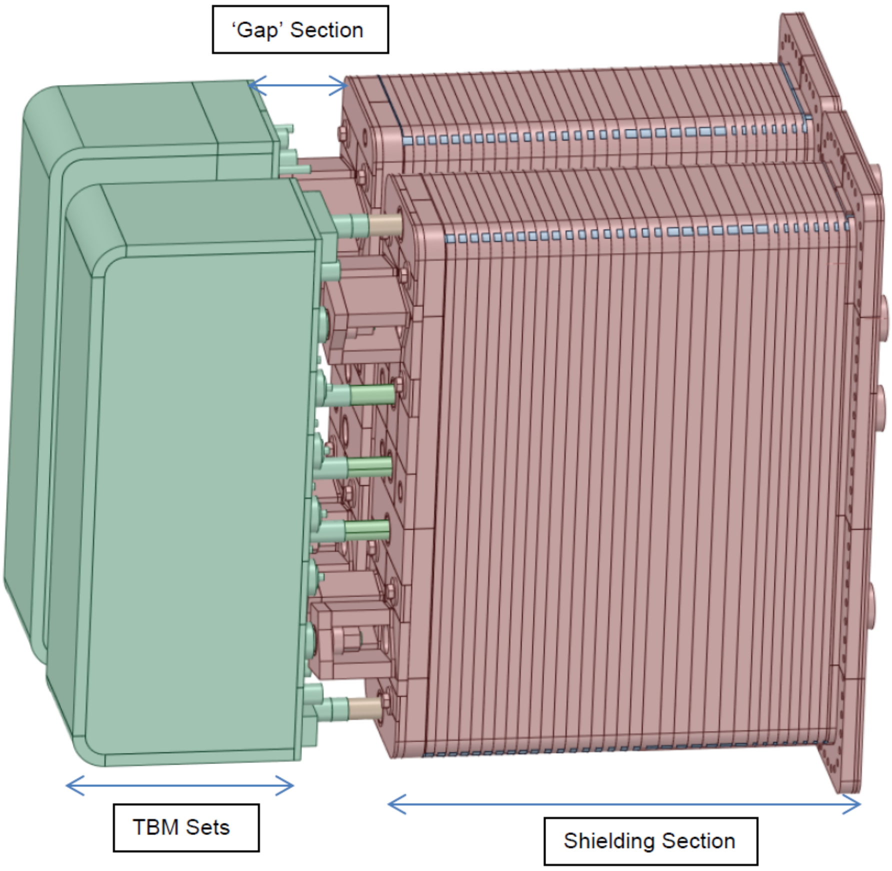


Figure 2.18: TBM assembly

Building on the historical ITER 1D model (presented in Section 2.3.2), in 2020, two additional benchmark were generated [76] by the F4E neutronic team which had a specific focus on the ITER TBM experiments. The ITER 1D original model was focused on shielding application and did not feature any port in the outboard region. On the contrary, in the new benchmarks, the outboard region was substituted with 1D models of the two proposed European concepts for the TBM: the Helium Cooled Pebbled bed (HCPB) and the Water Cooled Lithium Lead (WCLL).

Geometry and run parameters

The benchmarks are focused on the TBM set (i.e. the actual blanket module) and on the shielding section that can be found behind it. Fig. 2.18 displays a CAD image of such general architecture. While the shield section does not really change between the two different TBM concepts, the TBM sets do. Fig. 2.19 shows a section of the HCPB TBM Set while Fig. 2.20 of the WCLL one. TBM Set and Shielding section are the only regions that are modified in the ITER 1D model described in Section 2.3.2. A more detailed description of the different layers composing the geometry can be found in Appendix B.1.

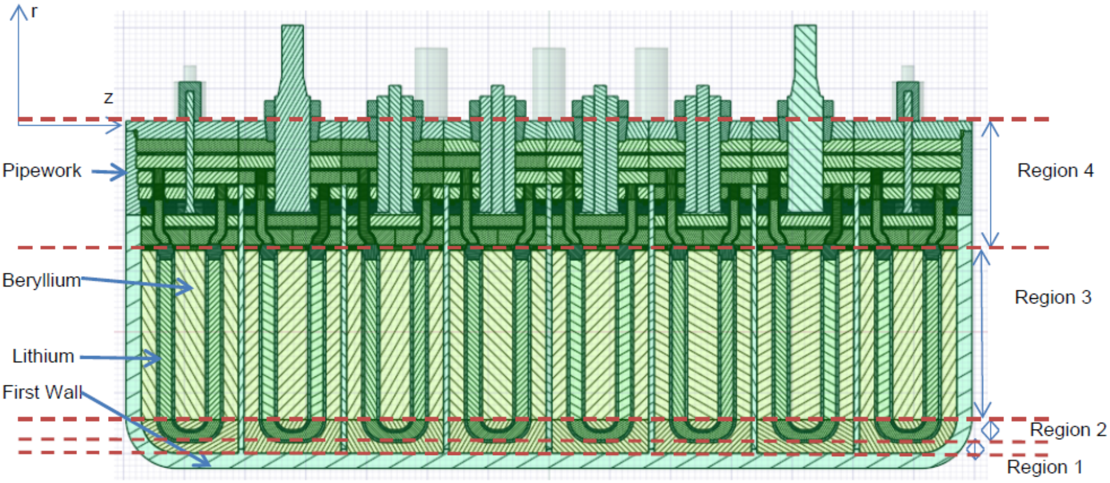


Figure 2.19: HCPB TBM Set section

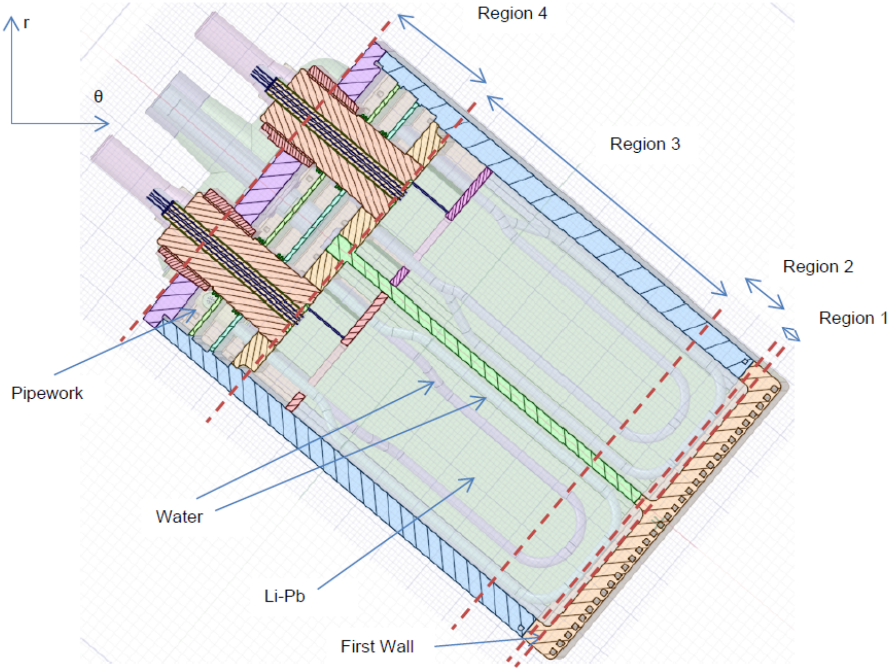


Figure 2.20: WCLL TBM Set section

Tallies

All Tritium production tallies that were defined for the ITER 1D benchmark (see Section 2.3.2 were retained also in the TBMs ones. Additionally, a 1-dimensional FMESH was placed on the outboard region (from R=830 up to R=1084.2) composed by 2000 bins. The following quantities were tallied on such grid:

Tally n. 214 Neutron heating [$MeV/cm^3/n_s$].

Tally n. 224 Photon heating [$MeV/cm^3/n_s$].

Tally n. 234 Tritium production [$atoms/cm^3/n_s$].

Tally n. 244 Neutron flux [$\#/cm^3/n_s$].

Tally n. 254 Photon flux [$\#/cm^3/n_s$].

2.3.4 C-Model

During the long life of the ITER project, many neutronics models have been generated to represent the TOKAMAK machine. These are used to conduct neutronic analyses on the reactor in order to investigate many direct and indirect effects induced by neutrons like heat generation, particle generation, DPA, dose rate, etc. C-Model [77] is an extremely detailed MCNP input of a 40° sector of ITER TOKAMAK. It was the most complete neutronic model available for the ITER machine until 2021, when E-lite was released [78] which is a full 360° model of ITER that was conceived to overcome some limitation encountered using the C-Model for specific application. Nevertheless, since E-lite is an extremely heavy model, C-model is still considered the reference basic model of the ITER TOKAMAK for neutronic analyses.

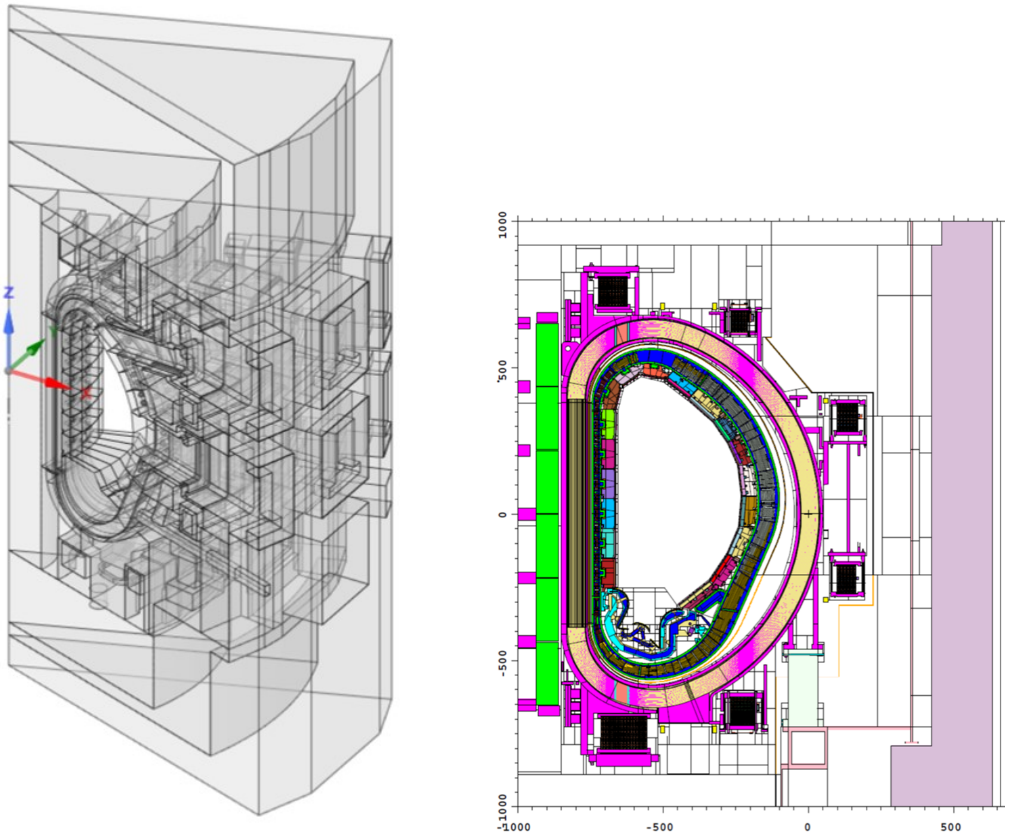
Since the model is property of the ITER project, the MCNP input template of this benchmark cannot be freely distributed with together with the JADE source code.

Geometry and run parameters

Due to its complexity, a thorough description of the C-Model benchmark geometry is considered out of the scope of this work and can be found, instead, in a dedicated F4E report [79]. A global view of the different universes composing the model and a poloidal section of the MCNP geometry are reported in Fig. 2.21.

Tallies

The standard tallies proposed in [79] have been used. They include neutron current, photon current and nuclear heating at different locations.



(a) Universe view

(b) MCNP section plot

Figure 2.21: C-Model benchmark geometry

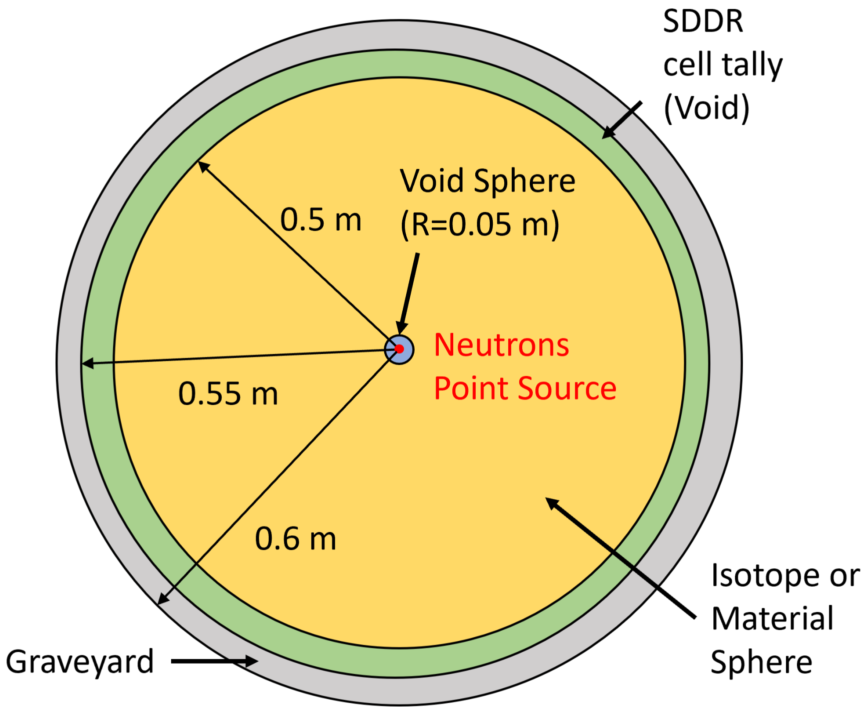


Figure 2.22: Schematic view of the Sphere SDDR model

2.3.5 Sphere SDDR

The Sphere SDDR benchmark is a variation of the Sphere Leakage benchmark described in Section 2.3.1 which is focused on isotopes activation and dose rate measurement. Once again, these kind of benchmarks allows to test all available isotopes in the library under assessment (this time being a D1S activation library) together with a few typical ITER materials. In particular, each single reaction channel (MT) of every isotope will be tested separately while, for the typical materials, all possible reactions foreseen by the library will be considered.

Geometry and run parameters

The geometry of the Sphere SDDR benchmarks, as it can be observed in Fig. 2.22, is practically the same as the one described in Section 2.3.1. The only difference is that externally to the filled sphere, a void spherical shell has been defined having a 10 cm radial thickness. This is the cell used to tally the shut down dose rate.

Similarly to what was done for the Sphere Leakage benchmark, the user have control on the densities to be applied for each element and material (default is set to NTP conditions with few exceptions) and control on the STOP parameters to be used.

SDDR parameters

The cool-down times that have been considered are $0s$, $2.7h$, $24h$, $11.6d$, $30d$ and $10y$. For the isotopes simulations, since only one reaction is considered, relative comparisons at different cool-down times will not lead to different results, hence,

Source Intensity [n/s]	Δt irr.	Multiplicity
1.0714E+17	2 y	1
8.2500E+17	10 y	1
0	9 m	1
1.6667E+18	15 m	1
0	3290 s	17
2.0000E+19	400 s	
0	3290 s	4
2.8000E+19	400 s	

Table 2.1: Irradiation schedule in SA2 mode at the ITER Blanket

during post-processing operations, only the results at $0s$ are elaborated. This does not apply to materials simulations, where all possible reactions are included and the results at the various cool-down times will be different. The irradiation schedule considered for the Sphere SDDR benchmark is reported in Table 2.1. This represents an actual equivalent irradiation scenario foreseen for ITER blanket (mode SA2). The Multiplicity column indicates how many times the irradiation step defined by the previous two columns is repeated in the irradiation schedule.

As previously discussed, the irradiation file and reaction file provided together with the MCNP input file are generated in two different ways depending on if the simulation is conducted on a single isotope or on a typical ITER material. In the first case, a single reaction is considered and the irradiation file will only contain the daughter of such irradiation. In the second case, all possible reactions that are available in the library and that can be originated in the material will be included. The irradiation file will be then generated accordingly.

Tallies

All neutron and photon related tallies scored in Section 2.3.1 have also been imported in the Sphere SDDR benchmark. For photons, the time binning necessary to cover all the cool-down times of interest have been added. Tally n. 104 have been also defined to tally the shut down dose rate at all cool-down times in the additional spherical shell added for this specific purpose [Sv/h].

2.3.6 ITER Cylinder SDDR

The ITER Cylinder SDDR is a very popular computational benchmark for SDDR computation in ITER [80][81] since it features dimensions, materials and streaming characteristics of a typical ITER equatorial port.

Geometry and run parameters

The ITER Cylinder SDDR is a simple yet effective benchmark. The model is composed by a 550 cm long, hollowed steel cylinder with internal and external radius

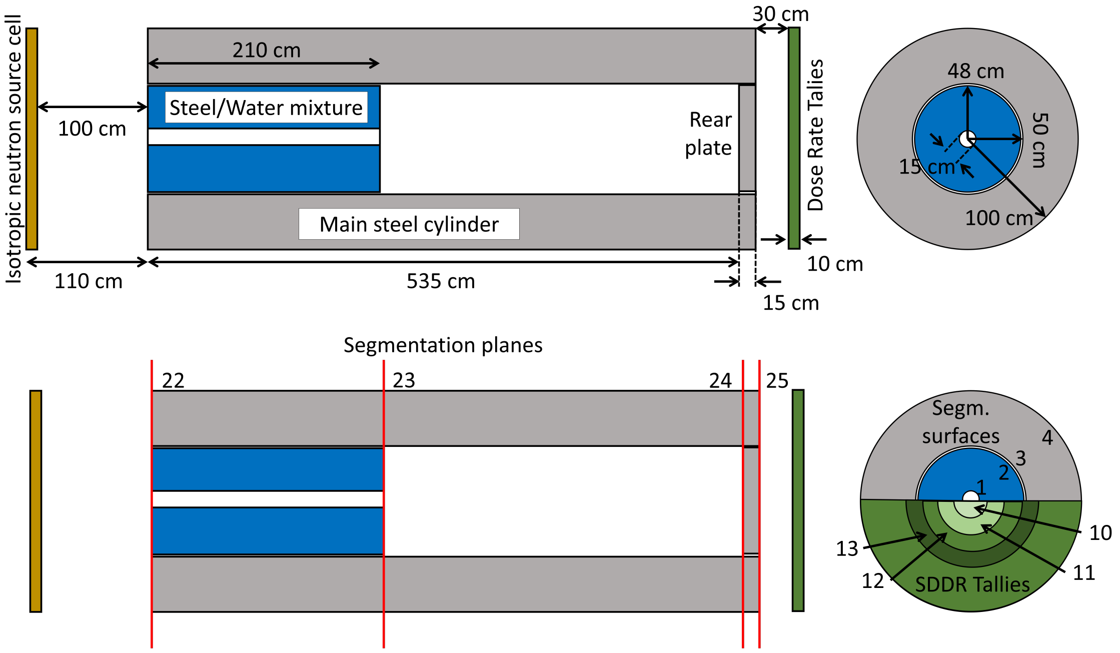


Figure 2.23: ITER Cylinder SDDR benchmark geometry visualization

respectively equal to 50 cm and 100 cm. The rear part of the cylinder is closed with a steel disk plate of 48 cm radius and 15 cm thick. The inner front part of the cylinder is filled with a smaller cylinder made of a water-steel mixture. This internal 48 cm radius cylinder is 210 cm long and features a central 15 cm diameter cylindrical hole. As it can be deducted from the given measures, a 2 cm gap is left between the main external hollow cylinder and its internal components.

A volumetric and isotropic neutron source is also defined. The volume of emission is a disk aligned with the front part of the cylinder assembly and positioned at a distance of 100 cm. The volume of the disk is 10 cm thick and has a radius equal to 100 cm. Fig. 2.23 gives a complete overview the described geometrical model.

A series of cells and surfaces is also defined for tallying purposes. The shutdown dose rate due to the activation of the assembly is evaluated in cell tallies located 30 cm past the end of the rear plate. The tally cells consist of concentric (hollow) disks which are 10 cm thick and are characterized from the following radii:

- from 0 cm to 15 cm (MCNP cell n. 10);
- from 15 cm to 30 cm (MCNP cell n. 11);
- from 30 cm to 45 cm (MCNP cell n. 12);
- from 45 cm to 60 cm (MCNP cell n. 13).

For flux tallying purposes, instead, the following cylindrical surfaces have been defined:

- n. 1, coincident with the external surface of the central hole;
- n. 2, coincident with the external surface of the water/steel cylinder;

- n. 3, coincident with the internal surface of the main steel cylinder;
- n. 4, coincident with the external surface of the main steel cylinder;

and the following planes orthogonal to the cylinder length:

- n. 22, coincident with the front of the assembly;
- n. 23, coincident with the rear of the water/steel cylinder;
- n. 24, coincident with the front of the rear plate;
- n. 25, coincident with the rear of the assembly.

All the described surfaces and cells are also reported in Fig. 2.23.

SDDR parameters

The irradiation schedule considered for ITER Cylinder SDDR benchmark is the same as the one of the Sphere SDDR benchmark reported in Table 2.1. Two different cool-down times were considered in the photon tallies: 0s and 1e6 s (approx. 11.5 days). That is, these are the time interval waited after the irradiation phase has finished before tallying the SDDR and the photon flux. The possible reactions allowed during the simulation are listed in Table 2.2.

Tallies

Neutron flux, (decay) gamma flux and SDDR are the only tallied quantities. The following is a description of the tallies defined in the benchmark:

Tally n. 202 Neutron flux per energy bin [$\#/cm^2/s$]. The flux is tallied in 16 energy bins ranging between 1E-10 MeV to 20 MeV. The flux is also binned geometrically using all surfaces described in Section 2.3.6.

Tally n. 242 Total neutron flux [$\#/cm^2/s$]. Same as Tally n. 202 but without the energy binning.

Tally n. 14 Gamma flux per energy bin in cell 10 [$\#/cm^2/s$]. The flux is tallied in 16 energy bins ranging from 0.1 MeV to 20 MeV. The flux is tallied at both cool-down times.

Tally n. 34 Gamma flux per energy bin in cell 11 [$\#/cm^2/s$]. The flux is tallied in 16 energy bins ranging from 0.1 MeV to 20 MeV. The flux is tallied at both cool-down times.

Tally n. 44 Gamma flux per energy bin in cell 12 [$\#/cm^2/s$]. The flux is tallied in 16 energy bins ranging from 0.1 MeV to 20 MeV. The flux is tallied at both cool-down times.

Tally n. 54 Gamma flux per energy bin in cell 13 [$\#/cm^2/s$]. The flux is tallied in 16 energy bins ranging from 0.1 MeV to 20 MeV. The flux is tallied at both cool-down times.

Parent	Daughter
Cr50	Cr51
Cr52	Cr51
Mn55	Mn54
Fe54	Mn54
Fe54	Cr51
Fe56	Mn54
Fe58	Fe59
Co59	Co58
Co59	Co60
Co59	Fe59
Ni58	Co58
Ni60	Co60
Ni61	Co60
Ni61	Co60
Ni62	Fe59
Cu63	Cu62
Cu63	Co60
Cu65	Cu66
Ta181	Ta182
W182	Ta182
W186	W187

Table 2.2: List of possible reactions considered during the ITER Cylinder SDDR benchmark

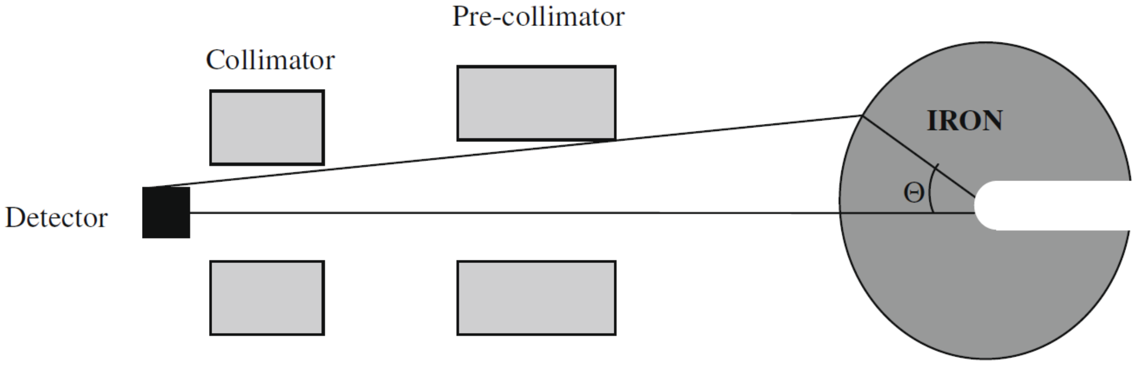


Figure 2.24: OKTAVIAN (Iron) TOF experiment layout

Tally n. 74 Total gamma flux [$\#/cm^2/s$]. The flux is tallied only by cell (i.e. 10, 11, 12 and 13).

Tally n. 124 SSDR behind the plate [Sv/h]. The SDDR is computed at all cell tallies (i.e. 10, 11, 12 and 13) and at both cool-down times.

2.3.7 OKTAVIAN

OKTAVIAN is an experimental facility located at the Osaka University which has been operative since 1981. It consists of an intense deuterium-tritium (D-T) neutron source (up to $3 \cdot 10^{12}$ n/s) that has been used during the years for many experiments on "fusion like" neutron transport. Among them, many Time Of Flight (TOF) experiments were conducted [82] and their results have been introduced in SINBAD. These experiments consisted of placing the neutron source inside a sphere made entirely by a specific material of interest and measuring the leakage photon spectra exiting from such sphere with the use of detectors. The photon energy measure is performed indirectly measuring the time of flight, which is then converted into a velocity. The MCNP input models of some of these experiments and their associated experimental results are now freely available at CoNDERC [83] a new neutronic experiment open repository developed under the auspices of the IAEA Nuclear Data Section developed in order to incentive V&V activities on nuclear data. The OKTAVIAN inputs distributed in JADE are practically the same files provided in CoNDERC.

Geometry and run parameters

An accelerated deuteron beam is led through a narrow tube to the centre of a sphere (every time composed by a different material) where pulsed 14.1 MeV monochromatic neutrons were produced by the d-t fusion reaction. The source is regarded to be 14 MeV monochromatic. Neutron leakage current spectrum of neutrons was measured in "absolute values" by the time-of-flight technique between 10 keV and 14 MeV, about 9.5 m from the sphere centre. For additional details on the experiment, which is schematized in Fig. 2.24, the reader is referred to [82].

A complete list of the materials used in the Oktavian benchmarks can be found in Appendix B.2.

Tallies

Only two tallies are defined for each input:

Tally n. 21 Neutron leakage current $[\#/cm^2]$ per source particle. 134 energy bins were defined spanning from 0.1 MeV to 20.6 MeV;

Tally n. 41 Photon leakage current $[\#/cm^2]$ per source particle. 57 energy bins were defined spanning from 0.5 MeV to 10.5 MeV.

Since experimental results are provided as flux per unit lethargy, the tally results are manipulated as follows:

$$d\Phi_u = d\Phi/d(\log E) \tag{2.4}$$

2.3.8 Frascati Neutron Generator

The Frascati Neutron Generator (FNG) [84] is an experimental facility designed and built by ENEA (the Italian National Agency for New Technologies, Energy and Sustainable Economic Development) in Frascati, Italy. The installation is able to produce 14 MeV neutrons based on the $T(d,n)\alpha$ fusion reaction and it is able to produce up to $5 \cdot 10^{11}$ n/s.

One of the key experiments that have been conducted at the FNG is the neutron irradiation experiment, where a mock-up of the outer vacuum vessel region of ITER was irradiated by means of 14 MeV neutrons for a sufficiently long time in order to achieve activation levels similar to the ones that are expected to be reached at the ITER end of life. Two distinct irradiation campaigns were conducted in May and August 2000 and, among other things, the SDDR values after different cooling time intervals were measured. Many benchmarks activities have been performed using the experiment in the past [85][86][87], and the benchmark is also included in the SINBAD database (listed as fng_dose).

Geometry

In the FNG, a deuterium beam is accelerated up to 300 KeV by means of a linear electro-static tube towards a target rich in tritium generating a 14 MeV neutron source. These are the neutrons that were used to irradiate the experimental assembly which consisted of a block of stainless steel and water equivalent material (perspex) with total thickness of 71.4 cm, and a lateral size of 100 cm x 100 cm. A cavity was obtained within the block (12.6 cm in the beam direction, 11.98 cm high) behind a 22.47 cm thick shield. A void channel (2.7 cm inner diameter) was included in front of the cavity to study the effect of streaming paths in the bulk shield. A squared box was used to locate detectors inside the cavity, with 2 mm thick bottom and lateral walls. Measurements were taken in the cavity, during the irradiation and after shut-down, to obtain the local neutron flux, the decay gamma-ray spectra and

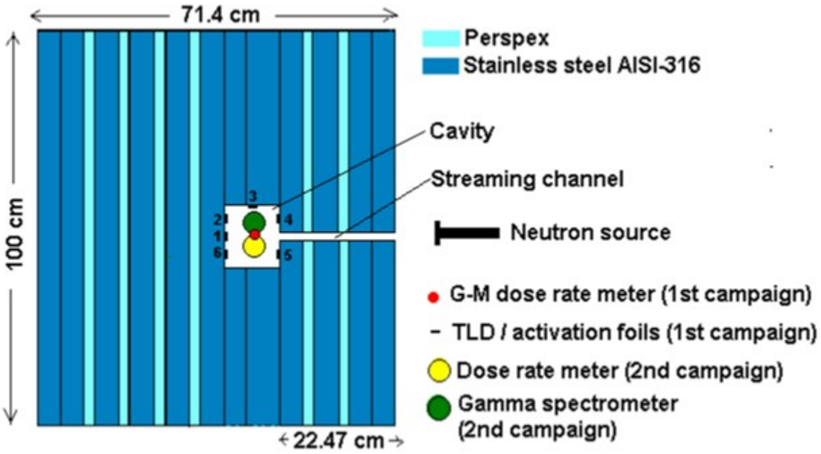

71.4 cm
100 cm
22.47 cm
Perspex
Stainless steel AISI-316
Cavity
Streaming channel
Neutron source
G-M dose rate meter (1st campaign)
TLD / activation foils (1st campaign)
Dose rate meter (2nd campaign)
Gamma spectrometer (2nd campaign)
1
2
3
4
5
6

Figure 2.25: Schematic view of the FNG irradiation experiment set up

the dose rates for different cooling times. A schematic view of the experiment set up is shown in Fig. 2.25.

JADE MCNP input template was built unifying inputs coming from the SINBAD database, ENEA-provided modifications and UNED-provided modifications. For this reason, it cannot be freely distributed together with the JADE source code. SINBAD provides two different MCNP inputs for the benchmark, one to be used during the neutron transport and the second one for dose rate calculation. This is because after the irradiation campaign were completed, some shielding material was removed in order to access the cavity where the detector was positioned, changing the geometry of the model. To reproduce this effect in a single D1S simulation, D1S-UNED provides convenient PMT cards that allow MCNP to consider different material depending on if it is simulating the neutron transport or the decay photons one. These cards provided by UNED were included in JADE's FNG input. Finally, both the UNED and SINBAD inputs used an external routine to define the neutron source. This method is rather inconvenient for JADE since these external routines need to be compiled, hence, an equivalent SDEF card provided by ENEA was used instead. It is likely though that ENEA and UNED neutron sources were not exactly equivalent.

SDDR Parameters

Table 2.3 and Table 2.4 describe the equivalent schedules considered for respectively the 1^{st} and 2^{nd} irradiation campaign conducted at the FNG. The experimentally measured SDDR values at different cooling times are reported in Table 2.5 for the 1^{st} irradiation campaign and in Table 2.6 for the second one. In order to determine which decay reactions to be followed during the simulation, a more general approach was defined with respect to the one used for the ITER Cylindrical SDDR benchmark (see Section 2.3.6). First of all, when simulating with the D1S approach, in order to reduce the computation time it is good practice to individuate the subset of decay isotopes which contribute the most to the dose rate. This subset will depend from the unirradiated material composition and the cool-down time that are considered. In

Δ_t [s]	Δ_t [min]	Neutron Intensity [n/s]
19440	324	2.32E+10
61680	1028	0
32940	549	2.87E+10
54840	914	0
15720	262	1.90E+10
6360	106	0
8940	149	1.36E+10

Table 2.3: Equivalent schedule of the 1^{st} FNG irradiation campaign

Δ_t [s]	Δ_t [min]	Neutron Intensity [n/s]
1748	29	3.04E+10
7820	130	4.28E+10
54140	902	0
22140	369	4.29E+10
900	15	0
3820	64	3.38E+10
420	7	0
140	2	2.86E+10

Table 2.4: Equivalent schedule of the 2^{nd} FNG irradiation campaign

Cooldown Time [d]	Cooldown Time [s]	Experimental SDDR [Sv/h]	Relative Error
1	86400	2.46E-06	0.1
7	604800	6.99E-07	0.1
15	1296000	4.95E-07	0.1
30	2592000	4.16E-07	0.1
60	5184000	3.16E-07	0.1

Table 2.5: Experimental measure of the SDDR during 1^{st} FNG irradiation campaign

Cooldown Time [s]	Cooldown Time [h]	Cooldown Time [d]	Experimental SDDR [Sv/h]	Relative Error
4380	1.22	0.05	4.88E-04	3.89E-02
6180	1.72	0.07	4.15E-04	3.86E-02
7488	2.08	0.09	3.75E-04	4.00E-02
11580	3.22	0.13	2.68E-04	3.73E-02
17280	4.80	0.20	1.73E-04	4.05E-02
24480	6.80	0.28	1.01E-04	3.96E-02
34080	9.47	0.39	5.06E-05	3.95E-02
45780	12.72	0.53	2.30E-05	3.91E-02
57240	15.90	0.66	1.17E-05	4.27E-02
72550	20.15	0.84	5.80E-06	3.97E-02
90720	25.20	1.05	3.56E-06	3.93E-02
132000	36.67	1.53	2.43E-06	3.70E-02
212400	59.00	2.46	1.78E-06	3.93E-02
345600	96.00	4.00	1.22E-06	4.10E-02
479300	133.14	5.55	9.52E-07	3.89E-02
708500	196.81	8.20	7.59E-07	3.95E-02
1050000	291.67	12.15	6.67E-07	3.90E-02
1670000	463.89	19.33	6.13E-07	3.92E-02
1710000	475.00	19.79	6.14E-07	3.91E-02

Table 2.6: Experimental measure of the SDDR during 2^{nd} FNG irradiation campaign

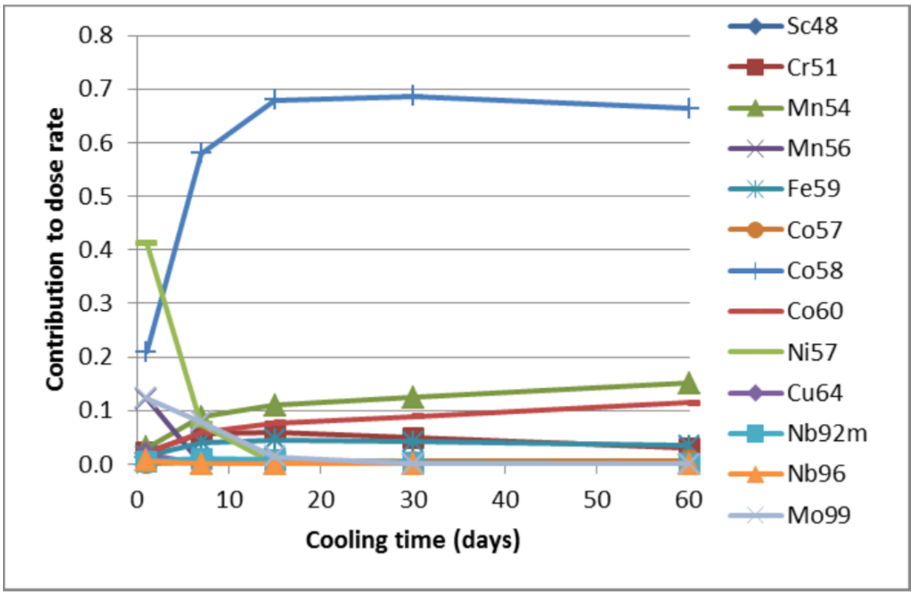


Figure 2.26: Isotope contribution to the the dose during the first FNG irradiation campaign

order to do so, preliminary activation calculation are usually performed with the help of activation codes like FISPACT or ACAB (ACtivation ABacus Inventory Code for Nuclear Applications) [88]. Fortunately these studies have been already conducted both during the D1S libraries initial V&V procedure [89] and when the experimental results were tested for the first time [85]. Fig. 2.26 is extracted from [89] and lists the isotopes contributing cumulatively to more than 95% of the dose rate during the first irradiation campaign, but similar results were also presented in [85]. At this point, the D1S reaction file can be generated: it will include all reactions that can originate in the material (i.e. that are also available in the activation library) which result in the creation of one of the daughters of interest. The D1S irradiation file will simply contain those daughters which are generated by at least one reaction. All of this implies that a comparison between two different libraries can often not be an exact one. Indeed, it is quite common that to a new library release corresponds an increase in the number of available reactions. Nevertheless, this is in line with the philosophy of JADE. If the Sphere benchmarks are the primary tools that should be used to identify specific inconsistencies at the single cross section level among libraries, all other benchmarks have a slightly different scope which is to show how big is the impact of these inconsistencies on more realistic applications.

Tallies

The only tallied result for the FNG benchmark is the dose rate at the dosimeter location inside the cavity (tally n.4).

Chapter 3

JADE Application Results

This chapter covers JADE's main application that have been carried out during the Ph.D. period.

3.1 Proof of concept

As soon as a first fully working version of JADE was finalized, the tool capabilities were tested comparing the results obtained from the Sphere Leakage benchmark (described in Section 2.3.1) using a number of different nuclear data libraries. Three releases of the FENDL libraries were tested, namely, 2.1, 3.0 and 3.1d, together with the ENDF/B-VII.1. The main goal of the work was to demonstrate how issues that affected FENDL releases in the past could have been easily spotted with JADE and, in general, to show the potential of the tool. The results of this first JADE application were published in [3], and some highlights are presented hereafter.

As described in the previous chapter, JADE does not only compare libraries results but also puts in place some consistency checks for the single library itself. One of them is to verify that the neutron or photon heating computed using the F4+FM method or the F6 one yield the same value (see Section 1.1.4 for additional information on the topic). This has been shown to be true for all the tested libraries with the exception of a number of isotopes in FENDL-3.1d. This was a known issue in the FENDL community and was already addressed in a FENDL Consultant's meeting official report [90]. Nevertheless, if it took months for the error to be spotted for the first time after the release publication, the systematic and complete testing of all isotopes provided by JADE allowed to do the same in a matter of days (mostly simulation time). In addition to this, natural Tin was found to produce a negative DPA value in FENDL-2.1. The fact that this was still an unknown issue, even if in an older release, can be considered a further proof of JADE V&V potential.

The vast majority of tested isotopes showed a substantial agreement between the different libraries results, although for a few of them this was not the case. An example is the case of Sulphur-36, shown in Fig. 3.1, where all libraries generated significantly different neutron leakage fluxes that converge only in the $>$ 1 MeV energy range. A similar, but less pronounced behavior is encountered in the Molybdenum-94 comparison shown in Fig. 3.2. In other cases, it appears that only one library diverges from the results of the remaining ones which, on the contrary, show a gen-

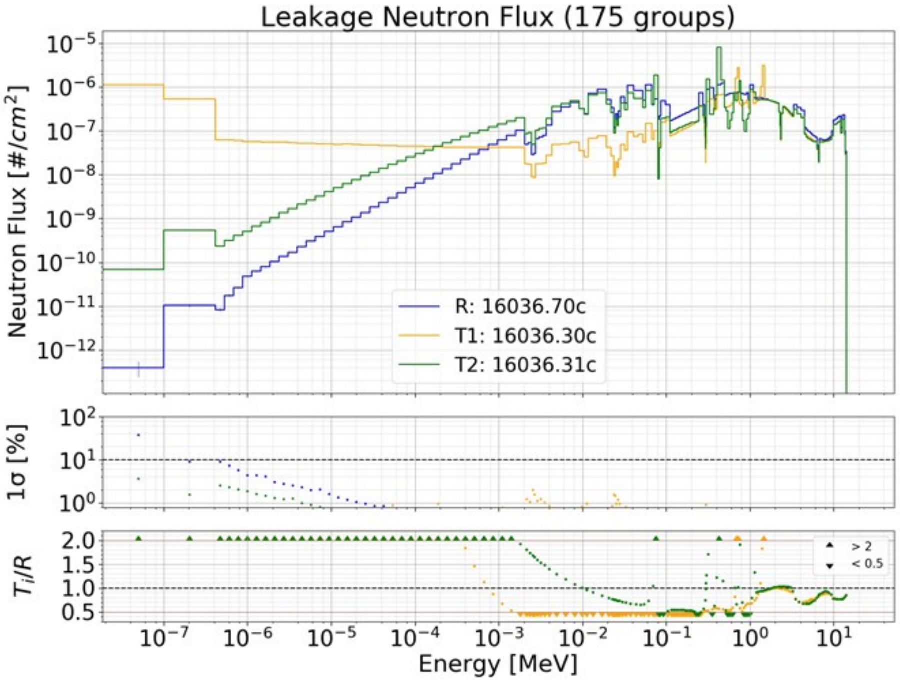


Figure 3.1: Vitamin-J 175 energy groups neutron leakage flux comparison of Sulfur-36 S^{36}_{16}

eral good agreement for the specific isotope. This is the case for Potassium-39 where FENDL 3.0 nuclear leakage flux highly differs from the other libraries in the energy range between 0.5 keV and 100 keV as reported in Fig. 3.3. A similar behavior is encountered in Cadmium-106, Fig. 3.4, where the FENDL 3.0 result values deviate in the energy range between 5 keV and 100 keV. Finally, for Iron-58 whose comparison is shown in Fig. 3.5, it can be observed how the libraries results are split in two: FENDL 2.1 and ENDF/B-VII.1 appears to be in agreement showing a different flux from the FENDL 3.0 and FENDL 3.1d ones that are in agreement too.

To conclude, this first JADE application was considered a successful proof of concept for the tool that was able to spot old and new inconsistencies and to easily highlight differences in the libraries responses. All of this with a far lower effort needed by a potential cross section evaluator thanks to JADE heavy automation.

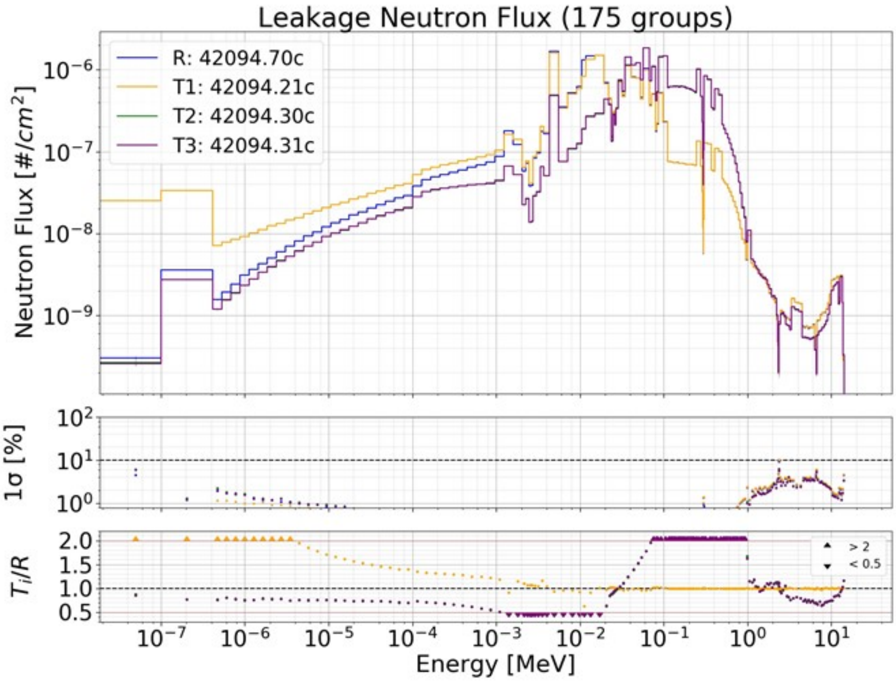


Figure 3.2: Vitamin-J 175 energy groups neutron leakage flux comparison of Molybdenum-94 S^{94}_{42}

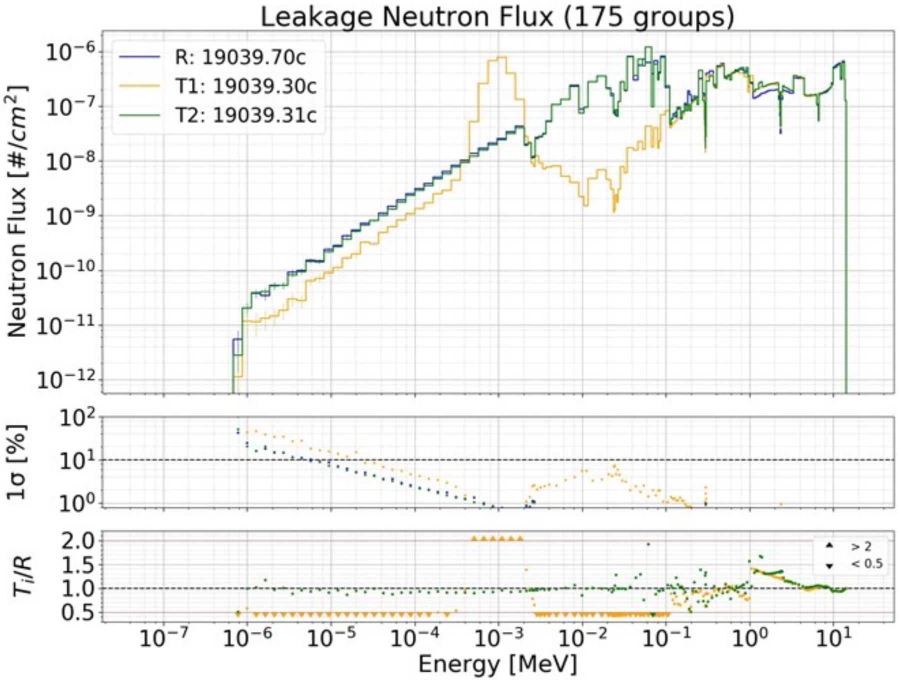


Figure 3.3: Vitamin-J 175 energy groups neutron leakage flux comparison of Potassium-39 K^{39}_{19}

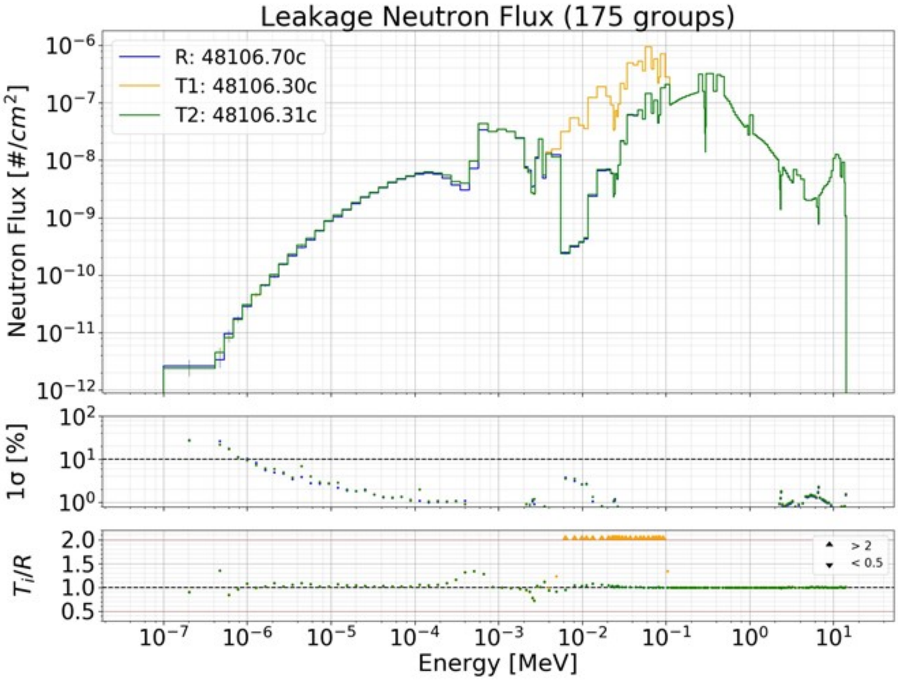


Figure 3.4: Vitamin-J 175 energy groups neutron leakage flux comparison of Cadmium-106 Cd_{48}^{106}

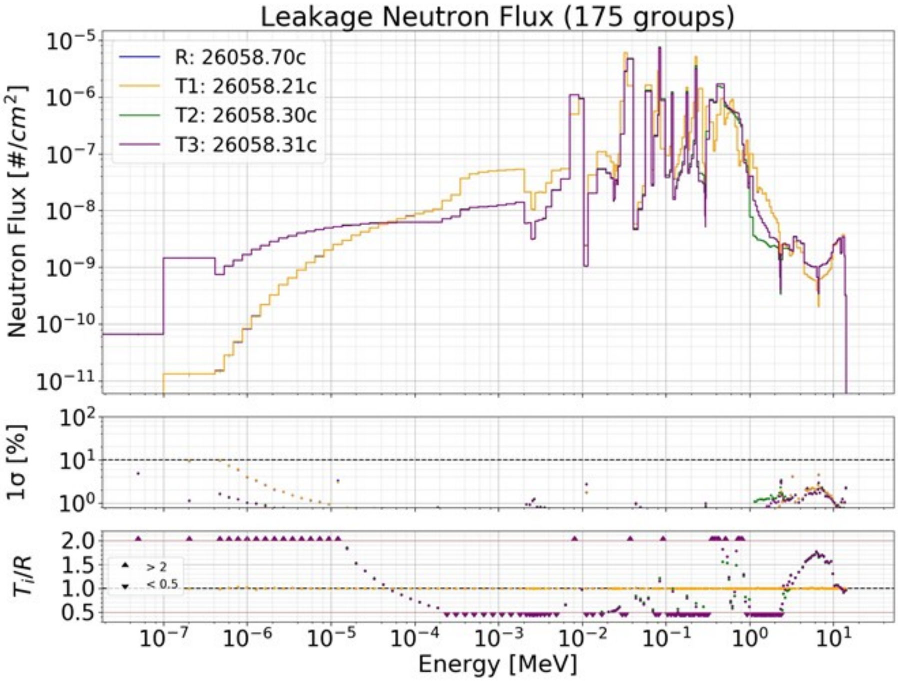


Figure 3.5: Vitamin-J 175 energy groups neutron leakage flux comparison of Iron-58 S_{26}^{58}

3.2 FENDL-3.2 release V&V

The release of a new FENDL library version (3.2) in 2021 provided the perfect opportunity to apply JADE to a first real production case.

3.2.1 Release-3.2 (Beta)

At first, the new FENDL-3.2 was distributed as a Beta version for the users to test. In particular, the library was tested on the Sphere Leakage (described in Section 2.3.1) and ITER 1D (described in Section 2.3.2) benchmarks and the results compared with the ones obtained from FENDL-2.1c, FENDL-3.1d and ENDF/B-VIII.0. The work was published in [4] and the most significant results are briefly discussed hereafter.

As far as consistency checks go, all inconsistencies found in the previous FENDL-3.1 release regarding nuclear heating computation (as discussed in the previous section) were corrected in the new FENDL release.

Regarding the comparison between the library results, as it could be expected, many differences were found at the single isotope level. Table 3.1 is an extract of the Sphere Leakage comparison between FENDL-3.1d and FENDL-3.2 excel sheet that lists the major variations for particle, production, DPA production and nuclear heating. Understanding how much these differences may affect real world calculations is not a trivial task, which is why it is important to cross-check the single isotope results with more application-specific benchmarks (fusion-related in this case). Surely, an initial idea of the potential impact of these variations is given by the typical ITER materials results coming from the Sphere Leakage benchmark and reported in Table 3.2. The first thing that can be noticed is the significant difference in tritium production that can be generally found between FENDL releases and the ENDF/B one in materials that compose the majority of the ITER tokamak such as steel and water. Clearly, the developing of an experimental benchmark oriented towards tritium production would help in clarifying which results are closer to reality. The dissimilarities that were more alarming though, were the ones resulting from the comparison between FENDL-3.2 and FENDL-3.1 for the neutron heating and DPA production in boron carbide and steel. The possible impact of this for fusion application was clearly underlined by the total nuclear heating (i.e., photon plus neutron) plot from the ITER 1D benchmark reported in Fig. 3.6. Thanks to the single isotope data it was easy to find the origin of this inconsistency, namely the Boron-10. Indeed, after a more in depth investigation, it turned out that the isotope values of DPA production and nuclear heating in FENDL-3.2 were six order of magnitude higher then the ones obtained using the other tested libraries due to an error occurred during the cross section evaluation (i.e. the .ace file generation).

To sum up, the application of JADE to the V&V of the new FENDL-3.2 Beta release proved the efficiency of automation and extensive testing in quickly discovering potential inconsistencies in a new data library release. The Boron-10 issue was later addressed and corrected in the official FENDL-3.2 release.

Zaid	Isotope Formula	T production	He ppm production	DPA production	Neutron heating F6	Gamma heating F6
5010	B-10	5.17%	0.49%	100.00%	100.00%	-0.50%
5011	B-11	39.56%	16.59%	0.21%	0.59%	3.73%
8016	O-16	-	17.63%	-3.09%	32.79%	-20.26%
8018	O-18	-	-91.74%	-35.81%	-12.19%	-47.54%
19040	K-40	14.38%	20.19%	-12.74%	2.71%	-43.23%
19041	K-41	31.85%	2.68%	-9.84%	-21.56%	-33.71%
22046	Ti-46	6553500%	0.00%	0.00%	Identical	0.00%
24050	Cr-50	89.61%	46.91%	4.19%	13.07%	7.22%
24052	Cr-52	62.82%	33.92%	14.74%	24.65%	20.19%
24053	Cr-53	-26.45%	-100.76%	0.51%	-6.21%	12.33%
24054	Cr-54	88.15%	13.32%	11.26%	12.60%	12.77%
25055	Mn-55	-0.01%	-28.15%	0.02%	0.43%	-0.10%
26054	Fe-54	100.00%	-8.83%	7.18%	-0.70%	2.34%
26056	Fe-56	-19581%	0.35%	1.32%	-0.61%	2.65%
26057	Fe-57	100.00%	7.65%	-3.49%	-1.82%	-1.99%
26058	Fe-58	0.00%	0.01%	5.48%	23.11%	0.72%
28062	Ni-62	-	-0.01%	6.22%	11.52%	1.61%
30070	Zn-70	-0.02%	0.00%	9.65%	14.95%	1.12%
48110	Cd-110	0.01%	-0.02%	-2.49%	-242.21%	100.00%
48112	Cd-112	0.03%	-0.01%	-3.43%	-169.67%	100.00%
48114	Cd-114	0.00%	0.00%	-2.95%	-142.71%	100.00%
48116	Cd-116	-0.02%	-0.02%	-3.52%	-160.08%	100.00%
74180	W-180	-	-91.18%	0.00%	0.00%	0.00%
74182	W-182	-	-98.61%	0.74%	1.80%	5.58%
74183	W-183	-	-96.15%	0.00%	0.02%	0.08%
74184	W-184	-	-98.15%	2.29%	3.96%	11.00%
74186	W-186	-	-97.22%	2.68%	4.25%	12.75%
90232	Th-232	-	-98.43%	0.01%	-4.00%	-0.01%

Table 3.1: Extract from the Sphere Leakage benchmark Excel comparison of FENDL-3.2 VS FENDL-3.1d

Material	FENDL-3.2 Vs	T Production	He Production	DPA	Neutron heating (F6)	Gamma heating (F6)
Water	FENDL-3.1d	1.09%	16.18%	-	6.51%	-2.34%
	ENDF/B-VIII.0	62.96%	7.82%	-	6.41%	-1.93%
Ordinary Concrete	FENDL-3.1d	-0.26%	3.10%	84.98%	96.72%	-2.53%
	ENDF/B-VIII.0	-4.52%	9.24%	85.99%	96.71%	2.52%
Boron Carbide	FENDL-3.1d	13.13%	0.80%	100.00%	100.00%	0.95%
	ENDF/B-VIII.0	-164.28%	-7.35%	100.00%	100.00%	3.56%
SS316L(N)-IG	FENDL-3.1d	8.58%	0.58%	67.87%	97.53%	3.91%
	ENDF/B-VIII.0	-79.74%	-8.15%	68.37%	97.64%	0.00%
Natural Silicon	FENDL-3.1d	-	-0.07%	-0.06%	-0.06%	-0.03%
	ENDF/B-VIII.0	-	-0.07%	-0.06%	-0.21%	0.07%
Polyethylene, non-borated	FENDL-3.1d	-	-0.01%	-0.01%	0.00%	0.02%
	ENDF/B-VIII.0	-	4.61%	-0.68%	-1.64%	-0.49%

Table 3.2: Sphere leakage benchmark results comparison for typical ITER materials

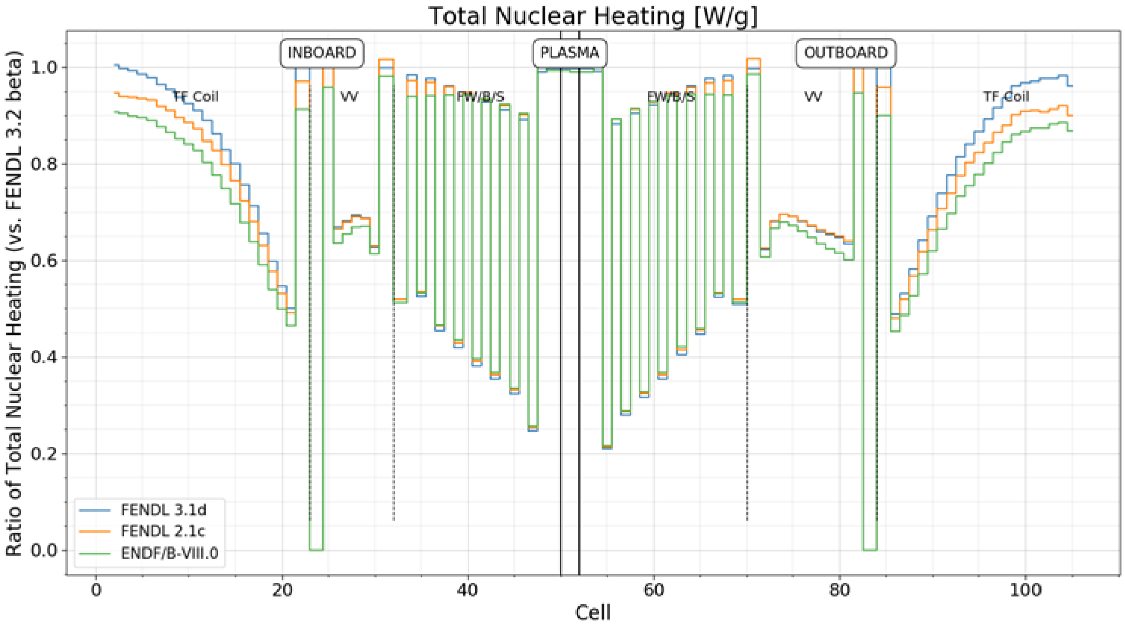


Figure 3.6: Nuclear heating along the ITER 1D model, ratio of different libraries results against the FENDL-3.2 one

3.2.2 Release-3.2 (Official)

After users feedback, FENDL-3.2 was adjusted and officially released. A comparison was run using JADE between FENDL-3.2 and FENDL-3.1d on the following benchmarks:

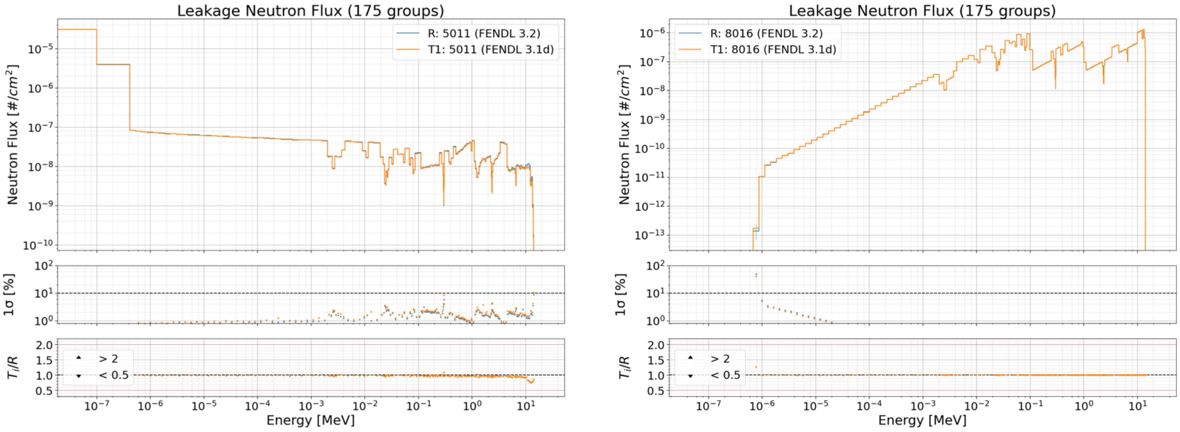
- the Sphere Leakage benchmark (see Section 2.3.1);
- the ITER 1D benchmark (see Section 2.3.2);
- the TBM (HCPB) benchmark (see Section 2.3.3);
- the Oktavian experimental benchmark (see Section 2.3.7).

As a starting point for the results discussion is useful to observe an extract from the typical ITER material comparison from the Sphere Leakage benchmark provided in Table 3.3. Two main results should draw the attention of the reader and they are the difference in tritium production observed in Boron Carbide ($\approx 13\%$) and in the neutron heating computed in water ($\approx 6.5\%$). For the Boron Carbide, the difference in tritium production can be traced back to the B-11 isotope that presented a 39.5% difference in the sphere leakage benchmark, while, for the water, a significant difference in neutron heating, 32.8%, was observed in O-16. Looking at the leakage neutron flux plots shown in Fig. 3.7 it can be concluded that the origin of the variations do not depend on the neutron flux, but on the specific cross-sections instead (i.e. tritium production and heating generation). Even if tritium production in Boron Carbide is interesting from a nuclear fusion reactor point of view, a variation in nuclear heating is significantly more important due to the amount of water used in the reactors and the general high value of heat that is generated from the neutron

Material	T production	He ppm production	DPA production	Neutron heating F6	Gamma heating F6
SS316L(N)-IG	8.59%	0.57%	0.15%	-8.67%	3.92%
Ordinary Concrete	-0.27%	3.09%	-0.42%	10.15%	-2.53%
Boron Carbide	13.13%	0.80%	-0.09%	-1.28%	0.95%
Water	1.10%	16.19%	-1.86%	6.52%	-2.35%
Natural Silicon	-	0.00%	0.00%	0.00%	0.00%
Polyethylene, non-borated	-	0.00%	0.01%	0.02%	0.01%

Table 3.3: Comparison of Sphere Leakage tally results for typical ITER materials between FENDL-3.2 (official release) and FENDL-3.1d

.



(a) Boron B_5^{11}

(b) Oxygen O_8^{16}

Figure 3.7: Leakage neutron flux comparison of FENDL-3.2 (official) VS FENDL-3.1d for B-11 and O-16

moderation. This relative difference between the two FENDL releases is further amplified when looking at a more realistic benchmark such as the ITER 1D. Fig. 3.8 plots the nuclear heating computed along the ITER 1D model and the “valleys” that can be seen are in proximity of the water layers, showing how the total nuclear heating computed in those areas can vary up to 12%.

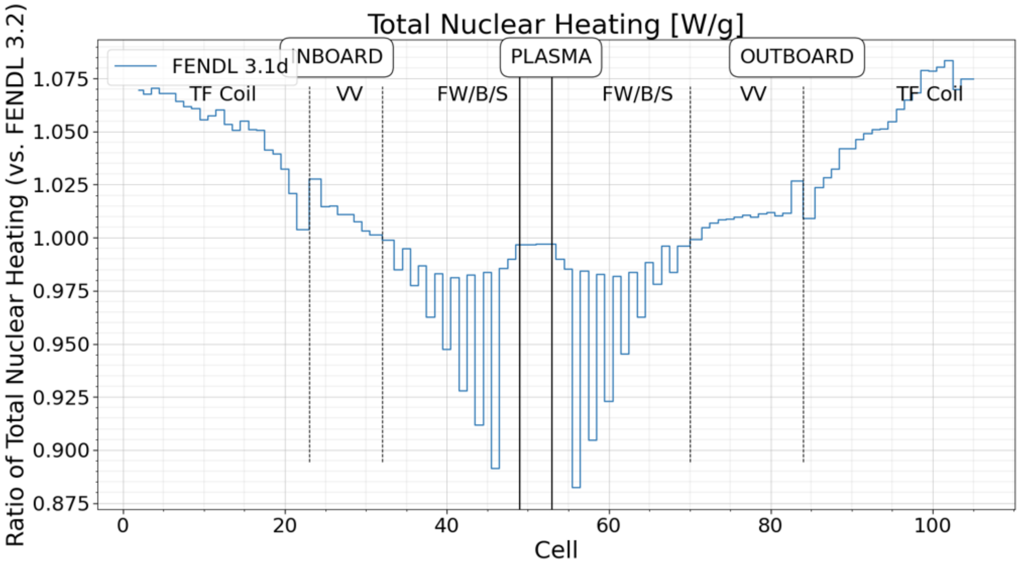


Figure 3.8: Neutron heating distribution along ITER 1D model

3.2.3 Release-3.2b

The final (stable) version released for FENDL was v3.2b. Here, the O-16 neutron heating discrepancy discussed in Section 3.2.2 was addressed correcting the NJOY processing routine used to obtain the KERMA data. No change in the B-11 data was performed though. This new release was also chosen as reference for an extensive paper [91] that summarizes the state of the FENDL libraries. JADE team contributed to such paper in the Verification & Validation section with the Sphere Leakage, the Oktavian and the TBMs benchmarks.

The author believes that there would not be any additional value in reporting all the results derived from these studies, partly because they are similar to what was already described in the previous sections and partly because they were detailed in the above mentioned paper. Nevertheless, it is still worth to show an example of application of the TBM benchmarks described in Section 2.3.3 since it has not been showcased yet here. Among the different tallies defined, the more interesting ones given the specific function of the TBMs are the neutron flux and the tritium production spatial distribution. These results are plotted for the TBM region respectively in Fig. 3.9 and Fig. 3.10 for HCPB benchmark and in Fig. 3.11 and Fig. 3.12 for the WCLL benchmark.

Neutron flux results appear to be more or less consistent between the different libraries in both benchmarks. The differences between the new and older FENDL releases are contained between ±10% and specifically in the breeding areas this is further contained to ±5%. In both benchmarks though, it can be observed how ENDF/B-VIII.0 results in the homogeneous water/steel mixture tend to diverge significantly from all the other tested libraries. With regard to the tritium production, it can be observed how in the breeding region the value is almost the same for all tested libraries while significant variations are observed between FENDL results and the other libraries. These variations though are far less important due to the very low absolute value of tritium production in these regions.

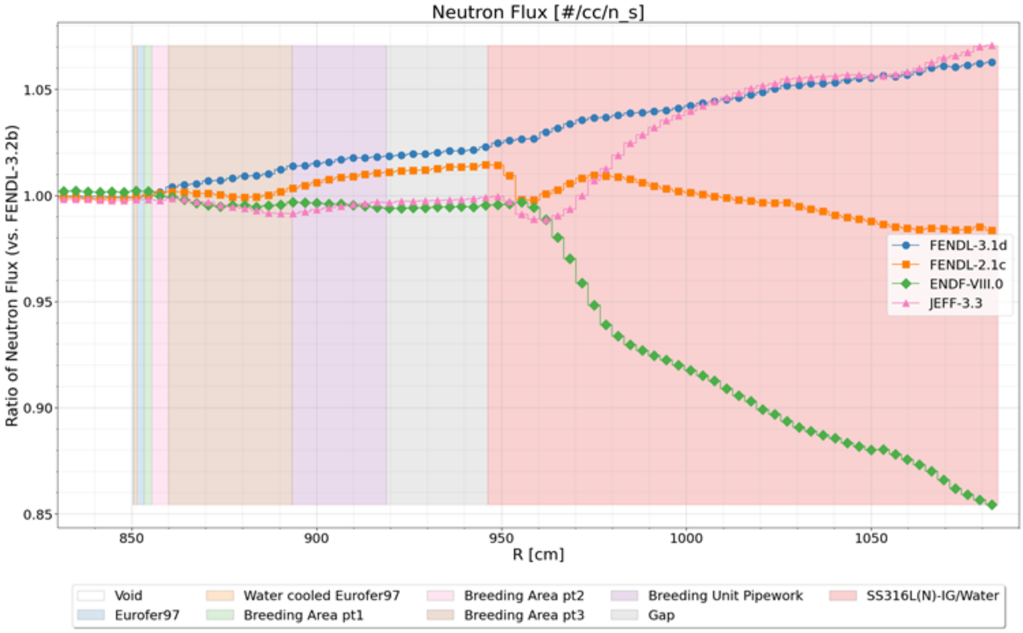


Figure 3.9: Neutron flux in the HCPB benchmark, focus on the TBM region

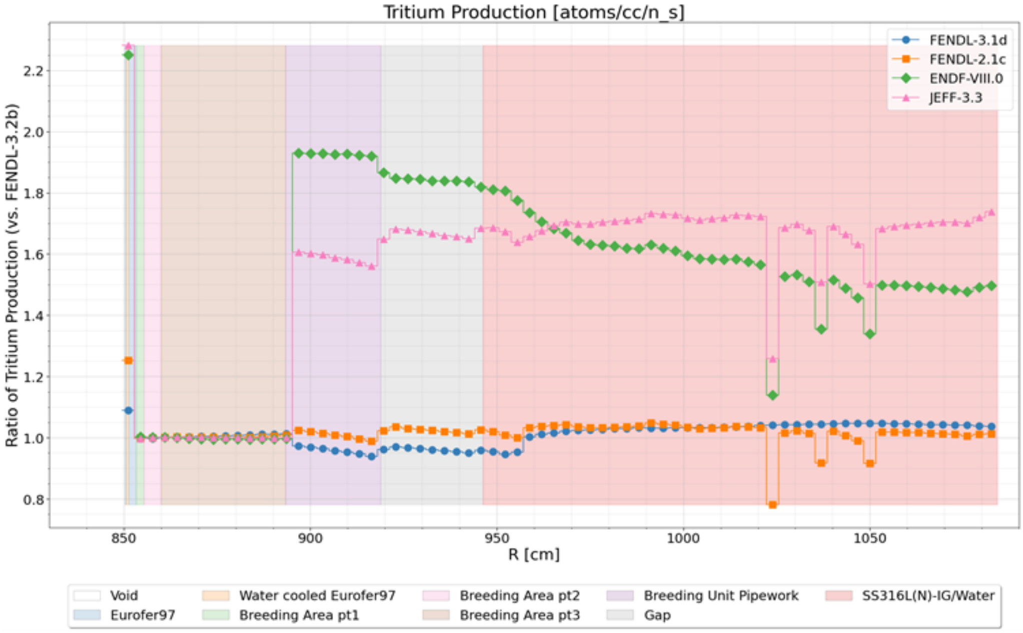


Figure 3.10: Tritium production in the HCPB benchmark, focus on the TBM region

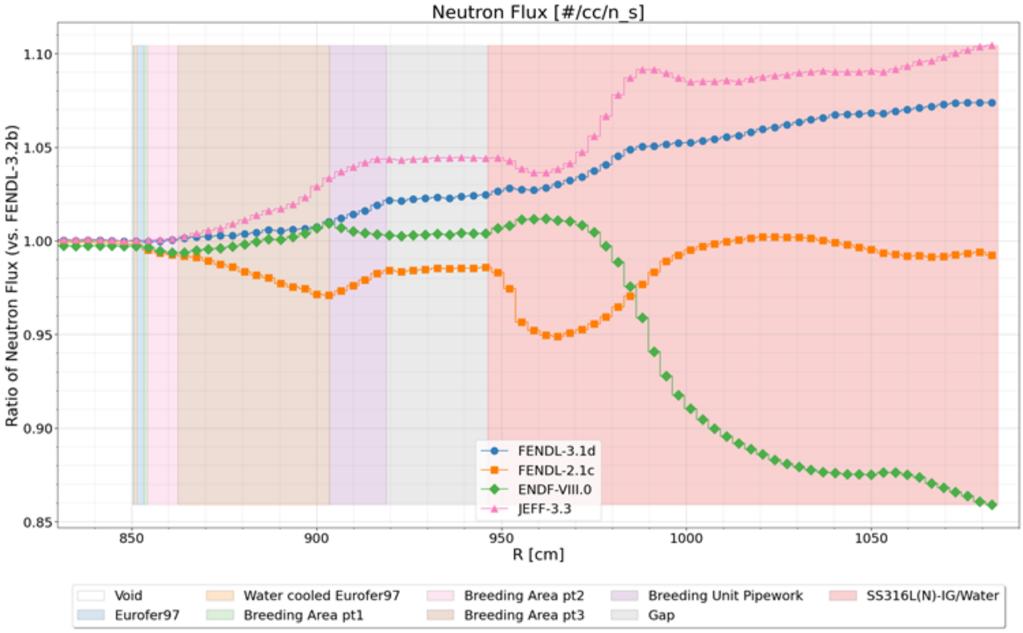


Figure 3.11: Neutron flux in the WCLL benchmark, focus on the TBM region

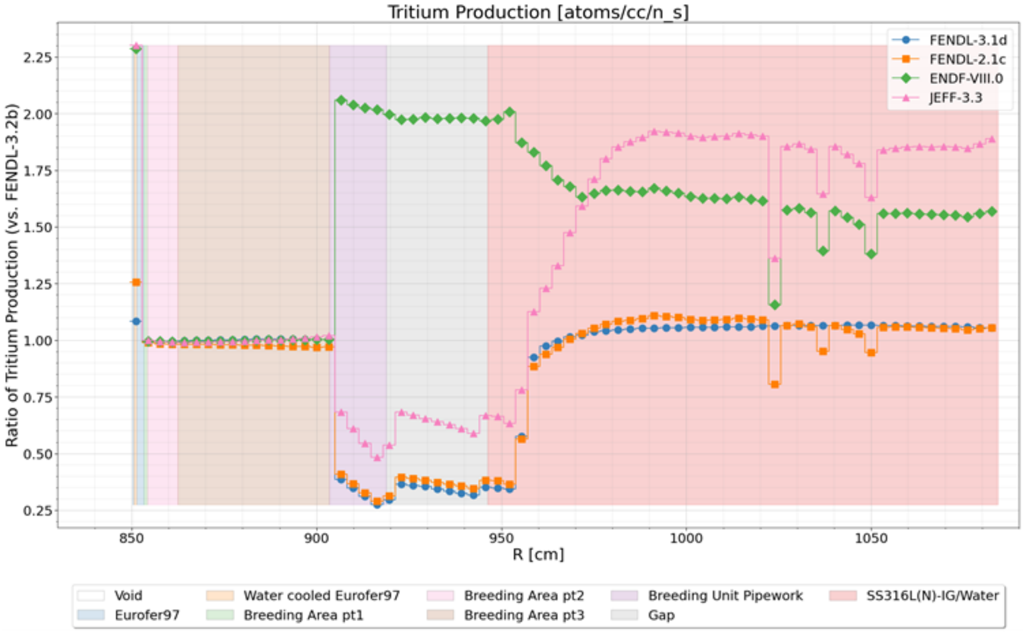


Figure 3.12: Tritium production in the WCLL benchmark, focus on the TBM region

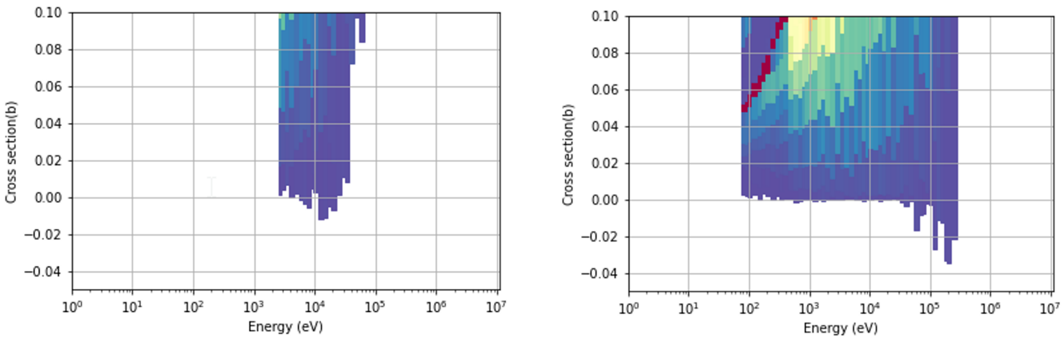
3.3 JEFF

In addition to FENDL, other fusion-oriented libraries were tested, such as the latest release of JEFF libraries (v4.0T1). Here, out of the more than 500 isotopes tested, 5 failed the neutron heating consistency check included in the Sphere Leakage benchmark, meaning that negative values were tallied. For Xe-128, Xe-129 and Xe-133, it appears that the issue may be related to negative values in the p-table (the probability table in the unresolved resonance region) as shown Fig. 3.13. For In-113 and In-115, instead, the negative heating values may be due to total negative KERMA (MT=301).

After the consistency checks, the library results were compared with the ones obtained from FENDL-3.2b, JEFF-v3.3 and ENDF/B-VIII.0. A good agreement was observed on the major quantities tallied in Sphere Leakage benchmark between JEFF-v4.0T1 and FENDL-3.2b as summarized in Table 3.4 which reports the comparison of the two libraries results on the typical ITER materials. It is worth to highlight though how the JEFF library predicted a 15% higher gamma flux leaking from the SS316L(N)-IG steel sphere with respect to the FENDL one. It seems that this difference does not depend on one single isotope, but on many key constituents of the material as shown in Table 3.5, which shows the energy binned comparison of the gamma leakage flux for Cr-52, Fe-56, Ni-58 and Ni-60.

It is also worth to discuss some of the results obtained from the OKTAVIAN benchmark. As a general comment, JEFF-4.0T1 results fit quite well the experimental data and are generally aligned with the other tested libraries. Some takeways are:

- LiF: JEFF-4.0T1 presents a better agreement with experimental data around 10 MeV but worse below 0.5 MeV;
- Al: JEFF-4.0T1 presents a worst agreement in the 7 to 10 MeV range;
- Co: JEFF-4.0T1 is aligned with others but for Cobalt all computational results underestimate the experimental ones;



(a) Xe-129

(b) Xe-133

Figure 3.13: Negative values observed for some Xe isotopes p-tables. Plots produced using NJOY21

ITER typical material	Total neutron flux	Total gamma flux	T production	He ppm production	DPA production	Neutron heating	Gamma heating
SS316L(N)-IG	-3.84%	15.30%	36.65%	-9.57%	-2.41%	11.73%	1.67%
Ordinary Concrete	5.22%	14.71%	4.83%	-8.21%	-2.35%	-1.55%	5.87%
Boron Carbide	-3.05%	-1.17%	-2.42%	-0.52%	0.15%	1.32%	-0.76%
Water	0.15%	-0.06%	5.42%	-1.00%	0.03%	0.21%	-0.03%
Natural Silicon	2.47%	-3.25%	N.A.	23.28%	-3.40%	2.51%	3.60%
Polyethylene, non-borated	0.06%	0.40%	N.A.	-4.81%	0.67%	1.60%	0.48%

Table 3.4: Comparison of the main Sphere Leakage tallies for ITER typical materials between JEFF-v4.0T1 and FENDL-3.2b (JEFF used as reference)

Isotope	E<0.01 [MeV]	0.01<E<0.1 [MeV]	0.1<E<1 [MeV]	1<E<5 [MeV]	5<E<20 [MeV]	Total
Cr-52	51.03%	-21.94%	-16.20%	-4.76%	-38.56%	-13.03%
Fe-56	-16.05%	14.15%	18.61%	42.76%	-58.51%	23.26%
Ni-58	51.73%	46.15%	44.11%	62.12%	-53.76%	40.99%
Ni-60	21.86%	65.77%	48.26%	63.26%	-80.17%	46.59%

Table 3.5: Comparison of the Sphere Leakage gamma flux in different energy bins for a few isotopes that are part of SS316L(N)-IG between JEFF-v4.0T1 and FENDL-3.2b (JEFF used as reference)

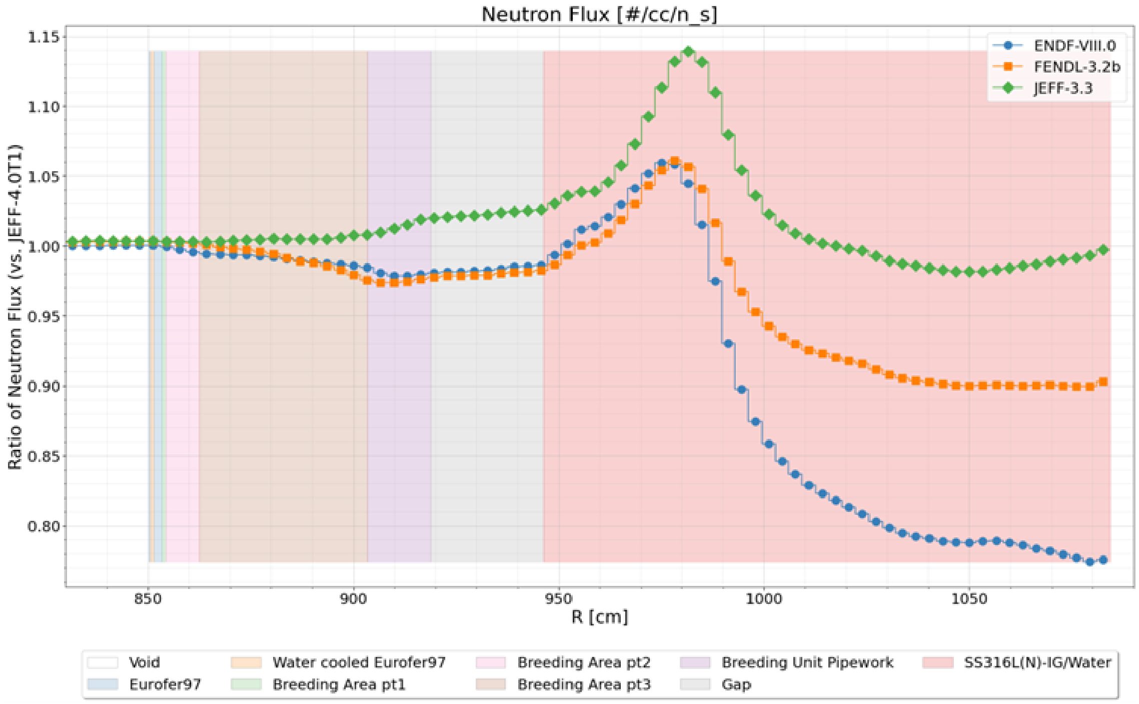


Figure 3.14: WCLL TBM benchmark, JEFF-v4.0T1 assessment, neutron flux

- Mo: JEFF-4.0T1 presents a worst agreement in the around 3 to 10 MeV range;
- Si: JEFF-4.0T1 presents a worst agreement in the around 6 to 10 MeV range;
- Zr: JEFF-4.0T1 presents a worst agreement in the around 7 to 8 MeV range.

The plots of the the neutron leakage current from which these considerations are derived can be found in Appendix C.1.

Finally, looking at the WCLL TBM results, some interesting considerations can be made. First of all, one can look at the neutron flux comparison reported in Fig. 3.14 and the neutron heating comparison shown in Fig. 3.15. For both tallies, a very good agreement is registered in the TBM Box area, while after that, in the TBM shielding composed by homogenized water and steel, only the values obtained from JEFF-v3.3 are quite in-line with the JEFF-v4.0T1, while FENDL-v3.2b underestimates the flux (and consequently the heating) by more or less 10% and the ENDF/B-VIII.0 by 20%. Somewhat more surprising though are the results obtained for the photon flux and heating, reported in Fig. 3.16 and Fig. 3.17. Here, it can be observed how more or less the values computed using FENDL, ENDF and JEFF-v3.3 are more or less comparable, while JEFF-v4.0T1 alone appears to underestimate the secondary photons flux (and consequently photon heating) even if it overestimates the neutron flux with respect to the other libraries.

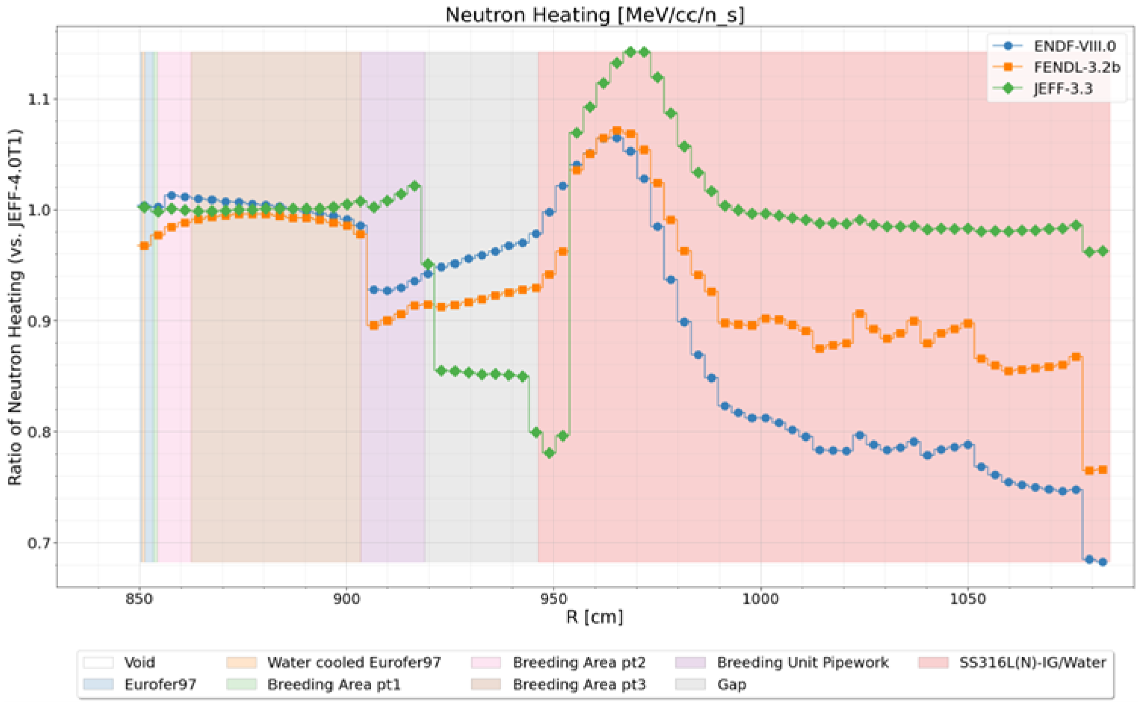


Figure 3.15: WCLL TBM benchmark, JEFF-v4.0T1 assessment, neutron heating

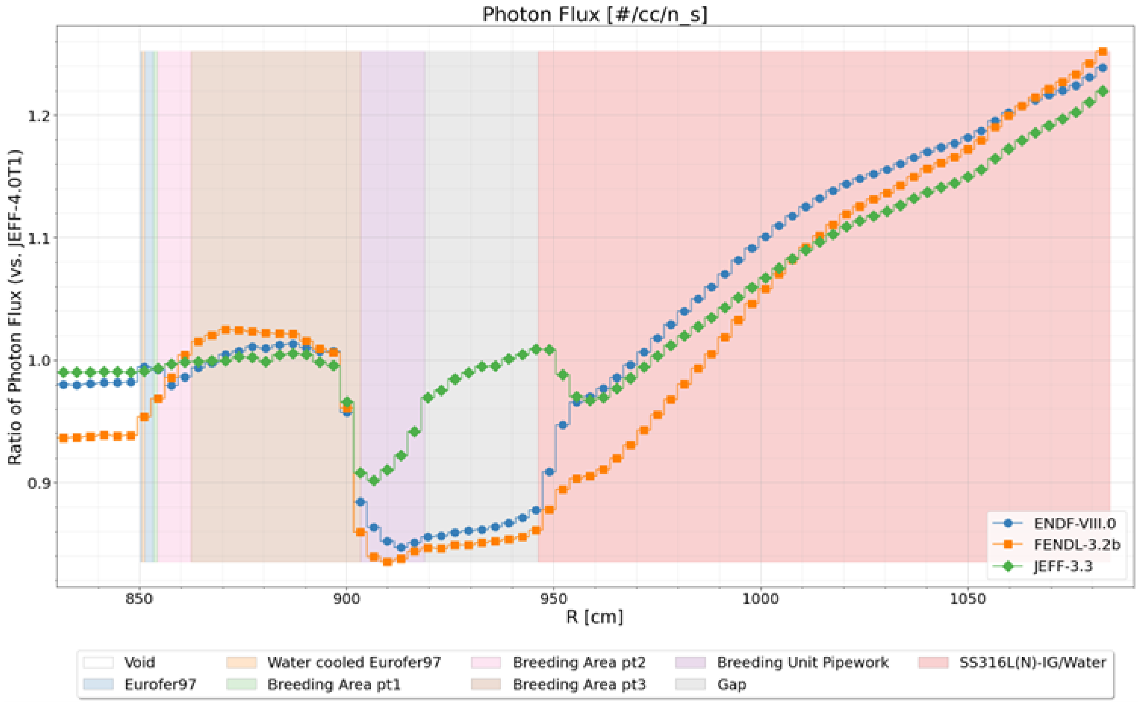


Figure 3.16: WCLL TBM benchmark, JEFF-v4.0T1 assessment, photon flux

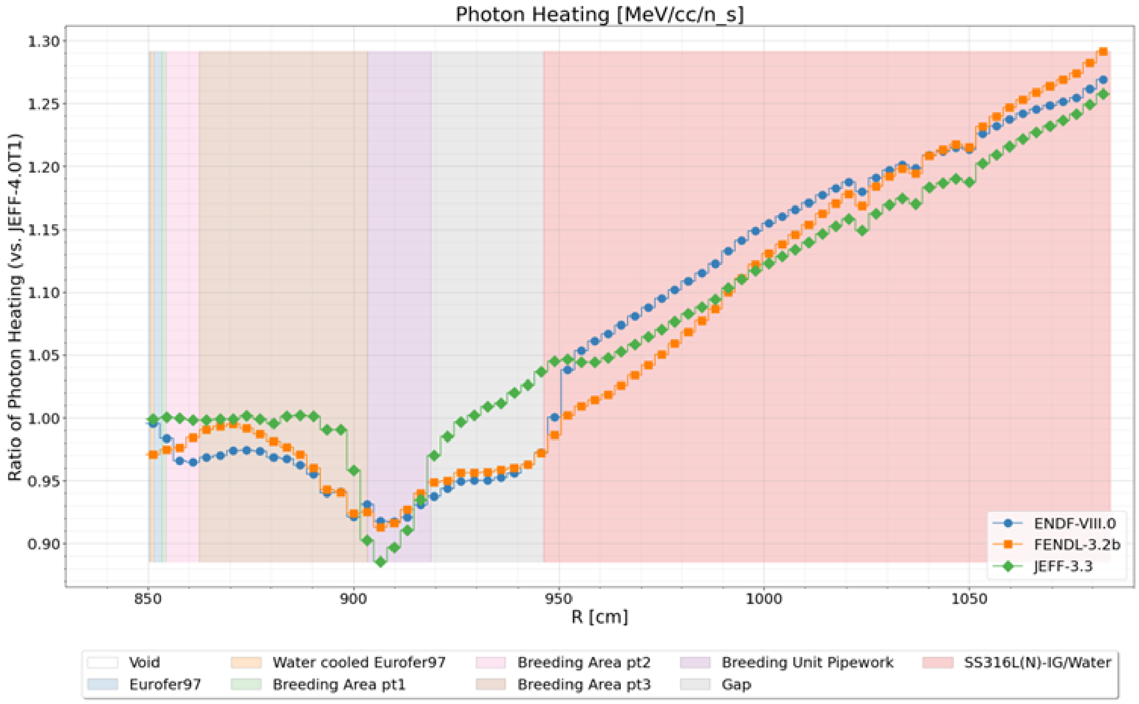


Figure 3.17: WCLL TBM benchmark, JEFF-v4.0T1 assessment, photon heating

3.4 MCNP cross sections models

Apart from nuclear data libraries, through JADE it is also possible to test the default theoretical models provided by MCNP when .ace cross sections are not available. In order to do so, it is necessary to modify the default inputs distributed with JADE as follows:

- add the “MPHYS: ON” card in order to activate the model physics;
- set the energy cutoff in the “PHYS” card to be equal to zero in order for the code to always use models instead of cross sections (e.g. “cutn” entry for neutrons).

MCNP models were tested on the Oktavian experimental benchmark and the results were compared with the ones obtained from FENDL-3.2b, JEFF-3.3 and END/B-VIII.0. As a general comment it can be stated that .ace cross sections seem to reproduce significantly better the experimental data with respect to the models. Nevertheless, the models performance vary considerably depending on the material under study. For instance, there are materials like the Aluminum (see Fig. 3.18), where the neutron leakage current obtained using the MCNP models is more similar to the experimental results. For other materials, like Molybdenum (see Fig. 3.19), the models appear to struggle at lower energies (i.e. below 1 MeV), while performing better at higher energies. Finally, for many materials, the models results are generally very distant from the experimental ones. For instance, this is the case for Copper, whose results are shown in Fig. 3.20.

The neutron leakage plot for each of the tested material can be found in Appendix C.1.

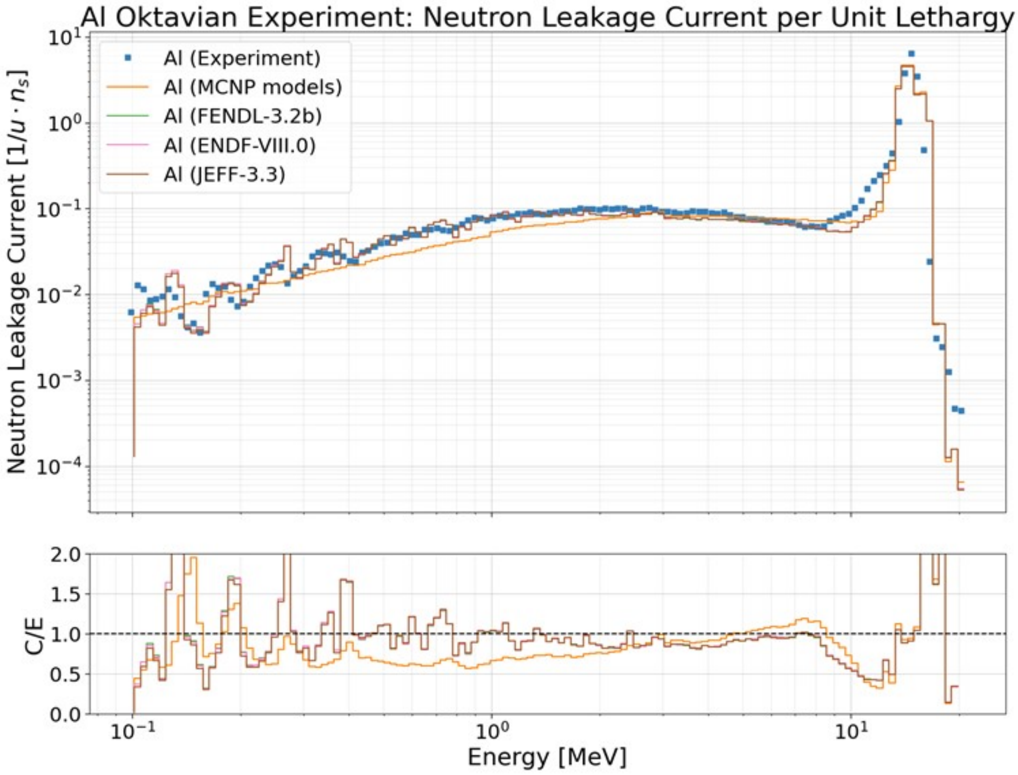


Figure 3.18: Oktavian experiment, neutron leakage current for Aluminum

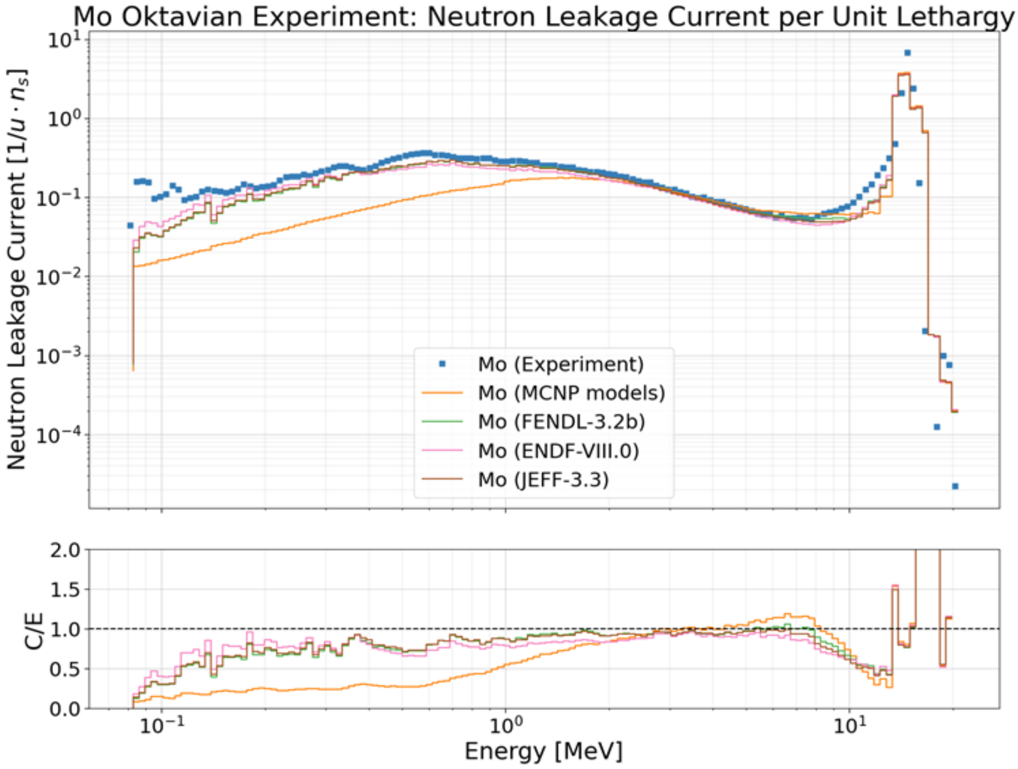


Figure 3.19: Oktavian experiment, neutron leakage current for Molybdenum

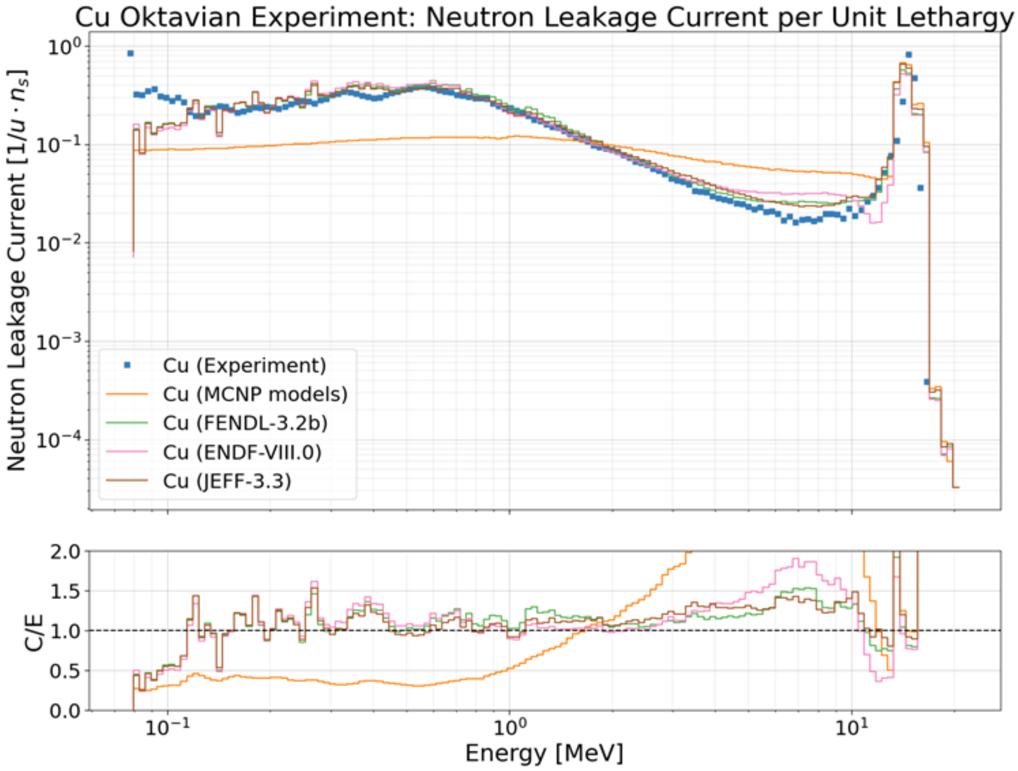


Figure 3.20: Oktavian experiment, neutron leakage current for Copper

3.5 SDDR applications

Unfortunately, at the time of writing, there are only two libraries available for D1S-UNED (see Section 1.1.5 for additional details on the D1S-UNED code). The activation data for both libraries is extracted from the European Activation File EAF-2007 [92], while for transport, the first is based on FENDL-2.1 while the second on FENDL-3.1d.

There is not a great interest in comparing these two libraries since the first one is considered to be obsolete. Nevertheless, the JADE SDDR benchmark suite is quite important since the generation updated D1S libraries is expected in the near future. In fact, the IAEA Nuclear Data Section now recommends TENDL 2017 as the go to library for activation calculation (and not the EAF anymore). For this reason, the two libraries were compared anyway in order to test the SDDR benchmarking capabilities of JADE and to individuate possible improvements to be performed on this part of the code.

3.5.1 Sphere SDDR

As detailed in Section 2.3.5, in the sphere SDDR benchmark each reaction channel of every isotope available in the library is individually tested. Fig. 3.21 shows a summary of the ratio of neutron leakage flux, photon leakage flux and SDDR (immediately after irradiation) for all the channels between the two tested libraries. This kind of plot can be used to rapidly visualize for which reaction channels the

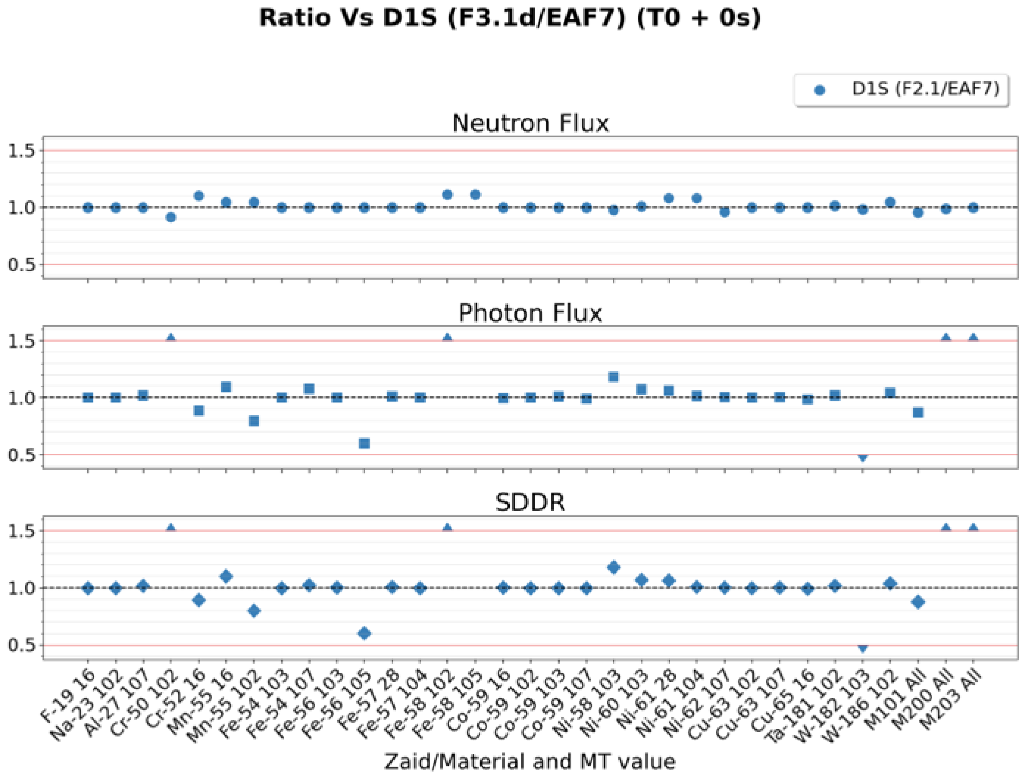


Figure 3.21: Summary of the comparison for the sphere SDDR benchmark between F3.1d/EAF7 and F2.1/EAF7 libraries

libraries results differed the most. The major variations can be observed in Cr-50 (MT=102), Fe-56 (MT=105), Fe-58 (MT=105, 102) and W-182 (MT=103) which are then reflected in differences in the SS316L(N)-IG (M101) and in the ordinary concrete (M200) results. Cross-checking these results with the one obtained for transport sphere leakage benchmark (FENDL-3.1d VS FENDL-2.1), it can be observed that for some of them the origin of the variation may be due to the transport and not the activation cross section. This does not seem the case for Fe-56 where the neutron leakage flux is almost identical between the two FENDL release, while, for both (decay) photon leakage flux and SDDR values, almost a 40% variation is registered.

3.5.2 ITER cylinder SDDR

For additional details on the ITER cylinder SDDR benchmark the reader is referred to Section 2.3.6.

The first thing to notice is that the neutron flux results obtained from the two libraries are in agreement with each other as showed in Fig. 3.22. Then, the (decay) gamma flux and dose rate were tallied at different radial sectors, 30 cm beyond the closing plate of the cylinder. These values were computed both considering a 0 s cool-down (Fig. 3.23) and 1E6 s (Fig. 3.24) cool-down. For the first case, it can be observed how the maximum difference in SDDR is around 7.9%, which is compatible with the differences observed in steel that were discussed in Section 3.5.1. In the

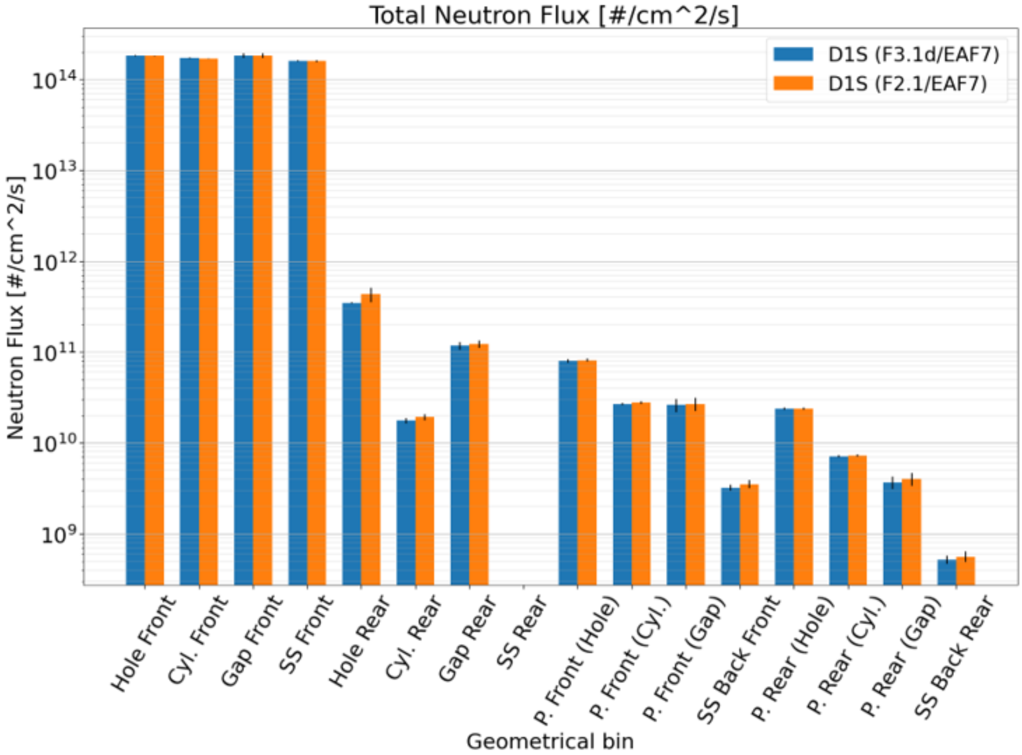


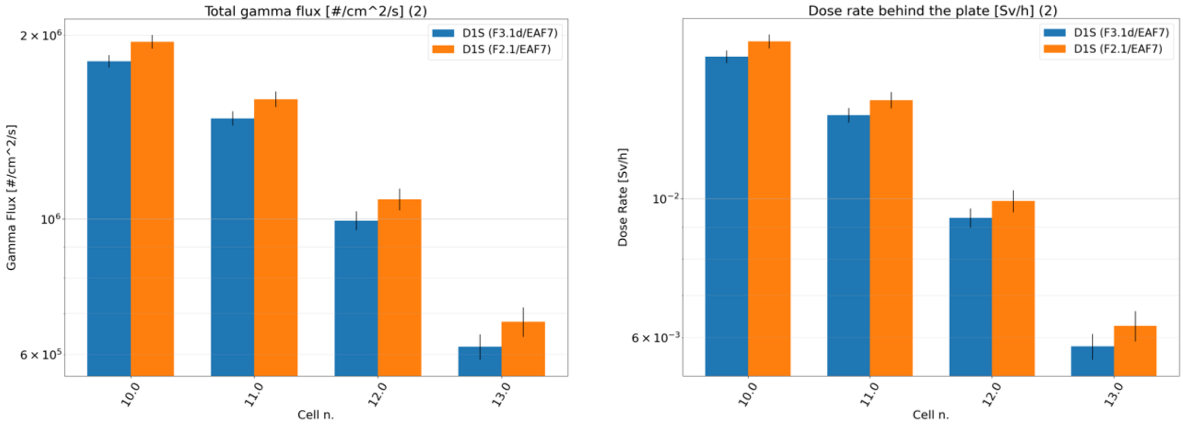
Figure 3.22: Neutron flux tally results at different model locations, comparison between the two D1S available libraries

second case, the differences between the results are lower (only 2.46%) and it can be observed how the obtained value are in line with the ones obtained during the D1S activation libraries validation [89] which are reported in Fig. 3.25.

3.5.3 FNG experiment

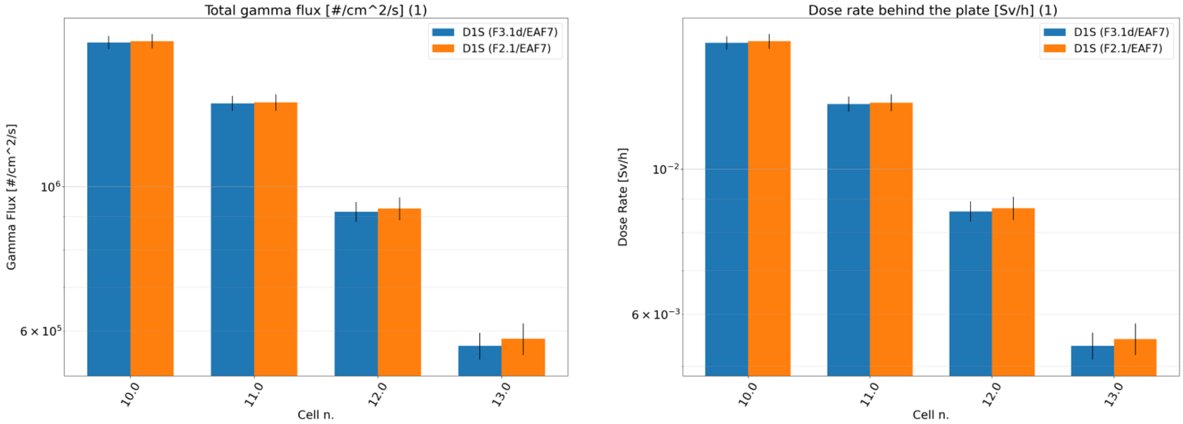
For additional details on the Frascati Neutron Generator SDDR benchmark the reader is referred to Section 2.3.8. Since this is an experimental benchmark, only the FENDL 3.1d based D1S library has been tested on the FNG benhmark., Indeed, at the time of writing, this is the official library for SSDR calculations for the ITER project and the only library that is not considered obsolete.

The SDDR C/E (i.e., computational over experimental) results for the first irradiation campaign are very close to one, as it can be observed in Fig. 3.26. Worse agreement between the computational and the experimental results can instead be found for the second irradiation campaign, as shown in Fig. 3.27. Indeed, the C/E value here is as low as 0.59 at 1.22 hours after cooldown and progressively improves up to ≈ 0.8 after 4 days. Similar results have been obtained also in [93] where the performance of different Monte Carlo codes was tested on a number of benchmarks including the FNG SDDR one. Since the results showed in [93] tend all to agree on these low C/E values among the different codes that were used, the shift between computational and experimental results is most likely due to limits in the modelling of the experiment (e.g. the characterization of the neutron source) instead of errors in the D1S library.



(a) Total photon flux **(b)** Dose rate

Figure 3.23: Comparison of the gamma flux and dose rate obtained using the two available D1S libraries at 0 s cool down



(a) Total photon flux **(b)** Dose rate

Figure 3.24: Comparison of the gamma flux and dose rate obtained using the two available D1S libraries at 1e6 s cool down

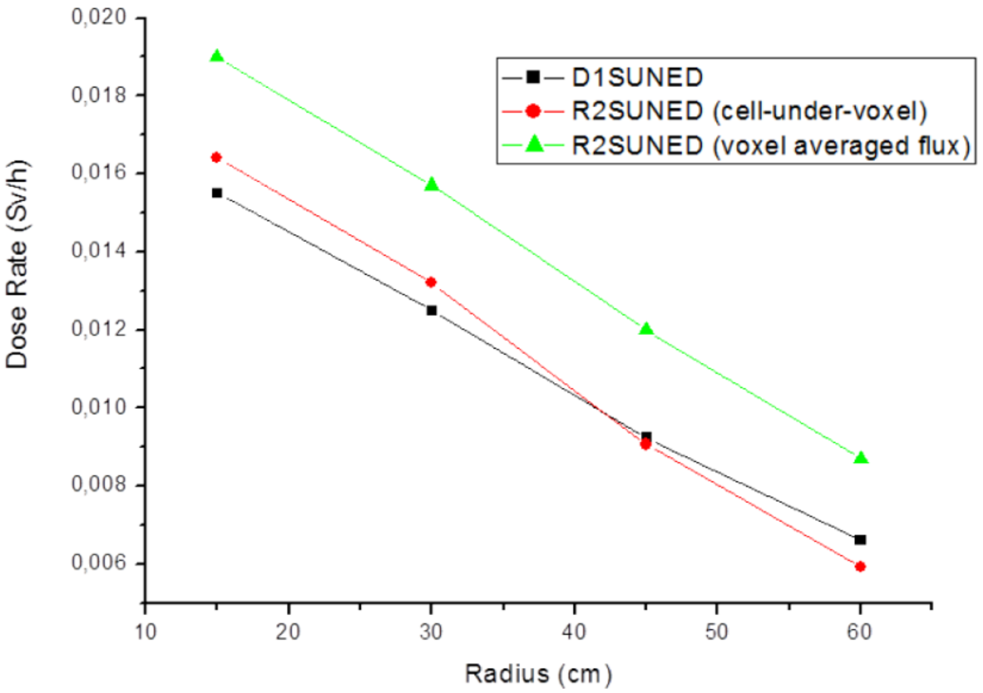


Figure 3.25: Dose rate 1E6 s after shutdown computed using different methods during the D1S-UNED code V&V

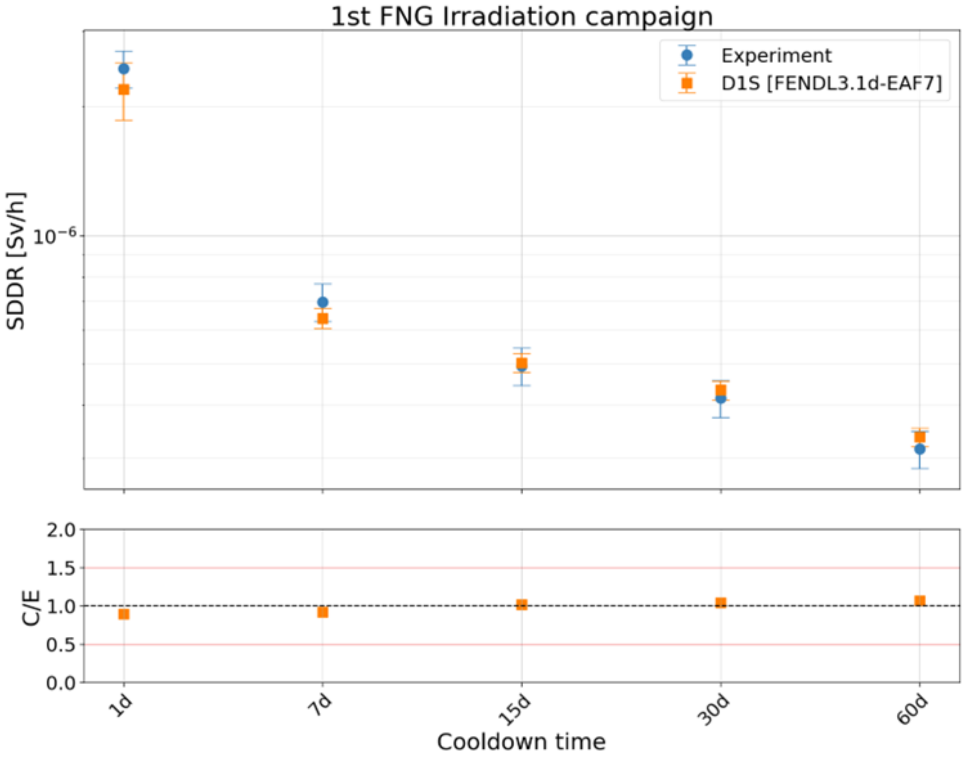


Figure 3.26: Comparison between experimental and FENDL-3.1d/EAF-2007 computational SDDR at different cooling times for the first FNG irradiation campaign

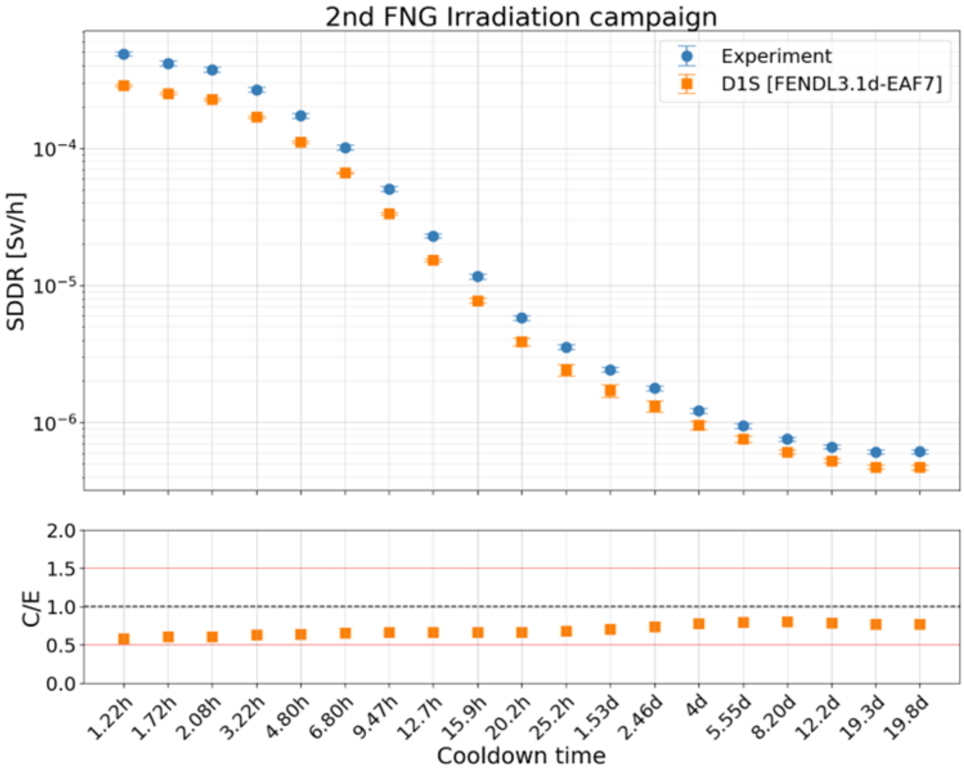
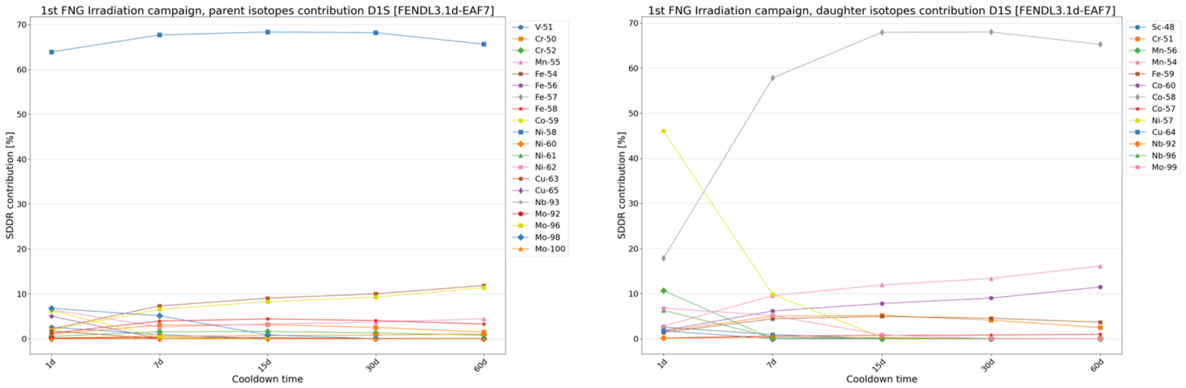
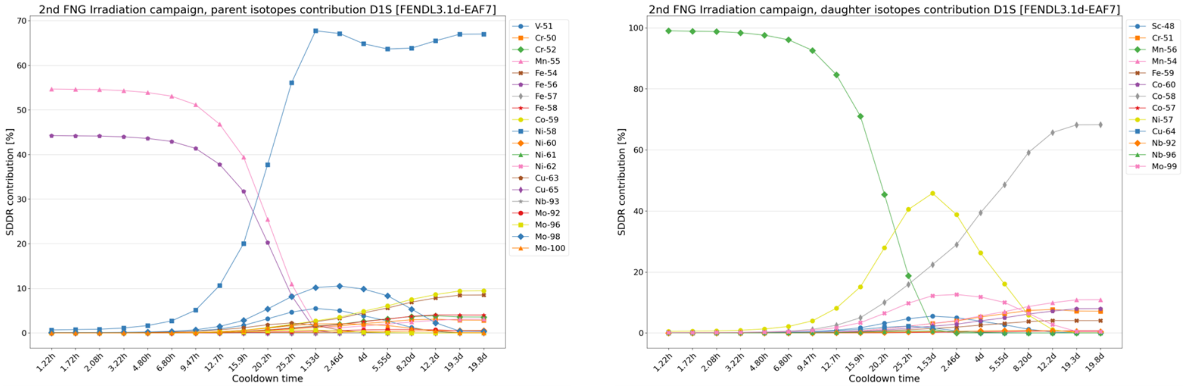


Figure 3.27: Comparison between experimental and FENDL-3.1d/EAF-2007 computational SDDR at different cooling times for the first FNG irradiation campaign



(a) Parents contribution **(b)** Daughters contribution

Figure 3.28: Percentage contribution of parent/daughter isotopes to the total SDDR value during the first FNG irradiation campaign



(a) Parents contribution **(b)** Daughters contribution

Figure 3.29: Percentage contribution of parent/daughter isotopes to the total SDDR value during the second FNG irradiation campaign

To complete the set of results, JADE also provides plots where the percentage contribution to the total SDDR are computed for each parent and daughter isotope considered during the simulation. Parent isotopes are the ones that are part of the original materials and are transmuted into the daughter ones by the irradiating neutrons. These daughters are the decaying isotopes that actually produce the secondary photons which are ultimately responsible for the dose rate. These computations are reported in Fig. 3.28 and in Fig. 3.29 respectively for the first and second irradiation campaign.

Chapter 4

Conclusion

4.1 Summary and final discussion

This dissertation describes JADE, a novel python-based and open source software that was developed jointly by NIER ingegneria, University of Bologna and Fusion For Energy to address a lack of standardization and automation in the field of nuclear data libraries Verification & Validation.

After introducing some basic concepts of particle transport theory, Monte Carlo transport codes and nuclear fusion, it has been discussed how nuclear data libraries are instrumental for the field of particles and radiation transport and how their production chain is fairly complex. It has been explained how an increase of standardization and automation of the V&V procedures to be performed on the libraries could greatly help in improving their quality and speed up their release cycle. It has also been argued how this is especially true for nuclear data libraries focused on fusion application, due to the fact that the majority of the V&V efforts have been concentrated on fission-related quantities for the last 70 years and are hence biased towards this kind of applications. One of the most important difference is that fission-dedicated cross sections can count on the definition of integral parameters to simplify their V&V workflow, while fusion must tally a variety of quantities discretized in energy and in space to have a clear estimate of the impact that cross section modifications may have on the transport simulation results (which calls for heavy automation). Additionally, a significant portion of the contributions to the release of new fusion-related libraries is performed "in kind" meaning that the creation of a single tool where to reunite these contributions could significantly help with the issue of effort fragmentation.

JADE aim is to build a general framework for the automatic generation, run and post-processing of nuclear data benchmarks. Currently, MCNP has been implemented as the Monte Carlo code of choice, with one of its patches, D1S-UNED, being used for SDDR calculations. JADE main architecture has been completed and seven computational benchmarks and two experimental ones have been implemented in it. For the moment, these benchmarks are focused on shielding and dose rate computation application, but JADE has been built in order to allow users and future developers to easily implement new ones. Indeed, no additional coding is required to add a computational benchmark to JADE, while only a limited amount is needed to add experimental benchmarks in order to properly customize their post-processing. JADE architecture has been described both on a high level, where the main functionalities are illustrated, and on a low level, where it is explained how these functionalities have been abstracted according to the Object Oriented Programming (OOP) paradigm used during JADE development. The code has been released as open source in GitHub and a detailed documentation has been prepared and it is hosted on ReadTheDocs. The implementation of software developing best practices such as version control, automatic testing and continuous integration increase JADE reliability and will help integrate future contribution coming from new developers.

The main applications of JADE performed in the last couple of years have also been presented. At first, it was shown how JADE was able to re-spot known errors on older releases of the FENDL libraries. This exercise, designed as a proof of concept, ended up uncovering also a few unknown inconsistencies. These early positive results allowed for further developing effort to be put into the code and to convince the IAEA Nuclear Data Section to include JADE developers in the beta testing of new FENDL releases. It has been shown how JADE has been a valuable asset during the release of the FENDL-3.2 library (the ITER project reference library), helping to individuate some inconsistencies that ultimately led to the FENDL-3.2b version. Moreover, it also has been discussed how JADE has been used to test the new release of JEFF libraries (v4.0T1) and the potential application of JADE to SDDR-related nuclear data has been presented.

To conclude, it could be argued how JADE reached its goals and demonstrated to have the capability to be an important player in the field of nuclear data libraries V&V (especially for fusion related ones). In addition to the presented results, a proof of this was the invitation to the IAEA technical meeting on the "Compilation of Nuclear Data Experiments for Radiation Characterization”. This meeting reunited around 20 experts from all around the world to discuss needs and advancements in the field of nuclear data Verification and Validation and experimental benchmarks. Among the participating institutions were the Massachusetts Institute of Technology (MIT), Oak Ridge Laboratories, Livermore Laboratories, Fusion For Energy, CEA, UKAEA, Argonne National Laboratories and Nuclear Energy Agency (NEA). In this context, the JADE V&V tool was presented and considered to enter into a standard validation path for XS Libraries future releases. The hope is that the automation and standardization provided by the tool could lead to a reduction of effort fragmentation and speed up the release cycle of nuclear data libraries, while, at the same time, improving their quality.

4.2 Future developments

The main objective of this work was to finalize a solid architecture for JADE and prove the feasibility and utility of the software. Now that this has been achieved, many developments might be foreseen for JADE and are discussed in the following paragraphs.

4.2.1 Short and mid-term developments

Expansion of the benchmarks suite

This is the most obvious short-term goal for the project. Now that the main architecture is in place and its effectiveness has been demonstrated, to fully take advantage of JADE automation capability, effort should be put in place to expand its benchmarks suite. Many other computational benchmarks are already available among the fusion community and, with relative little effort, they could be included in JADE. Indeed, no additional coding is required to do so and only some cleanup of the MCNP inputs and the definition of ad hoc configuration file would be sufficient. Slightly more work will be needed to implement new experimental benchmarks due to the need for additional coding, but many experiments already have been modelled in MCNP and are available today.

Removal of Microsoft Office dependency and migration to Linux operative systems

Linux operative system is arguably a staple in the science development. The fact that JADE cannot be used with Linux systems together with the fact that it requires Microsoft Word and Excel to function is clearly a limiting factor in terms of wider adoption. The main constraint here is the use of Microsoft Excel and Word which forces to use non-portable python packages. This is the major factor impeding to run the code on Linux, since, out of the box, python is a portable language. The Word dependency could be easily solved generating only pdf versions of JADE atlases. Eliminating Excel from the equation will be a little more complex. If the configuration file could be substituted with a .csv version (or maybe a proper GUI in the future), the post-processing outputs and their formatting are more central to JADE usage and more difficult to substitute. Either the same functionalities will need to be implemented using open-source alternatives like *OpenOffice*, or a re-structuring of the post-processing features will need to be foreseen for the Linux version.

Implementation of open-source Monte Carlo codes

There are at least two good reasons to implement an additional open-source Monte Carlo code into JADE. The first one is that the tool could expand its field of application, in addition to just comparing nuclear data libraries it would also be possible to benchmark different codes against each other (using the same library). The second is that it would allow to obtain a fully open source project, which would boost adoption. The most promising code candidate to be implemented would probably be OpenMC [94] due to its recent increase in popularity and performance and also due to the fact that it already foresees a python API to interact with its inputs and outputs.

4.2.2 A new philosophy for nuclear data libraries V&V

The final and probably most ambitious development that is discussed here is not related solely to JADE, but to the nuclear data libraries industry as a whole. What it is going to be proposed is almost a new philosophy of work, one that is based on three pillars of modern software developing: open source, continuous integration & development and machine learning.

Open source

The open-source model is a model that is based on sharing a project source code freely in such a way that modification and redistribution are allowed. This decentralized software development model encourages open collaboration and can be of vital importance in fields where the available manpower is limited. JADE is already an open source project and it is released on GitHub, which is arguably the most important and used repository for open source in the world. Additionally, a great effort has been put into producing a detailed documentation for the project which is also open-source and hosted online. This grants that users and new developers alike can more easily get acquainted with JADE. Another piece of open source on which JADE has built on is CoNDERC from which come the original Oktavian MCNP inputs and experimental data (see Section 2.3.7).

Unfortunately, this is not enough and many JADE key components are not open source yet. As discussed in the previous section, in parallel with a particularly restricted access software like MCNP, some other open access alternatives such as OpenMC should be implemented in JADE. Similarly, it has been discussed the importance of eliminating the Microsoft (Windows and Office) dependencies in order to remove all obstacles that impede to port JADE on a open source OS such as Linux. Finally, a number of benchmarks included in JADE are based on SINBAD inputs, which is a restricted access database, meaning that such benchmarks cannot be freely distributed together with JADE. Only when these last open points will be successfully addressed, JADE will be considered an entirely open source project.

Continuous integration and deployment

A fully open source JADE is mandatory in order to start discussing the second pillar: Continuous Integration (CI) and Continuous Deployment (CD). What happens today is that new releases of data libraries are made periodically, usually every few years. A new release is a big deal and small modifications or bug corrections may need to wait if they are not sufficient in number to justify the effort to go through a release cycle. To adopt CI and CD would make the process more agile.

If nuclear data libraries were to move to more structured repositories like GitHub, this would be possible, and, for instance, the FENDL project is already experimenting with that. As an hypothetical case study the following could be considered. A team is working on a particular issue which have very specific needs in terms of some cross sections. They may work on it, hopefully improving the evaluation and push the modified cross sections to the main branch of FENDL. This would trigger an automatic workflow that runs JADE in cloud (hence the need for complete open source) and only if some tests based on JADE results are passed, the modifications are merged into the official FENDL libraries and immediately deployed to all users (an additional check from a FENDL representative may be added). This would dramatically speed up the libraries release cycle and reduce the effort necessary for it.

The key challenge to implement something like this would be to define what exactly are the tests to be performed and when these tests can be considered as passed. There are easy checks like verifying that all tallied quantities of interest present non-negative values, but this clearly do not tell the full story. As discussed in Section 1.3.2, there are no integral parameters readily available for fusion applications and the tally of interest are many and discrete in space and energy. But this is exactly why a software like JADE was conceptualized in the first place: to answer with heavy automation and 'brute force' to this kind of issues.

Machine learning

If a solution to the 'acceptance test' was to be found, Machine Learning (ML), the third pillar, could start to be a real possibility for nuclear data libraries. Recently, the efforts to port Monte Carlo transport codes from a CPU to a GPU architecture have multiplied and it is expected to be completed in the near future. This could decrease the simulation run time by a few order of magnitudes and this speed could be leveraged in an automated framework like JADE to run and post-process a number of simulation which is today inconceivable. This would allow to introduce machine learning algorithms in the evaluation process of the cross sections.

The developing of something on this line has already started but, once again, focused on fission applications. The project is called NucML [95], and it affirms to be the *"only end-to-end python-based supervised machine learning pipeline for enhanced bias-free nuclear data generation and evaluation"*. Having in mind the production chain discussed in Section 1.1.3, NucML framework collects all available and applicable dataset for a specific cross section and then train a variety of ML models on them. In practice, this means that evaluated cross sections are generated by the ML models which are then used to run integral benchmarks. From this kind of benchmarks it is easy to derive quality metrics (i.e., how well the library has performed) that are fed back to the ML model in order to train it. These algorithms may be able to leverage data from well explored isotope and reaction-channel pairs and learn patterns and behaviors that can be applied to other less measured isotopes. The framework is not meant to completely replace the evaluator but only to enhance its analytical insight.

An integration with NucML would be possible where the part related to data compilation, feature extraction and ML model design and training is kept as it is, while the portion related to the integral benchmarks is substituted with JADE benchmarks suite, where the quality metrics to be used for the ML models training would be strictly dependant from the the acceptance criteria defined for the CI and CD. This type of integration would be greatly helped by the fact that both projects are python based and open source.

Bibliography

[1] Davide Laghi. *JADE code repository.* 2022. URL: https://github.com/dodu94/JADE.

[2] Davide Laghi. *JADE official documentation.* 2022. URL: https://jade-a-nuclear-data-libraries-vv-tool.readthedocs.io/en/latest/.

[3] Davide Laghi, Marco Fabbri, Lorenzo Isolan, Raul Pampin, Marco Sumini, A. Portone, and Andrej Trkov. “JADE, a new software tool for nuclear fusion data libraries verification & validation”. In: *Fusion Engineering and Design* 161 (Dec. 2020), p. 112075. DOI: 10.1016/j.fusengdes.2020.112075.

[4] Davide Laghi, Marco Fabbri, Lorenzo Isolan, Raul Pampin, Marco Sumini, G. Shnabel, and Andrej Trkov. “Application of JADE V&V capabilities to the new FENDL v3.2 beta release”. In: *Nuclear Fusion* 61 (2021), p. 116073.

[5] Laghi D., Fabbri M., La Rovere S., Isolan L., Pampin R., Portone A., and Sumini M. “Status of JADE, an open-source software for nuclear data libraries V&V”. In: *Fusion Engineering and Design* 187 (2022), p. 113380.

[6] Laghi D., M. Fabbri, Isolan L., and Sumini M. “Progress report on JADE, a new nuclear data libraries V&V test suite”. In: XIVth ITER Neutronics Meeting. Barcelona, Spain, 2020.

[7] Laghi D., M. Fabbri, Isolan L., and Sumini M. “A new V&V philosophy for fusion nuclear data libraries”. In: 27th International Conference on Transport Theory - ICTT27. Bertinoro, Italy, 2022.

[8] Laghi D., M. Fabbri, Isolan L., and Sumini M. “JADE, a novel V&V software for fusion nuclear data libraries”. In: XVth ITER Neutronics Meeting. Barcelona, Spain, 2022.

[9] Laghi D., Fabbri M., La Rovere S., Isolan L., Pampin R., and Sumini A. Portone M. “Status of JADE, an open-source software for nuclear data libraries V&V”. In: 32nd Symposium on Fusion Technology - SOFT. Dubrovnik, Croatia, 2022.

[10] Laghi D., Fabbri M., La Rovere S., Isolan L., Pampin R., and Sumini A. Portone M. “Status of JADE, an open-source software for nuclear data libraries V&V”. In: IAEA - Technical Meeting on the Compilation of Nuclear Data Experiments for Radiation Characterization. Vienna, Austria, 2022.

[11] Victor (Wai Kin) Chan, ed. *Theory and applications of Monte Carlo Simulations.* Croatia: InTech, 2013.

[12] D. E. Raeside. “Monte Carlo principles and applications”. In: *Physics in Medicine & Biology* 21.2 (1976), p. 181. DOI: 10.1088/0031-9155/21/2/001.

[13] Fisher Hans. *A History of the Central Limit Theorem Book.* New York, United States: Springer, 2011.

[14] Alireza Haghighat. *Monte Carlo Methods for Particle Transport.* United States: CRC Press, 2020.

[15] H. Feshbach and V. F. Weisskopf. “A Schematic Theory of Nuclear Cross Sections”. In: *Physical Review* 76.11 (1949).

[16] R. J. Moore. “Nuclear Cross-Section Theory”. In: *Reviews of Modern Physiscs* 33.1 (1960), pp. 101–116.

[17] Lee A. Bernstein, David A Brown, Arjan J. Koning, Bradley T. Rearden, Catherine E. Romano, Alejandro A. Sonzogni, Andrew S. Voyles, and Walid Younes. “Our Future Nuclear Data Needs”. In: *Annual Review of Nuclear and Particle Science* 69 (2019), pp. 109–136.

[18] R. Capote, D. L Smith, and A. Trkov. “Our Future Nuclear Data Needs”. In: *EPJ Web of Conferences* 8 (2010), p. 04001.

[19] Members of the Cross Sections Evaluation Working Group. *ENDF-6 Formats Manual.* Ed. by H. Herman and A. Trkov. United States: Brookhaven National Laboratory (BNL) National Nuclear Data Center, 2009.

[20] Robert Macfarlane, Douglas W. Muir, R. M. Boicourt, Albert Comstock Kahler III, and Jeremy Lloyd Conlin. “The NJOY Nuclear Data Processing System, Version 2016”. In: (2017).

[21] Los Alamos National Laboratories. *MCNP USER’S MANUAL Code Version 6.2.* 2017.

[22] David H. Laidlaw, W. Benjamin Trumbore, and John F. Hughes. “Constructive Solid Geometry for Polyhedral Objects”. In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques.* New York, NY, USA: Association for Computing Machinery, 1986, pp. 161–170. DOI: 10.1145/15922.15904.

[23] D W Muir. *Gamma rays, Q-values, and kerma factors.* LA-6258-MS. Los Alamos Laboratories, Feb. 1976. DOI: 10.2172/7352303.

[24] R. Garcia, M. Garcia, R. Pampin, and J. Sanz. “Status of reliability in determining SDDR for manual maintenance activities in ITER: Quality assessment of relevant activation cross sections involved”. In: *Fusion Engineering and Design* 112 (2016), pp. 177–191. ISSN: 0920-3796. DOI: https://doi.org/10.1016/j.fusengdes.2016.08.016.

[25] Z. B. Alfassi. *Activation Analysis.* United States: CRC Press, 1990.

[26] L Petrizzi, P Batistoni, S Migliori, Y Chen, U Fischer, P Pereslavtsev, M Loughlin, and A Secco. “Two computational approaches for Monte Carlo based shutdown dose rate calculation with applications to the JET fusion machine”. In: *Proceedings of International conference on supercomputing in nuclear applications SNA’2003.* 2003.

[27] M. Fleming, T. Stainer, and M. Gilbert. *The FISPACT-II User Manual.* UKAEA. Culham, 2018.

[28] P. Sauvan, R. Juarez, J.P. Catalan, F. Ogando, and J. Sanz. “Development of D1SUNED Package for shutdown dose rate calculations”. In: Xth ITER Neutronics Meeting. Cadarache, France, 2015.

[29] Richard F Post. “Nuclear fusion”. In: *Annual Review of Energy* 1.1 (1976), pp. 213–255.

[30] Cyriel Wagemans, ed. *The nuclear fission process.* London, UK: CRC press, 1991.

[31] Dilhan Ezer and AGW Cameron. “The evolution of hydrogen-helium stars”. In: *Astrophysics and Space Science* 14.2 (1971), pp. 399–421.

[32] R Aymar, P Barabaschi, and Y Shimomura. “The ITER design”. In: *Plasma physics and controlled fusion* 44.5 (2002), p. 519.

[33] LA Artsimovich. “Tokamak devices”. In: *Nuclear Fusion* 12.2 (1972), p. 215.

[34] H Abu-Shawareb, R Acree, P Adams, J Adams, B Addis, R Aden, P Adrian, BB Afeyan, M Aggleton, L Aghaian, et al. “Lawson criterion for ignition exceeded in an inertial fusion experiment”. In: *Physical Review Letters* 129.7 (2022), p. 075001.

[35] X. et al Litaudon. “Overview of the JET results in support to ITER”. In: *Nuclear Fusion* 57.10 (2017), p. 102001.

[36] Lyman Spitzer Jr. “The stellarator concept”. In: *The Physics of Fluids* 1.4 (1958), pp. 253–264.

[37] J. Nührenberg, W. Lotz, P. Merkel, C. Nührenberg, U. Schwenn, E. Strumberger, and C. Hayashi. “Overview on Wendelstein 7-X Theory”. In: *Fusion Technology* 27.3T (1995), pp. 71–78.

[38] R Betti and OA Hurricane. “Inertial-confinement fusion with lasers”. In: *Nature Physics* 12.5 (2016), pp. 435–448.

[39] G. H. Miller, E. I. Moses, and C. R. Wuest. “The National Ignition Facility”. In: *Optical Engineering* 43.12 (204).

[40] Ronald C Kirkpatrick, Irvin R Lindemuth, and Marjorie S Ward. “Magnetized target fusion: An overview”. In: *Fusion Technology* 27.3 (1995), pp. 201–214.

[41] General Fusion inc. *website.* 2022. URL: `https://generalfusion.com`.

[42] M. Keilhacker, A. Gibson, C. Gormezano, and P.H. Rebut. “The scientific success of JET”. In: *Nuclear Fusion* 41.12 (2001), pp. 1925–1966.

[43] Jack PC Kleijnen. “Verification and validation of simulation models”. In: *European journal of operational research* 82.1 (1995), pp. 145–162.

[44] I. Kodeli, G. Žerovnik, and Milocco A. “Examples of Use of SINBAD Database for Nuclear data and Code Validation”. In: *EPJ Web of Conferences* 153 (2017), p. 02010.

[45] M Herman. *ENDF-6 Formats Manual Data Formats and Procedures for the Evaluated Nuclear Data File ENDF/B-VI and ENDF/B-VII.* Tech. rep. Brookhaven National Lab.(BNL), Upton, NY (United States), 2009.

[46] E. Cullen. *PREPRO 2019.* NDS-229. Vienna: IAEA, 2019.

[47] Kenichi Tada, Yasunobu Nagaya, Satoshi Kunieda, Kenya Suyama, and Tokio Fukahori. "Development and verification of a new nuclear data processing system FRENDY". In: *Journal of Nuclear Science and Technology* 54 (2017), pp. 806–817.

[48] Bret R. Beck. "FUDGE: a Program for Performing Nuclear Data Testing and Sensitivity". In: *AIP Conference Proceedings* 769 (2005), p. 503.

[49] Ping Liu, Xiaofei Wu, Zhigang Ge, Songyang Li, Haicheng Wu, Lili Wen, Wenming Wang, and Huanyu Zhang. "Progress on China nuclear data processing code system". In: *EPJ Web of Conferences* 146 (2017), p. 02004.

[50] "Continuous Integration and Deployment Software to Automate Nuclear Data Verification and Validation". In: *Nuclear Data Sheets* 118 (2014), pp. 422–425.

[51] V. Zerkin. *MyENDF - web tool for ENDF evaluators.* 2013. URL: `https://www-nds.iaea.org/exfor/myendf.htm`.

[52] C.J. Diez, F. Michel-Sendis, O. Cabellos, M. Bossant, and N. Soppera. "NDEC: A NEA platform for nuclear data testing, verification and benchmarking". In: *EPJ Web of Conferences* 146 (2017), p. 02026.

[53] Brian Triplett, Morgan White, and Samim Anghaiea. "Development of a Test System for Verification and Validation of Nuclear Transport Simulations". In: International Conference on the Physics of Reactors. 2008.

[54] R Lindau, A Möslang, M Rieth, M Klimiankou, E Materna-Morris, A Alamo, A-AF Tavassoli, C Cayron, A-M Lancha, P Fernandez, et al. "Present development status of EUROFER and ODS-EUROFER for application in blanket concepts". In: *Fusion Engineering and Design* 75 (2005), pp. 989–996.

[55] R Zanino, D Bessette, and L Savoldi Richard. "Quench analysis of an ITER TF coil". In: *Fusion Engineering and Design* 85.5 (2010), pp. 752–760.

[56] R.A. Forrest, R. Capote, N. Otsuka, T. Kawano, A.J. Koning, S. Kunieda, J-Ch. Sublet, and Y. Watanabe. *FENDL-3 Library Summary Documentation.* (INDC)NDS-0628. Vienna: IAEA, 2012.

[57] Cross Section Evaluation Working Group (CSEWG). *ENDF/B-VIII.0 evaluated nuclear reaction data library.* 2018. URL: `https://www.nndc.bnl.gov/endf/b8.0/`.

[58] A.J.M. Plompen and et al. "The joint evaluated fission and fusion nuclear data library, JEFF-3.3". In: *European Physical Journal* 56 (2011), p. 181.

[59] K. Shibata and et al. "JENDL-4.0: A New Library for Nuclear Science and Engineering". In: *Journal of Nuclear Science and Technology* 48 (2011), pp. 1–30.

[60] A.J. Koning, D. Rochman, J. Sublet, N. Dzysiuk, M. Fleming, and S. Van der Marck. “TENDL: Complete Nuclear Data Library for Innovative Nuclear Science and Technology”. In: *Nuclear Data Sheets* 155 (2019), pp. 1–55.

[61] Blokhin A.I., Gai E.V., Ignatyuk A.V., Koba I.I., Manokhin V.N., and Pronyaev V.G. *NEW VERSION OF NEUTRON EVALUATED DATA LIBRARY BROND-3.1.* 2022. URL: `https://vant.ippe.ru/en/year2016/2/neutron-constants/1150-5.html`.

[62] V. Goulo. *Fusion Evaluated Nuclear Data Library (FENDL).* INDC(NDS)-223/GF. Vienna: IAEA, 1989.

[63] S. Ganesan and P.K McLaughlin. *FENDL/E Evaluated nuclear data library of neutron nuclear interaction cross-sections and photon production cross-sections and photon-atom interaction cross sections for fusion applications.* INDC(NDS)-128. Vienna: IAEA, 1994.

[64] M. Herman and A.B Pashchenko. *Report on an IAEA Advisory Meeting on Extension and Improvement of the FENDL Library for Fusion Applications (FENDL2).* INDC(NDS)-373. Vienna: IAEA, 1997.

[65] D. Lopez Aldama and A. Trkov. *FENDL-2.1, Update of an evaluated nuclear data library for fusion applications.* INDC(NDS)-467. Vienna: IAEA, 2004.

[66] A. Moeslang, V. Heinzel, H. Matsui, and M. Sugimoto. “The IFMIF test facilities design”. In: *Fusion Engineering and Design* 81.8 (2006). Proceedings of the Seventh International Symposium on Fusion Nuclear Technology, pp. 863–871. ISSN: 0920-3796. DOI: `https://doi.org/10.1016/j.fusengdes.2005.07.044`. URL: `https://www.sciencedirect.com/science/article/pii/S0920379605005995`.

[67] *Stack Overflow Trends.* 2022. URL: `shorturl.at/nvy89`.

[68] *xlwings project web-page.* 2022. URL: `https://www.xlwings.org`.

[69] *openpyxl html documentation.* 2022. URL: `https://openpyxl.readthedocs.io/en/stable`.

[70] *Git project web-page.* 2022. URL: `https://git-scm.com`.

[71] *sphinx online documentation.* 2022. URL: `https://www.sphinx-doc.org/en/master`.

[72] *Pytest project web-page.* 2022. URL: `https://docs.pytest.org`.

[73] *coverage package documentation.* 2022. URL: `https://coverage.readthedocs.io`.

[74] E. Sartori. *VITAMIN-J, a 175 Group Neutron Cross Section Library Based on JEF-1 for Shielding Benchmark Calculations.* JEF-l/Doc-100. NEA, 1985.

[75] M. Sawan. *FENDL Neutronics Benchmark: Specifications for the calculational and shielding benchmark.* INDC(NDS)-316. Vienna: IAEA, 1994.

[76] H. Chohan. *Generation of 1D TBM WCLL and HCPB MCNP computational models.* Memo [2JMRZR v1.0]. F4E, 2019.

[77] D. Leichtle, B. Colling, M. Fabbri, Juarez R., M. Loughlin, R. Pampin, Serikov A. Polunovskiy E., A. Turner, and Bertalot L. “The ITER tokamak neutronics reference model C-Model”. In: *Fusion Engineering and Design* 136 (2018), pp. 742–746.

[78] R. Juarez, G. Pedroche, M. J. Loughlin, R. Pampin, P Martinez, M. De Pietri, J. Alguacil, F. Ogando, P. Sauvan, A. J. Lopez-Revelles, A. Kolšek, E. Polunovskiy, M. Fabbri, and J. Sanz. “A full and heterogeneous model of the ITER tokamak for comprehensive nuclear analyses”. In: *Nature Energy* 6 (2021), pp. 150–157.

[79] E. Polunovskiy. *Description of ITER Nuclear Analysis Tokamak Reference Model C-model R181031.* Technical Report [XETSWC v1.5]. Iter Organization, 2019.

[80] M. Youssef, Feder R., Batistoni P., Fischer U., Jakhar S., Konno C., Loughlin M., and Villari R. “Benchmarking of the 3-D CAD-based Discrete Ordinates code “ATTILA” for dose rate calculations against experiments and Monte Carlo calculations”. In: *Fusion Engineering and Design* 88 (2013), pp. 3033–3040.

[81] R. Pampin, A. Davis, J. Izquierdo, D. Leichtle, M.D. Loughlin, J. Sanz, A. Turner, R. Villari, and P.P.H. Wilson. “Developments and needs in nuclear analysis of fusion technology”. In: *Fusion Engineering and Design* 88 (2013), pp. 454–460.

[82] A. Milocco, A. Trkov, and I. A. Kodeli. “The OKTAVIAN TOF experiments in SINBAD: Evaluation of the experimental uncertainties”. In: *Annals of Nuclear Energy* 37 (2010), pp. 443–449.

[83] IAEA Nuclear Data Section. *CoNDERC.* 2021. URL: `https://www-nds.iaea.org/conderc/`.

[84] M. Martone, M. Angelone, and M. Pillon. “The 14 MeV Frascati neutron generator”. In: *Journal of Nuclear Materials* 212-215 (1994). Fusion Reactor Materials, pp. 1661–1664. DOI: `https://doi.org/10.1016/0022-3115(94)91109-6`.

[85] P. Batistoni, M. Angelone, L. Petrizzi, and M. Pillon. “Benchmark Experiment for the Validation of Shut Down Activation and Dose Rate in a Fusion Device”. In: *Journal of Nuclear Science and Technology* 39.sup2 (2002), pp. 974–977.

[86] K. Seidel, Y. Chen, U. Fischer, H. Freiesleben, D. Richter, and S. Unholzer. “Measurement and analysis of dose rates and gamma-ray fluxes in an ITER shut-down dose rate experiment”. In: *Fusion Engineering and Design* 63-64 (2002), pp. 211–215.

[87] R. Pampin, A. Davis, R.A. Forrest, D.A. Barnett, I. Davis, and M.Z. Youssef. “Status of novel tools for estimation of activation dose”. In: *Fusion Engineering and Design* 85.10 (2010). Proceedings of the Ninth International Symposium on Fusion Nuclear Technology, pp. 2080–2085.

[88] J. Sanz, O. Cabellos, and N. Garcia-Herranz. *Inventory Code for Nuclear Applications: User’s Manual V. 2008.* RSICC". 2008.

[89] R. Villari, P. Sauvan, D.Flammini, and L. Petrizzi. *Deliverable FPA-395-02 D4: Validation and updated activation libraries for D1S method (IDM F4E D 2AJN42 v1.1).* Technical Report. ENEA/UNED, 2019.

[90] L. Packer and A. Trkov. *Summary Report from the Consultants' Meeting IAEA Headquarters.* INDC(NDS)-0769. Vienna: IAEA, 2018.

[91] G. Schnabel et al. "TBD". In: *Nuclear Data Sheets* TBD (under submission), TBD.

[92] J-Ch. Sublet, R. A. Forrest, and J. Kopecky. "The EAF-2007 Neutron, Deuteron, and Proton Activation Libraries". In: *Nuclear Technology* 168 (2009), pp. 279–283.

[93] T. Eade, B. Colling, J. Naish, L.W. Packer, and A. Valentine. "Shutdown dose rate benchmarking using modern particle transport codes". In: *Nuclear Fusion* 60 (2020), p. 056024.

[94] Paul K. Romano, Nicholas E. Horelik, Bryan R. Herman, Adam G. Nelson, Benoit Forget, and Kord Smith. "OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development". In: *Annuals of Nuclear Energy* 82 (2015), pp. 90–97.

[95] Pedro Jr. Vicente-Valdez and Massimiliano Fratoni. *NucML html documentation.* 2022. URL: `https://pedrojrv.github.io/nucml/index.html`.

Appendix A

Coverage report detail

Table A.1 reports an extract of the coverage output for JADE complemented with brief description of the modules and eventual notes on their coverage.

Module	Stat.	Miss.	Cov.	Brief description	Notes
__init__.py	0	0	100%	required for correct imports, it is empty.	-
atlas.py	130	50	62%	Module responsible for the generation of the atlas (not of the plots).	Code related to the actual generation of the .docx and .pdf file could not be tested
computational.py	56	56	0%	Contains functions that link the GUI with the input generation and run.	No critical code and low number of lines
configuration.py	60	2	97%	Handle the parsing of the configuration file.	-
excel_support.py	21	19	10%	Will probably be deprecated in the future.	Code related to .xlsx file could not be tested
expoutput.py	409	409	0%	Handle the post-processing of experimental outputs.	Majority of the code involves the creation of .xlsx and .docx files
gui.py	276	276	0%	Handle the GUI operation.	No critical code here, this module only links input from the users to separately tested functions. The effort to cover this part of the code would be way higher than the actual benefit in terms of reliabilty. Moreover, GUI is constantly traditionally tested because is the only way to send requests to JADE
inputfile.py	239	14	94%	Handle the parsing, editing and creation of MCNP input files.	-
libmanager.py	166	3	98%	Handle all operations regarding the libraries and XSDIR (e.g. translation) .	-
main.py	85	85	0%	Initialize the GUI and JADE session.	This code is tested simply installing JADE and verify that the application starts.

Module	Stat.	Miss.	Cov.	Brief description	Notes
matreader.py	484	29	94%	Parser of the MCNP material card. Handles all operation regarding materials in MCNP inputs.	-
meshtal.py	106	9	92%	Parser of MESHTAL file. This parser is not completed and only contains the minimum amount of parsing necessary for JADE correct functioning.	-
output.py	651	533	18%	Handle the post-processing of general computational benchmarks.	Majority of the code involves the creation of .xlsx and .docx files.
outputFile.py	51	5	90%	Creates a higher-level object that represent all outputs coming from an MCNP run.	-
parsersD1S.py	180	5	97%	Parser for all D1S-UNED input files.	-
plotter.py	436	39	91%	Responsible for the plot generation.	-
postprocess.py	38	38	0%	Contains functions that link the GUI with the post-processing operations.	No critical code and low number of lines,
sphereoutput.py	755	627	17%	Handle the post-processing of the Sphere benchmarks.	Majority of the code involves the creation of .xlsx and .docx files.
status.py	222	8	96%	Allows to update the information about benchmarks that have been run/post-processed in order to allow operations and overview overriding operations.	-
testrun.py	385	81	79%	Creates benchmarks inputs and run them.	The run portion of the code cannot be tested since it requires MCNP.

Module	Stat.	Miss.	Cov.	Brief description	Notes
utilitiesgui.py	176	31	82%	Handles all the utilities that are provided separately in JADE.	Majority of untested code is related to exception raised in case of missing files.
Total	4926	2319	53%		

Table A.1: Summary of JADE code coverage at 23/02/2022

Appendix B

Benchmarks additional details

B.1 Test Blanket Module

Table B.1 describe all the different layers composing the TBM set, Gap and Shielding section in both TBM models. Table B.2 and Table B.3 respectively report the detailed layer description of the HCPB and WCCL TBM set.

Cell	Radius (cm)	Region
51	506.6–834	Plasma
52	834-850.3	Void
53	850.3-850.6	First Wall-Part 1
54	850.6-851.3	First Wall-Part 2 containing cooling channels
55	851.3-853.3	First Wall-Part 3
56 – 59	853.3-918.8	Breeding Zone
60	918.8-946.3	Gap
61	946.3-955.3	Shield- SS316L(N)-IG
62	955.3-958.38	Shield- Water
63	958.38-959.58	Shield- SS316L(N)-IG
64	959.58-961.87	Shield- Water
65	961.87-963.07	Shield- SS316L(N)-IG
66	963.07-965.36	Shield- Water
67	965.36-966.56	Shield- SS316L(N)-IG
68	966.56-968.85	Shield- Water
69	968.85-970.05	Shield- SS316L(N)-IG
70	970.05-972.34	Shield- Water
71	972.34-973.54	Shield- SS316L(N)-IG
72	973.54-975.83	Shield- Water
73	975.83-977.03	Shield- SS316L(N)-IG
74	977.03-979.32	Shield- Water
75	979.32-980.52	Shield- SS316L(N)-IG
76	980.52-982.81	Shield- Water
77	982.81-984.01	Shield- SS316L(N)-IG
78	984.01-986.3	Shield- Water
79	986.3-987.5	Shield- SS316L(N)-IG

Cell	Radius (cm)	Region
80	987.5-989.788	Shield- Water
81	989.788-991.688	Shield- SS316L(N)-IG
82	991.688-993.577	Shield- Water
83	993.577-995.477	Shield- SS316L(N)-IG
84	995.477-997.366	Shield- Water
85	997.366-999.266	Shield- SS316L(N)-IG
86	999.266-1001.155	Shield- Water
87	1001.155-1003.055	Shield- SS316L(N)-IG
88	1003.055-1004.944	Shield- Water
89	1004.944-1006.844	Shield- SS316L(N)-IG
90	1006.844-1008.733	Shield- Water
91	1008.733-1010.633	Shield- SS316L(N)-IG
92	1010.633-1012.522	Shield- Water
93	1012.522-1014.422	Shield- SS316L(N)-IG
94	1014.422-1016.311	Shield- Water
95	1016.311-1018.211	Shield- SS316L(N)-IG
96	1018.211-1020.1	Shield- Water
97	1020.1-1022	Shield- SS316L(N)-IG
98	1022-1025.42857	Shield- Water
99	1025.42857-1026.42857	Shield- SS316L(N)-IG
100	1026.42857-1029.65714	Shield- Water
101	1029.65714-1031.05714	Shield- SS316L(N)-IG
102	1031.05714-1034.08571	Shield- Water
103	1034.08571-1035.48571	Shield- SS316L(N)-IG
104	1035.48571-1038.51428	Shield- Water
105	1038.51428-1039.91428	Shield- SS316L(N)-IG
106	1039.91428-1043.14285	Shield- Water
107	1043.14285-1044.14285	Shield- SS316L(N)-IG
108	1044.14285-1047.57142	Shield- Water
109	1047.57142-1048.57142	Shield- SS316L(N)-IG
110	1048.57142-1052	Shield- Water
111	1052-1053.9	Shield- SS316L(N)-IG
112	1053.9-1055	Shield- Water
113	1055-1056.9	Shield- SS316L(N)-IG
114	1056.9-1058	Shield- Water
115	1058-1059.9	Shield- SS316L(N)-IG
116	1059.9-1061	Shield- Water
117	1061-1062.9	Shield- SS316L(N)-IG
118	1062.9-1064	Shield- Water
119	1064-1065.9	Shield- SS316L(N)-IG
120	1065.9-1067	Shield- Water
121	1067-1068.9	Shield- SS316L(N)-IG
122	1068.9-1070	Shield- Water
123	1070-1071.9	Shield- SS316L(N)-IG
124	1071.9-1073	Shield- Water

Cell	Radius (cm)	Region
125	1073-1074.9	Shield- SS316L(N)-IG
126	1074.9-1076.2	Shield- Water
127	1076.2-1084.2	Shield- SS316L(N)-IG

Table B.1: Detailed description of the TBM Set, Gap and Shielding Section layers

Cell	Radius (cm)	Region
56	853.3000 – 855.4088	Beryllium
57	855.4088 - 859.9088	With z lithium structure
58	859.9088 - 893.3000	With radial lithium structure
59	893.3000 - 918.8000	With pipework

Table B.2: Description of the HCPB TBM set layers

Cell	Radius (cm)	Region
56	853.300 – 854.300	Lithium-Lead
57	854.300 - 862.475	With azimuthal water pipes
58	862.475 - 903.400	With radial water pipes
59	903.400 - 918.800	With pipework

Table B.3: Description of the WCLL TBM set layers

B.2 Oktavian Materials

Table B.4 reports a detailed description of the materials tested with the Oktavian benchmark.

Material	Apparent Density $[g/cm^3]$	Element	Isotope	Mass Fraction	Atom Fraction
Aluminum	1.223	Al	Al-27 [13027]	9.96E-01	9.98E-01
		Si	Si-28 [14028]	1.38E-03	1.33E-03
			Si-29 [14029]	7.24E-05	6.75E-05
			Si-30 [14030]	4.94E-05	4.45E-05
		Fe	Fe-54 [26054]	1.13E-04	5.65E-05
			Fe-56 [26056]	1.84E-03	8.87E-04
			Fe-57 [26057]	4.32E-05	2.05E-05
			Fe-58 [26058]	5.85E-06	2.73E-06
		Cu	Cu-63 [29063]	6.85E-05	2.94E-05
			Cu-65 [29065]	3.15E-05	1.31E-05
Cobalt	1.94	Co	Co-59 [27059]	9.91E-01	9.87E-01
		Zn	Zn-64 [30064]	3.00E-05	2.75E-05
		Ni	Ni-58 [28058]	1.02E-03	1.03E-03
			Ni-60 [28060]	3.93E-04	3.85E-04
			Ni-61 [28061]	1.71E-05	1.65E-05
			Ni-62 [28062]	5.45E-05	5.17E-05
			Ni-64 [28064]	1.39E-05	1.28E-05
		Si	Si-28 [14028]	3.69E-04	7.74E-04
			Si-29 [14029]	1.87E-05	3.80E-05
			Si-30 [14030]	1.23E-05	2.42E-05
		Fe	Fe-54 [26054]	7.01E-05	7.63E-05
			Fe-56 [26056]	1.10E-03	1.16E-03
			Fe-57 [26057]	2.54E-05	2.62E-05
			Fe-58 [26058]	3.38E-06	3.43E-06
		Ca	Ca-40 [20040]	2.91E-03	4.27E-03
			Ca-42 [20042]	1.94E-05	2.72E-05
			Ca-43 [20043]	4.05E-06	5.53E-06
			Ca-44 [20044]	6.27E-05	8.37E-05
			Ca-46 [20046]	1.20E-07	1.53E-07

Material	Apparent Density $[g/cm^3]$	Element	Isotope	Mass Fraction	Atom Fraction
			Ca-48 [20048]	5.61E-06	6.87E-06
		Mn	Mn-55 [25055]	2.00E-03	2.14E-03
		S	S-32 [16032]	7.60E-04	1.40E-03
			S-33 [16033]	6.00E-06	1.07E-05
			S-34 [16034]	3.37E-05	5.82E-05
			S-36 [16036]	1.60E-07	2.61E-07
		Cu	Cu-63 [29063]	6.92E-05	6.45E-05
			Cu-65 [29065]	3.08E-05	2.79E-05
		C	C-0 [6000]	3.00E-04	1.47E-03
		Pb	Pb-204 [82204]	2.80E-07	8.06E-08
			Pb-206 [82206]	4.82E-06	1.37E-06
			Pb-207 [82207]	4.42E-06	1.25E-06
			Pb-208 [82208]	1.05E-05	2.96E-06
Cromium	3.72	Cr	Cr-50 [24050]	4.34E-02	4.51E-02
			Cr-52 [24052]	8.36E-01	8.37E-01
			Cr-53 [24053]	9.48E-02	9.31E-02
			Cr-54 [24054]	2.36E-02	2.27E-02
		Fe	Fe-54 [26054]	9.35E-05	9.01E-05
			Fe-56 [26056]	1.47E-03	1.36E-03
			Fe-57 [26057]	3.39E-05	3.09E-05
			Fe-58 [26058]	4.51E-06	4.05E-06
		C	C-0 [6000]	2.10E-04	9.08E-04
		Si	Si-28 [14028]	6.46E-05	1.20E-04
			Si-29 [14029]	3.28E-06	5.88E-06
			Si-30 [14030]	2.16E-06	3.75E-06
Copper	6.0123	Cu	Cu-63 [29063]	6.85E-01	6.92E-01
			Cu-65 [29065]	3.15E-01	3.08E-01
		S	S-32 [16032]	9.48E-06	1.88E-05
			S-33 [16033]	7.71E-08	1.49E-07
			S-34 [16034]	4.46E-07	8.34E-07
			S-36 [16036]	2.24E-09	3.96E-09
		As	As-75 [33075]	6.00E-07	5.09E-07
		Pb	Pb-204 [82204]	4.13E-09	1.29E-09
			Pb-206 [82206]	7.19E-08	2.22E-08
			Pb-207 [82207]	6.62E-08	2.03E-08
			Pb-208 [82208]	1.58E-07	4.82E-08
		Sb	Sb-121 [51121]	1.14E-07	5.97E-08
			Sb-123 [51123]	8.64E-08	4.47E-08
		Fe	Fe-54 [26054]	1.13E-07	1.33E-07
			Fe-56 [26056]	1.84E-06	2.09E-06
			Fe-57 [26057]	4.32E-08	4.82E-08

Material	Apparent Density $[g/cm^3]$	Element	Isotope	Mass Fraction	Atom Fraction
			Fe-58 [26058]	5.85E-09	6.42E-09
Lithium Fluoride	1.76361	H	H-1 [1001]	1.90E-05	2.46E-04
		Li	Li-6 [3006]	1.71E-02	3.70E-02
			Li-7 [3007]	2.45E-01	4.56E-01
		O	O-16 [8016]	7.11E-05	5.80E-05
		F	F-19 [9019]	7.37E-01	5.06E-01
		Si	Si-28 [14028]	6.62E-05	3.09E-05
			Si-29 [14029]	3.48E-06	1.57E-06
			Si-30 [14030]	2.37E-06	1.03E-06
		Fe	Fe-54 [26054]	2.82E-06	6.83E-07
			Fe-56 [26056]	4.60E-05	1.07E-05
			Fe-57 [26057]	1.08E-06	2.48E-07
			Fe-58 [26058]	1.46E-07	3.29E-08
Manganese	4.36894	Mn	Mn-55 [25055]	1.00E+00	9.99E-01
		C	C-0 [6000]	5.00E-05	2.29E-04
		Si	Si-28 [14028]	1.84E-05	3.61E-05
			Si-29 [14029]	9.66E-07	1.83E-06
			Si-30 [14030]	6.59E-07	1.21E-06
		P	P-31 [15031]	1.00E-06	1.77E-06
		S	S-32 [16032]	2.08E-04	3.58E-04
			S-33 [16033]	1.70E-06	2.83E-06
			S-34 [16034]	9.81E-06	1.59E-05
			S-36 [16036]	4.94E-08	7.54E-08
		Fe	Fe-54 [26054]	1.13E-05	1.15E-05
			Fe-56 [26056]	1.84E-04	1.80E-04
			Fe-57 [26057]	4.32E-06	4.17E-06
			Fe-58 [26058]	5.85E-07	5.55E-07
Molybdenum	2.15	Mo	Mo-92 [42092]	1.42E-01	1.48E-01
			Mo-94 [42094]	9.05E-02	9.25E-02
			Mo-95 [42095]	1.57E-01	1.59E-01
			Mo-96 [42096]	1.67E-01	1.67E-01
			Mo-97 [42097]	9.65E-02	9.55E-02
			Mo-98 [42098]	2.46E-01	2.41E-01
			Mo-100 [42100]	1.00E-01	9.63E-02
Silicium	1.29581	Si	Si-28 [14028]	9.19E-01	9.22E-01
			Si-29 [14029]	4.83E-02	4.68E-02
			Si-30 [14030]	3.29E-02	3.09E-02
		Ti	Ti-46 [22046]	8.23E-02	8.54E-02
			Ti-47 [22047]	7.42E-02	7.54E-02
			Ti-48 [22048]	7.35E-01	7.31E-01
			Ti-49 [22049]	5.40E-02	5.26E-02

Titanium 1.54

Material	Apparent Density $[g/cm^3]$	Element	Isotope	Mass Fraction	Atom Fraction
			Ti-50 [22050]	5.17E-02	4.93E-02
		Mg	Mg-24 [12024]	2.29E-04	4.56E-04
			Mg-25 [12025]	2.90E-05	5.54E-05
			Mg-26 [12026]	3.19E-05	5.86E-05
		Fe	Fe-54 [26054]	4.91E-05	4.34E-05
			Fe-56 [26056]	7.71E-04	6.57E-04
			Fe-57 [26057]	1.78E-05	1.49E-05
			Fe-58 [26058]	2.37E-06	1.95E-06
		N	N-14 [7014]	2.00E-05	6.81E-05
		C	C-0 [6000]	6.00E-05	2.38E-04
		Cl	Cl-35 [17035]	6.36E-04	8.68E-04
			Cl-37 [17037]	2.04E-04	2.63E-04
		H	H-1 [1001]	3.00E-05	1.42E-03
		Mn	Mn-55 [25055]	2.00E-05	1.74E-05
		O	O-16 [8016]	6.10E-04	1.82E-03
Tungsten	4.43	W	W-182 [74182]	2.65E-01	2.68E-01
			W-183 [74183]	1.43E-01	1.44E-01
			W-184 [74184]	3.06E-01	3.07E-01
			W-186 [74186]	2.84E-01	2.81E-01
Zirconium	2.77813	Zr	Zr-90 [40090]	5.07E-01	5.15E-01
			Zr-91 [40091]	1.12E-01	1.12E-01
			Zr-92 [40092]	1.73E-01	1.72E-01
			Zr-94 [40094]	1.79E-01	1.74E-01
			Zr-96 [40096]	2.94E-02	2.80E-02

Table B.4: Detail composition of all materials composing the spheres of the Oktavian benchmark

Appendix C

Additional Results

C.1 MCNP models, Oktavian benchmark

From Fig. C.1 to Fig. C.11 are reported the plots comparing the neutron leakage current obtained computationally and experimentally for the Oktavian benchmarks for different materials. The computational results include MCNP models, FENDL-3.2b, ENDF-VIII.O and JEFF-3.3.

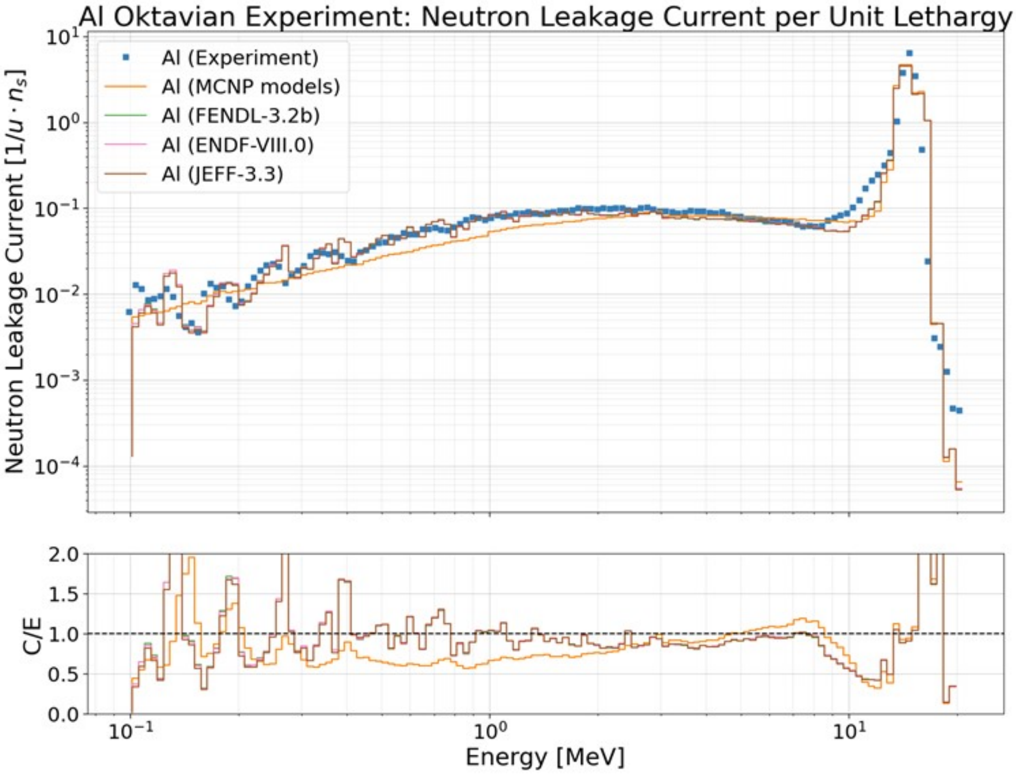


Figure C.1: Oktavian experiment, neutron leakage current for Aluminum

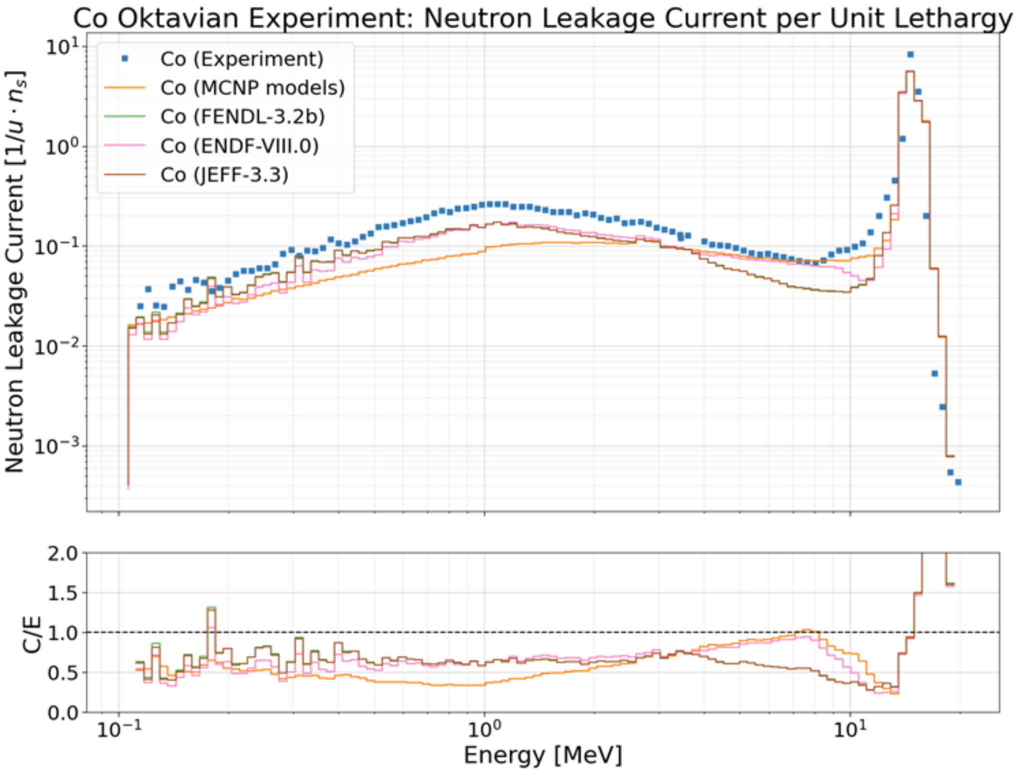


Figure C.2: Oktavian experiment, neutron leakage current for Cobalt

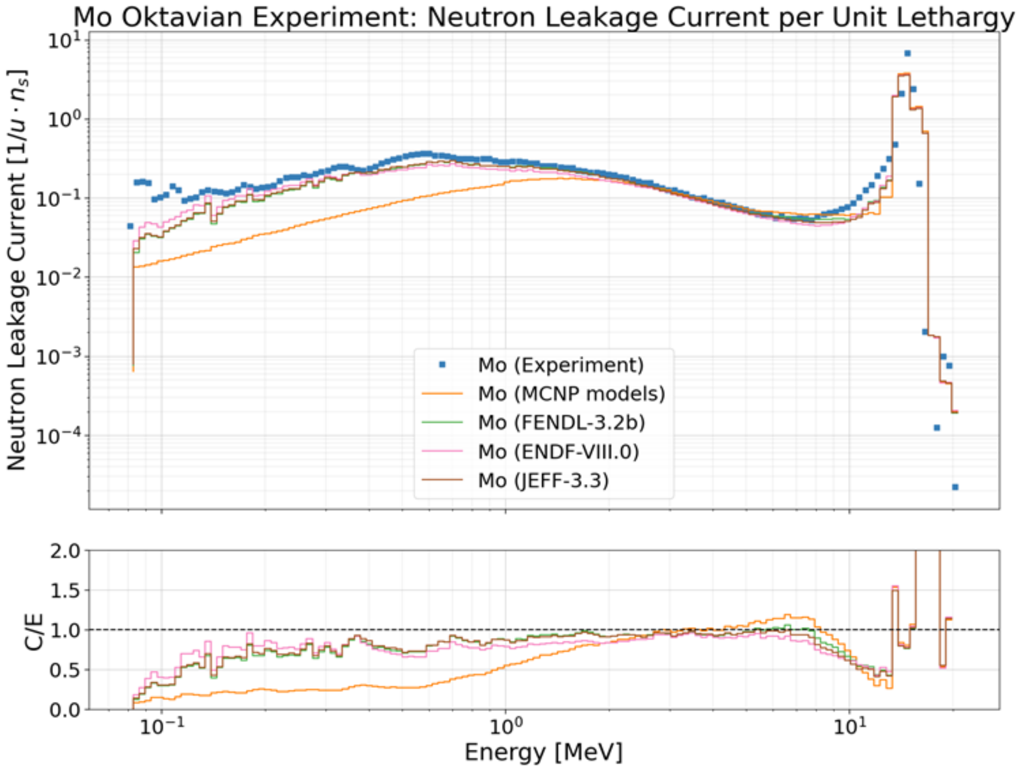


Figure C.3: Oktavian experiment, neutron leakage current for Molybdenum

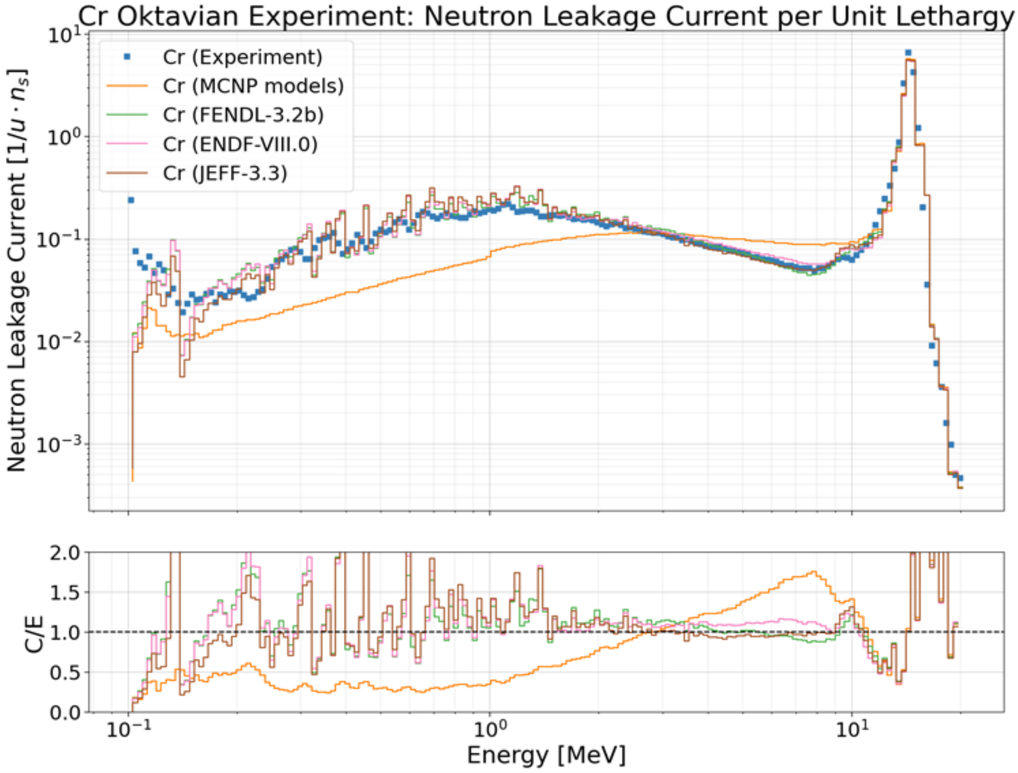


Figure C.4: Oktavian experiment, neutron leakage current for Chromium

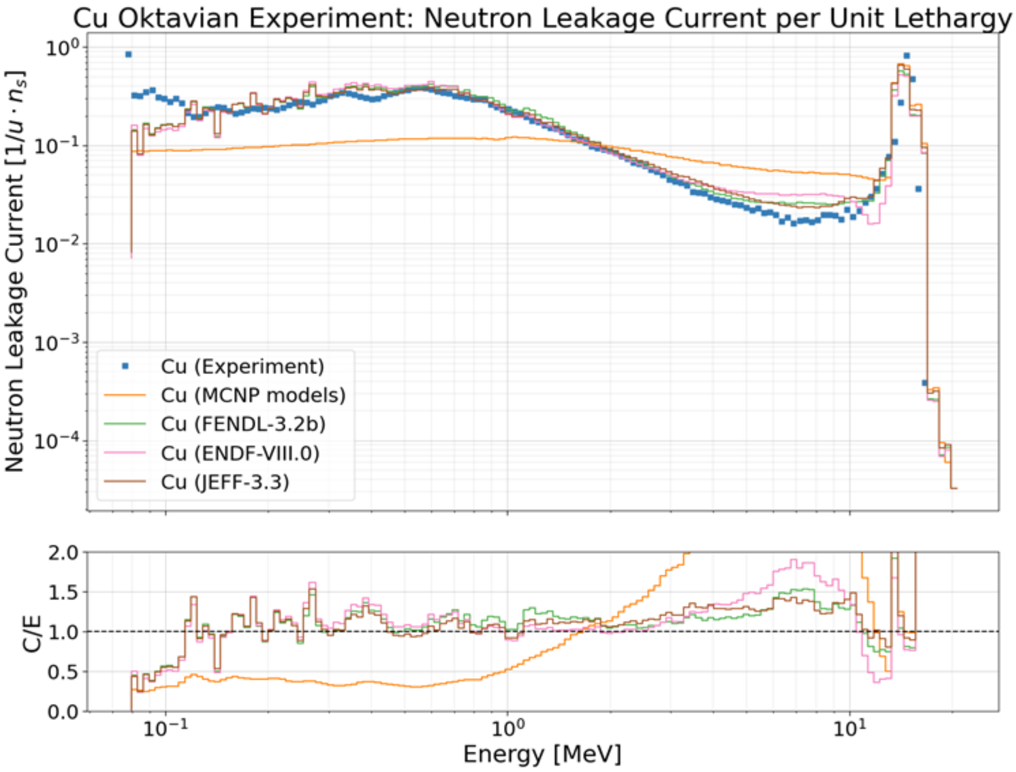


Figure C.5: Oktavian experiment, neutron leakage current for Copper

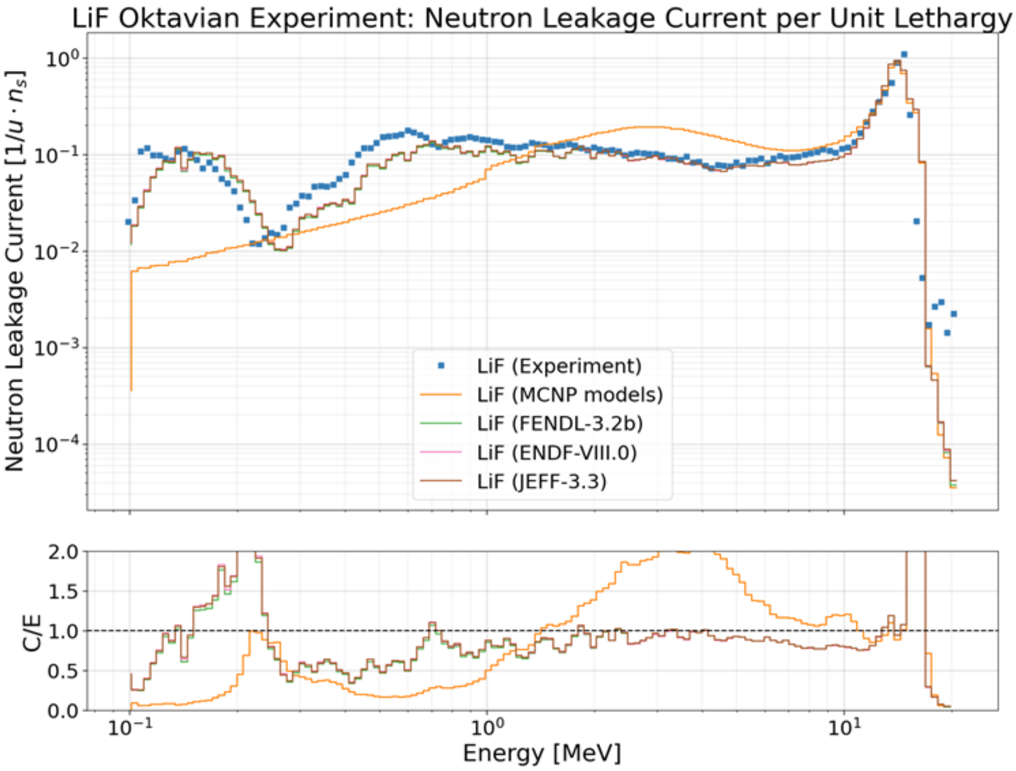


Figure C.6: Oktavian experiment, neutron leakage current for LiFe

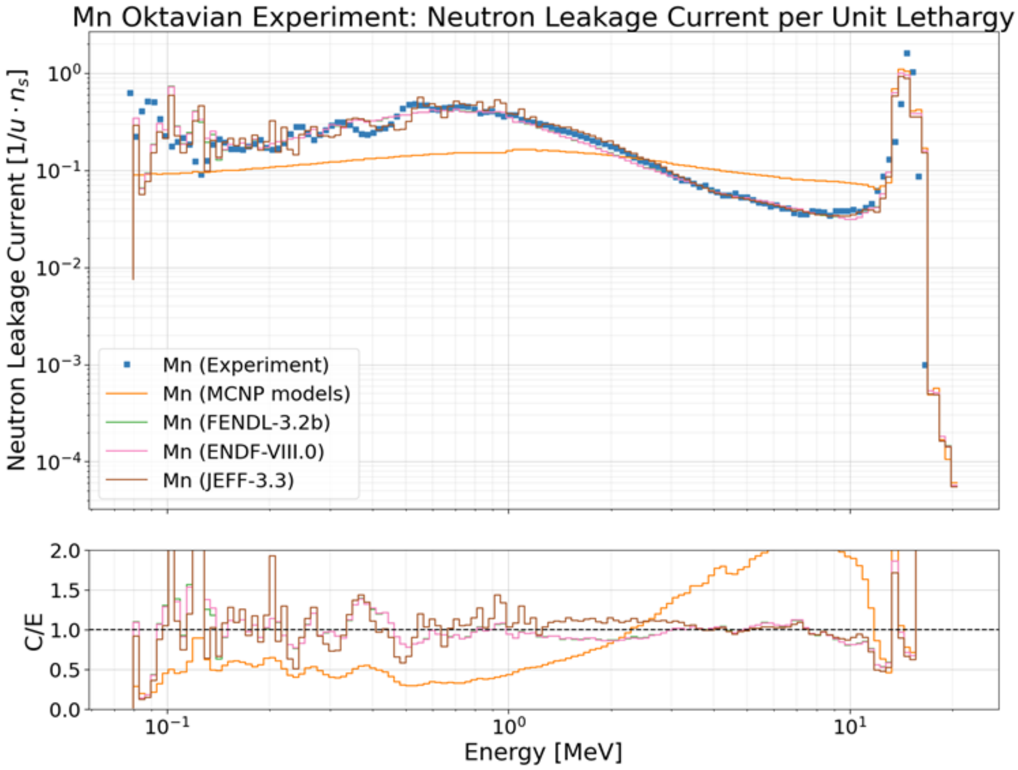


Figure C.7: Oktavian experiment, neutron leakage current for Manganese

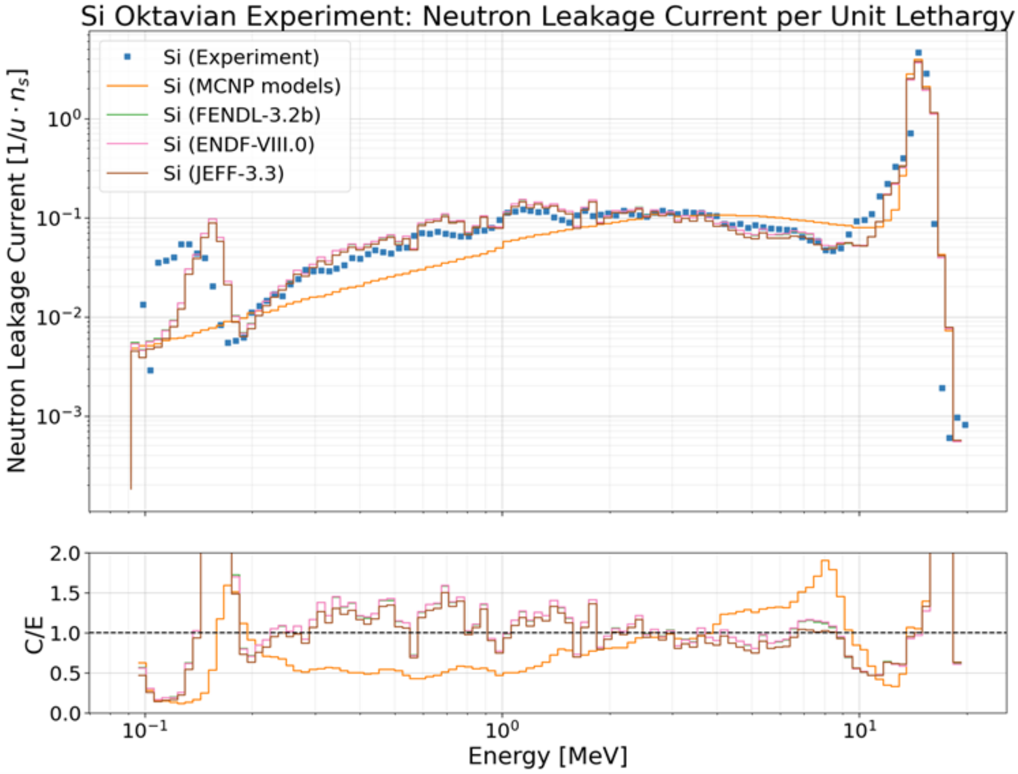


Figure C.8: Oktavian experiment, neutron leakage current for Silica

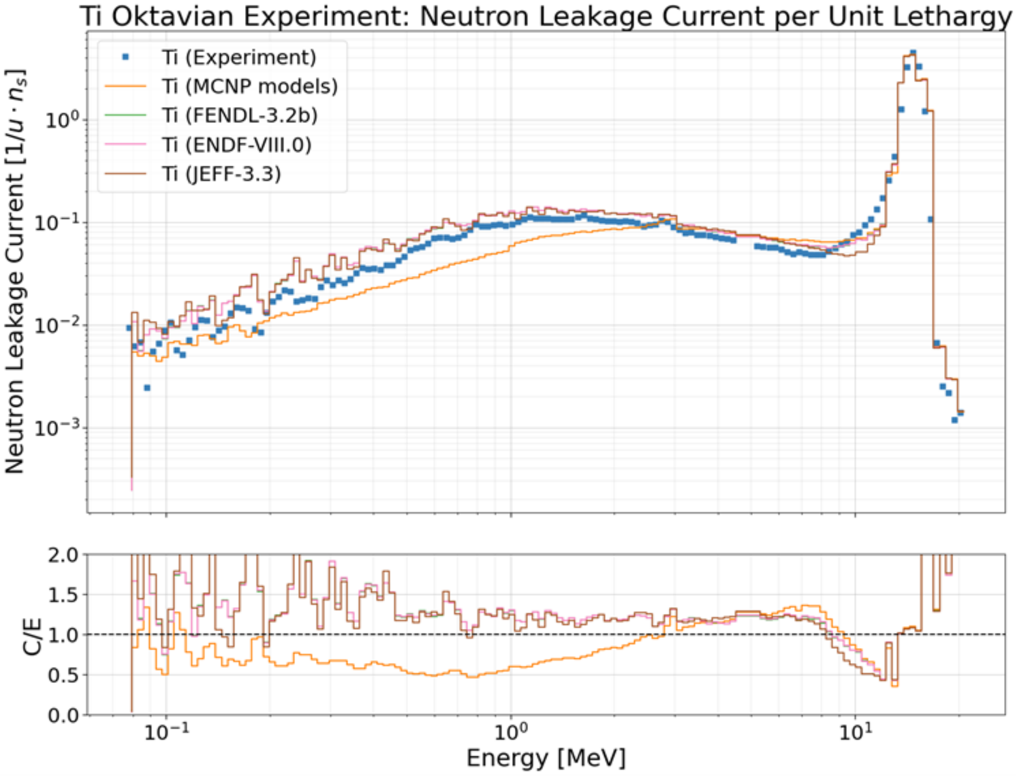


Figure C.9: Oktavian experiment, neutron leakage current for Titanium

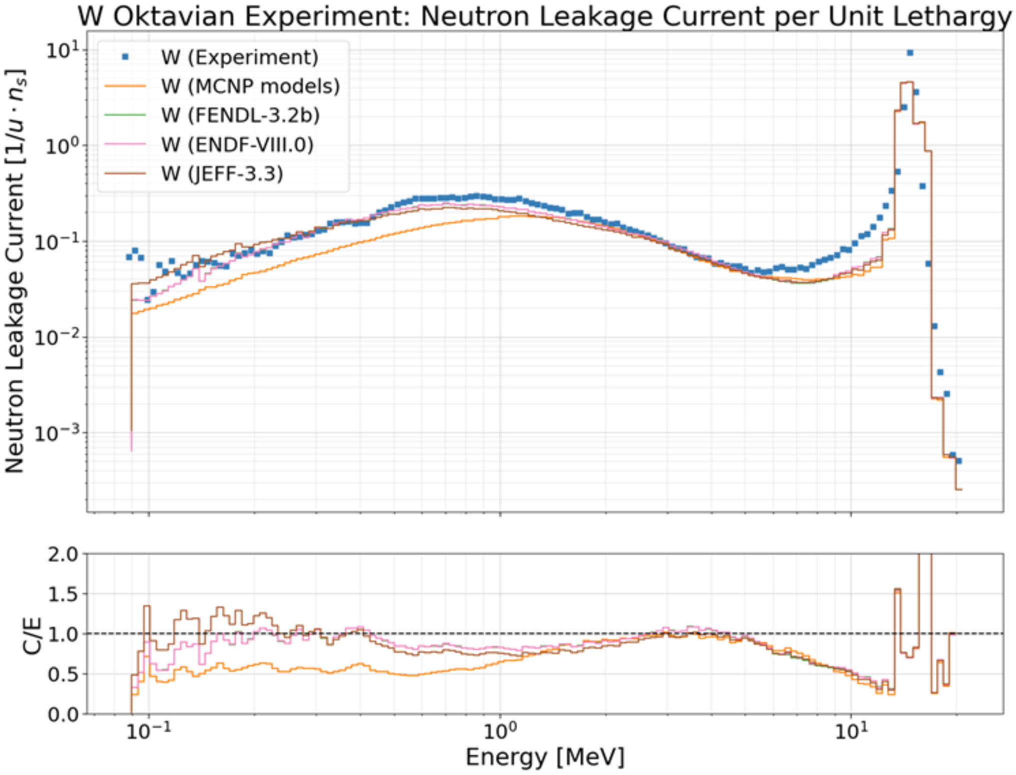


Figure C.10: Oktavian experiment, neutron leakage current for Tungsten

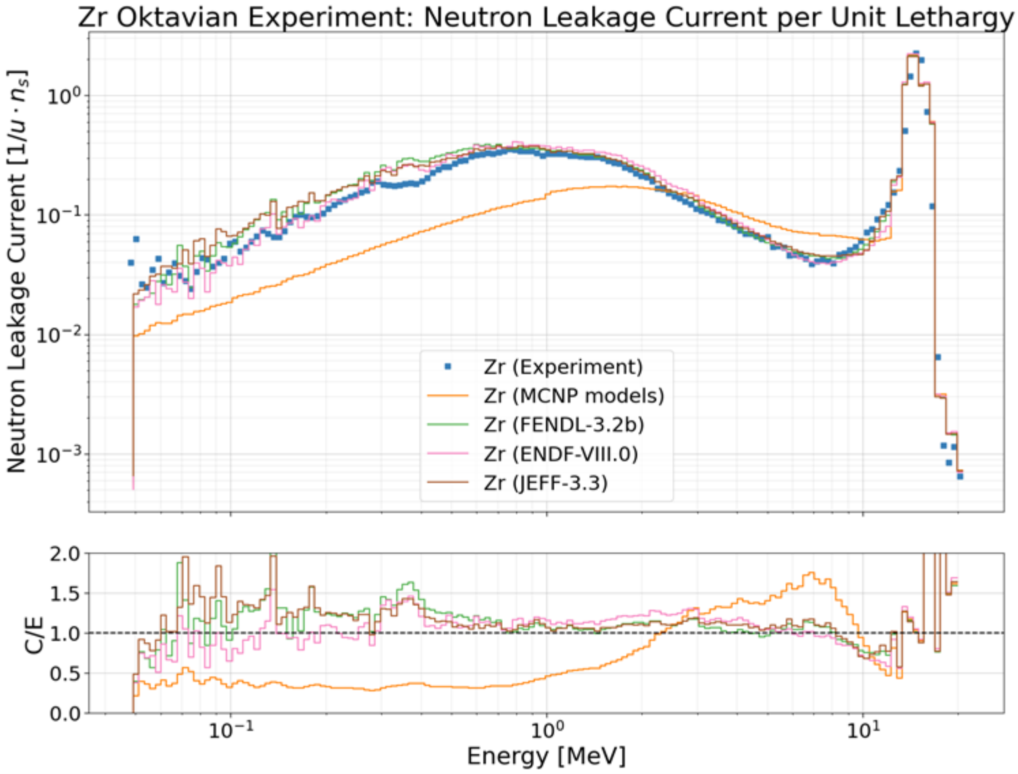


Figure C.11: Oktavian experiment, neutron leakage current for Zirconium

C.2 JEFF-v4.0T1 assessment, Oktavian benchmark

From Fig. C.12 to Fig. C.17 are reported the plots comparing the neutron leakage current obtained computationally and experimentally for the Oktavian benchmarks of different materials. The computational results include JEFF-v4.0T1, FENDL-3.2b, ENDF-VIII.O and JEFF-3.3.

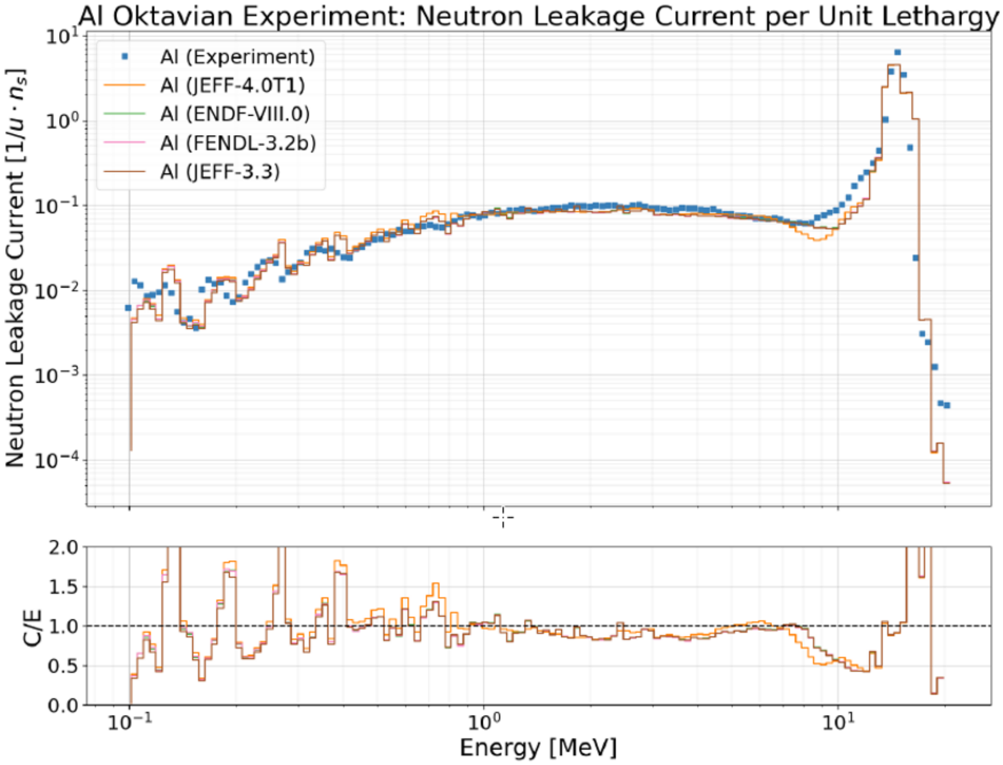


Figure C.12: JEFF v4.0T1 assessment, Oktavian experiment, neutron leakage current for Aluminum

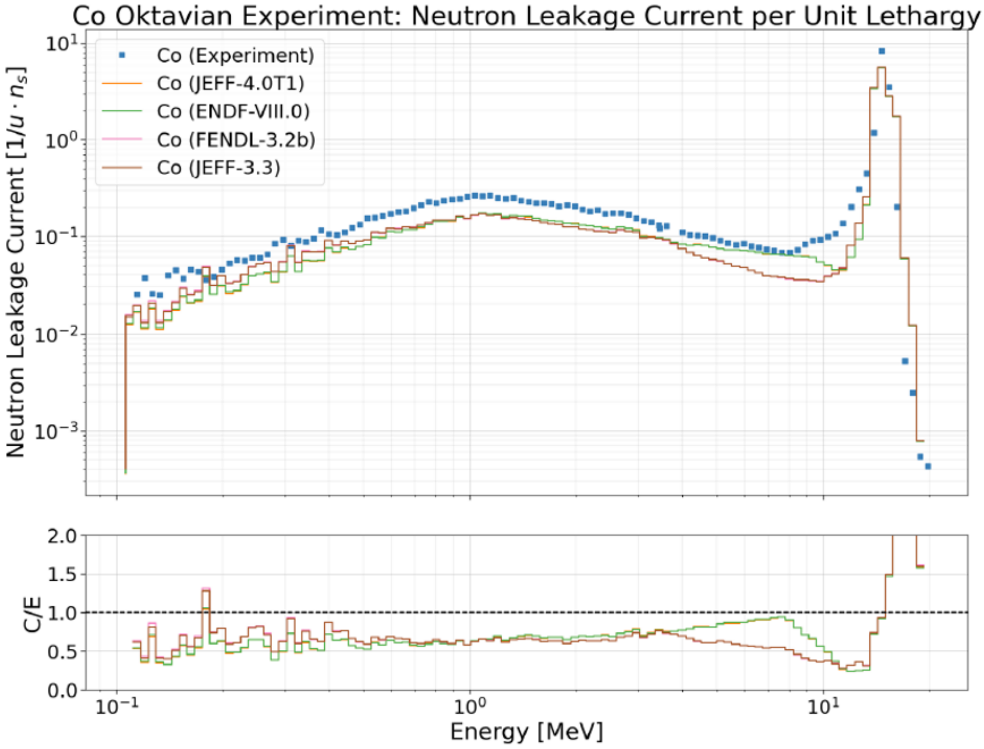


Figure C.13: JEFF v4.0T1 assessment, Oktavian experiment, neutron leakage current for Cobalt

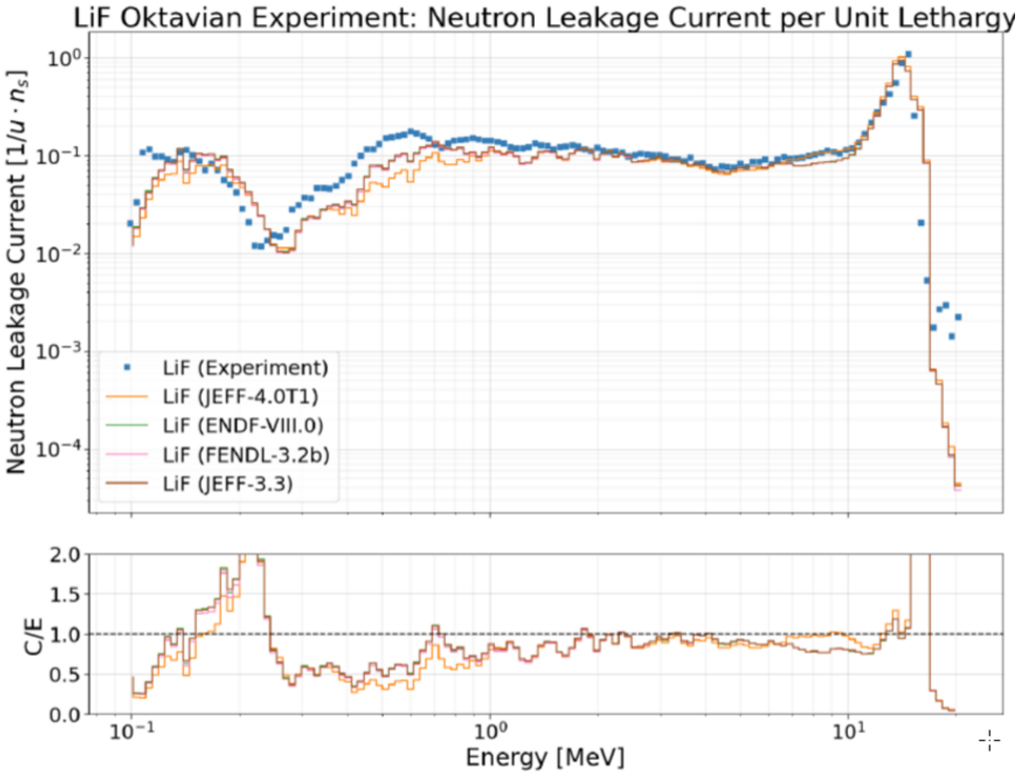


Figure C.14: JEFF v4.0T1 assessment, Oktavian experiment, neutron leakage current for Lithitum Fluoride

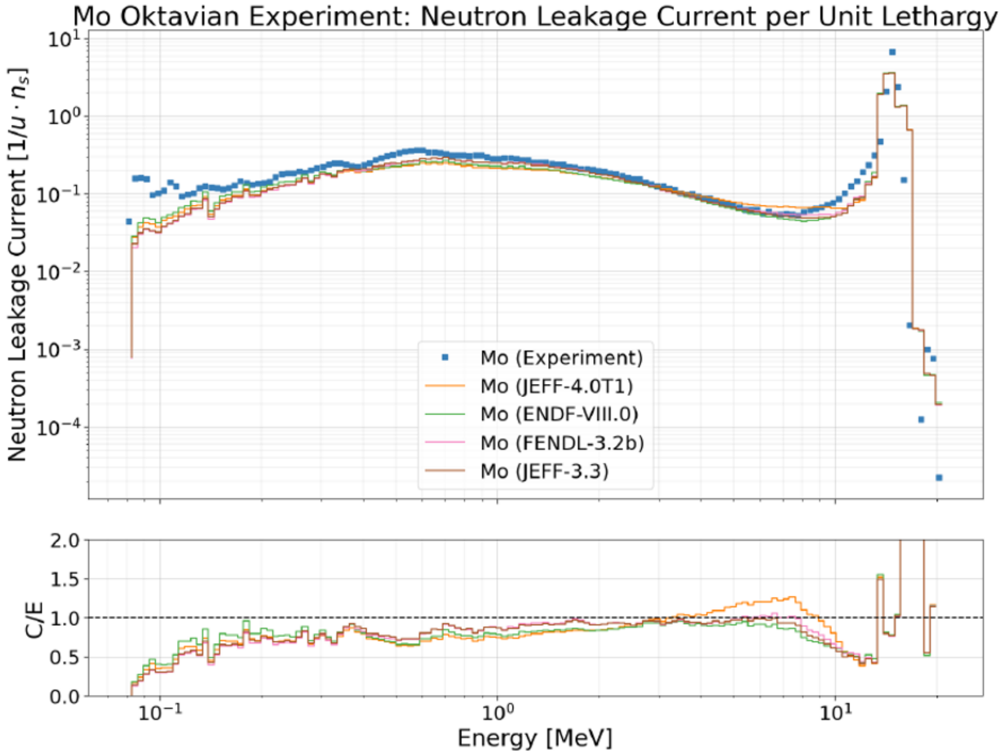


Figure C.15: JEFF v4.0T1 assessment, Oktavian experiment, neutron leakage current for Molybdenum

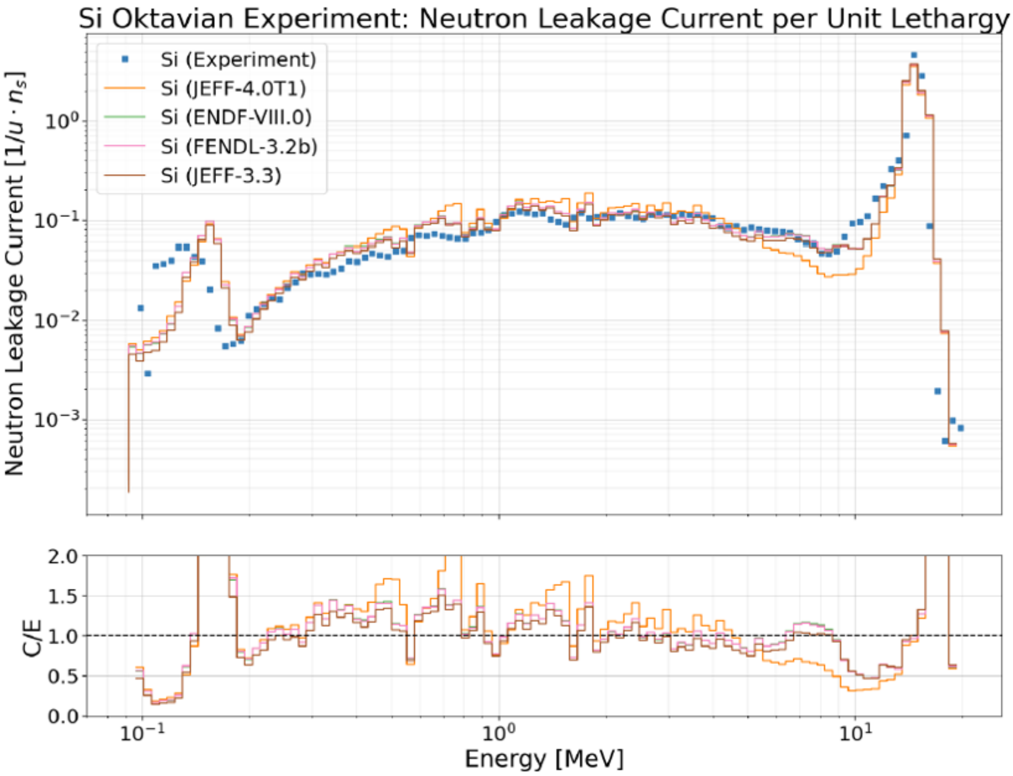


Figure C.16: JEFF v4.0T1 assessment, Oktavian experiment, neutron leakage current for Silicon

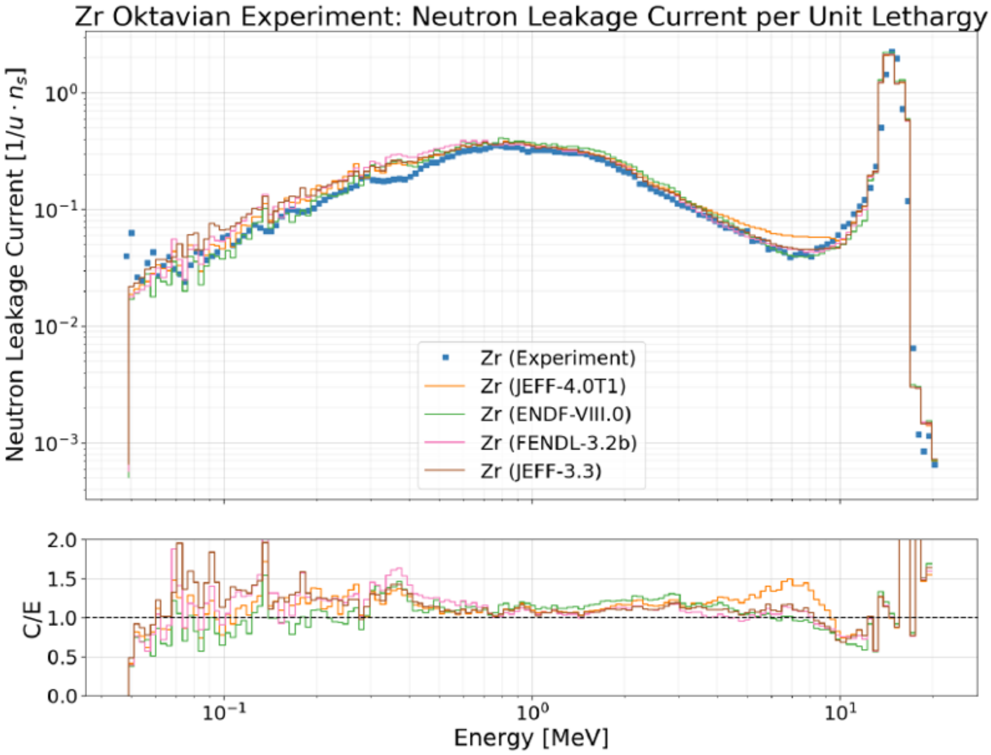
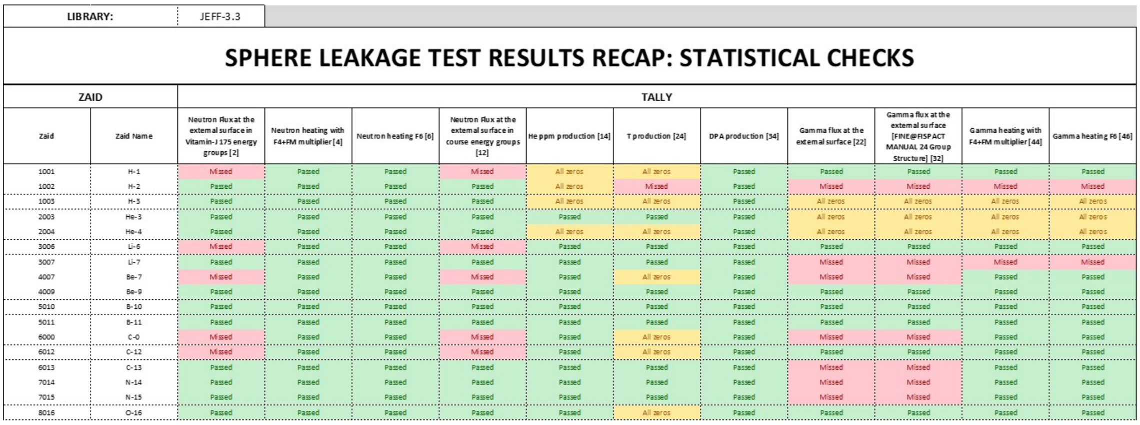


Figure C.17: JEFF v4.0T1 assessment, Oktavian experiment, neutron leakage current for Zirconium

Appendix D

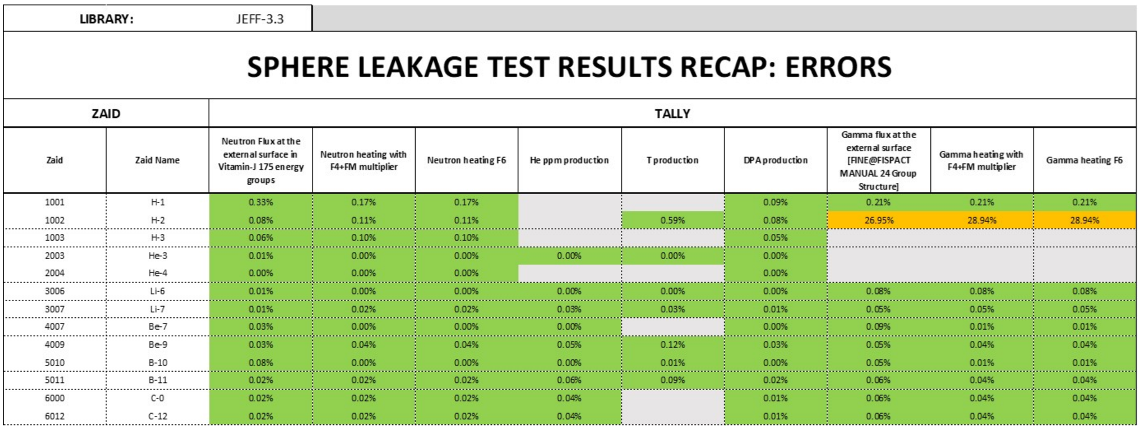
MS Excel output gallery

D.1 Examples from Sphere Leakage benchmark



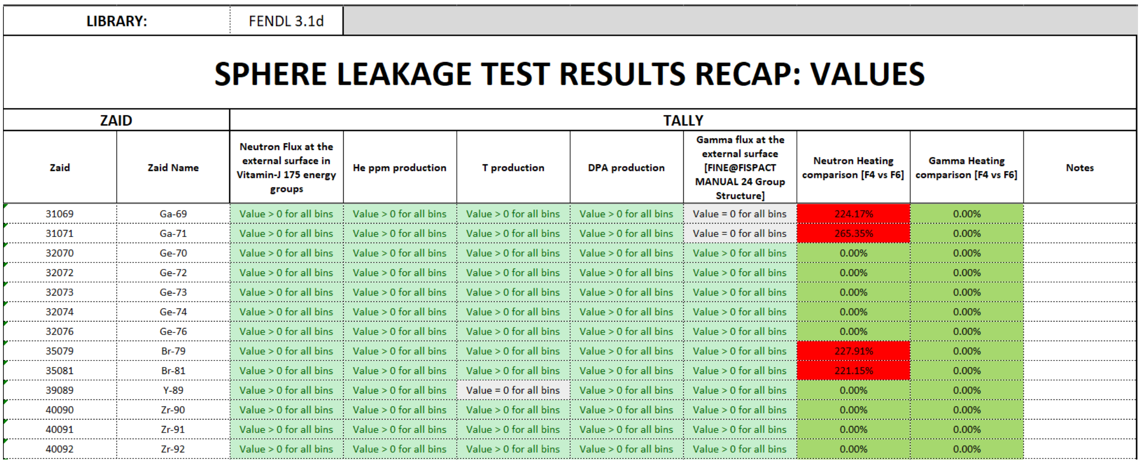
LIBRARY:	JEFF-3.3											
SPHERE LEAKAGE TEST RESULTS RECAP: STATISTICAL CHECKS												
ZAID		TALLY										
Zaid	Zaid Name	Neutron Flux at the external surface in Vitamin-J 175 energy groups [2]	Neutron heating with F4+FM multiplier [4]	Neutron heating F6 [6]	Neutron Flux at the external surface in course energy groups [12]	He ppm production [14]	T production [24]	DPA production [34]	Gamma flux at the external surface [22]	Gamma flux at the external surface [FINE@FISPACT MANUAL 24 Group Structure] [32]	Gamma heating with F4+FM multiplier [44]	Gamma heating F6 [46]
1001	H-1	Missed	Passed	Passed	Missed	All zeros	All zeros	Passed	Passed	Passed	Passed	Passed
1002	H-2	Passed	Passed	Passed	Passed	All zeros	Missed	Passed	Missed	Missed	Missed	Missed
1003	H-3	Passed	Passed	Passed	Passed	All zeros	All zeros	Passed	All zeros	All zeros	All zeros	All zeros
2003	He-3	Passed	Passed	Passed	Passed	Passed	Passed	Passed	All zeros	All zeros	All zeros	All zeros
2004	He-4	Passed	Passed	Passed	Passed	All zeros	All zeros	Passed	All zeros	All zeros	All zeros	All zeros
3006	Li-6	Missed	Passed	Passed	Missed	Passed	Passed	Passed	Passed	Passed	Passed	Passed
3007	Li-7	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Missed	Missed	Missed	Missed
4007	Be-7	Missed	Passed	Passed	Missed	Passed	All zeros	Passed	Missed	Missed	Passed	Passed
4009	Be-9	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed
5010	B-10	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed
5011	B-11	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed
6000	C-0	Missed	Passed	Passed	Missed	Passed	All zeros	Passed	Missed	Missed	Passed	Passed
6012	C-12	Missed	Passed	Passed	Missed	Passed	All zeros	Passed	Passed	Passed	Passed	Passed
6013	C-13	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Missed	Missed	Passed	Passed
7014	N-14	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Missed	Missed	Passed	Passed
7015	N-15	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Missed	Missed	Passed	Passed
8016	O-16	Passed	Passed	Passed	Passed	Passed	All zeros	Passed	Passed	Passed	Passed	Passed

Figure D.1: Example of extract of the statistical checks sheet from the excel output of the Sphere Leakage benchmark



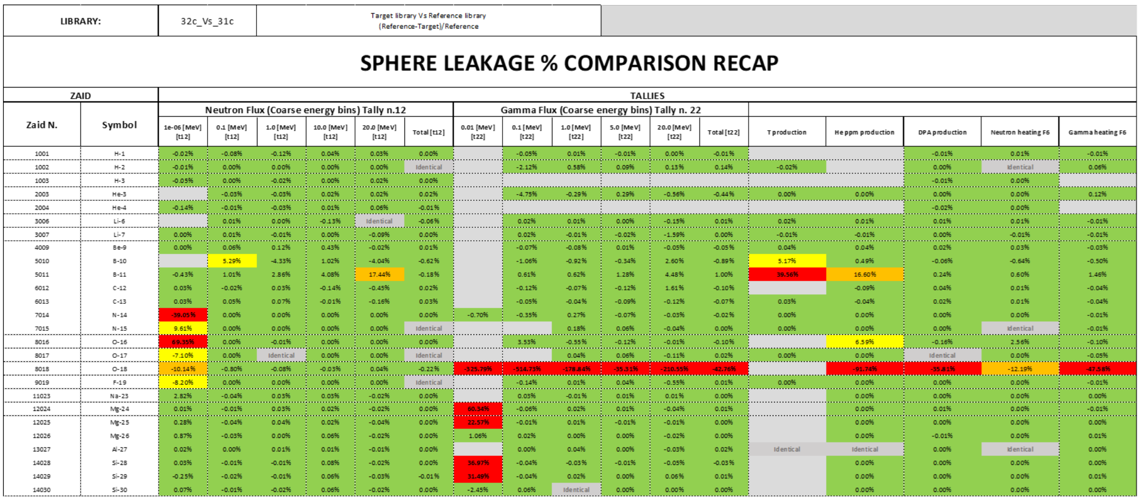
LIBRARY:	JEFF-3.3									
SPHERE LEAKAGE TEST RESULTS RECAP: ERRORS										
ZAID		TALLY								
Zaid	Zaid Name	Neutron Flux at the external surface in Vitamin-J 175 energy groups	Neutron heating with F4+FM multiplier	Neutron heating F6	He ppm production	T production	DPA production	Gamma flux at the external surface [FINE@FISPACT MANUAL 24 Group Structure]	Gamma heating with F4+FM multiplier	Gamma heating F6
1001	H-1	0.33%	0.17%	0.17%			0.09%	0.21%	0.21%	0.21%
1002	H-2	0.08%	0.11%	0.11%		0.59%	0.08%	26.95%	28.94%	28.94%
1003	H-3	0.06%	0.10%	0.10%			0.05%			
2003	He-3	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%			
2004	He-4	0.00%	0.00%	0.00%			0.00%			
3006	Li-6	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%	0.08%	0.08%	0.08%
3007	Li-7	0.01%	0.02%	0.02%	0.03%	0.03%	0.01%	0.05%	0.05%	0.05%
4007	Be-7	0.03%	0.00%	0.00%	0.00%		0.00%	0.09%	0.01%	0.01%
4009	Be-9	0.03%	0.04%	0.04%	0.05%	0.12%	0.03%	0.05%	0.04%	0.04%
5010	B-10	0.08%	0.00%	0.00%	0.00%	0.01%	0.00%	0.05%	0.01%	0.01%
5011	B-11	0.02%	0.02%	0.02%	0.06%	0.09%	0.02%	0.06%	0.04%	0.04%
6000	C-0	0.02%	0.02%	0.02%	0.04%		0.01%	0.06%	0.04%	0.04%
6012	C-12	0.02%	0.02%	0.02%	0.04%		0.01%	0.06%	0.04%	0.04%

Figure D.2: Example of extract of the statistical errors sheet from the excel output of the Sphere Leakage benchmark



LIBRARY:	FENDL 3.1d								
SPHERE LEAKAGE TEST RESULTS RECAP: VALUES									
ZAID		TALLY							
Zaid	Zaid Name	Neutron Flux at the external surface in Vitamin-J 175 energy groups	He ppm production	T production	DPA production	Gamma flux at the external surface [FINE@FISPACT MANUAL 24 Group Structure]	Neutron Heating comparison [F4 vs F6]	Gamma Heating comparison [F4 vs F6]	Notes
31069	Ga-69	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value = 0 for all bins	224.17%	0.00%	
31071	Ga-71	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value = 0 for all bins	265.35%	0.00%	
32070	Ge-70	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
32072	Ge-72	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
32073	Ge-73	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
32074	Ge-74	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
32076	Ge-76	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
35079	Br-79	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	227.91%	0.00%	
35081	Br-81	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	221.15%	0.00%	
39089	Y-89	Value > 0 for all bins	Value > 0 for all bins	Value = 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
40090	Zr-90	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
40091	Zr-91	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	
40092	Zr-92	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	Value > 0 for all bins	0.00%	0.00%	

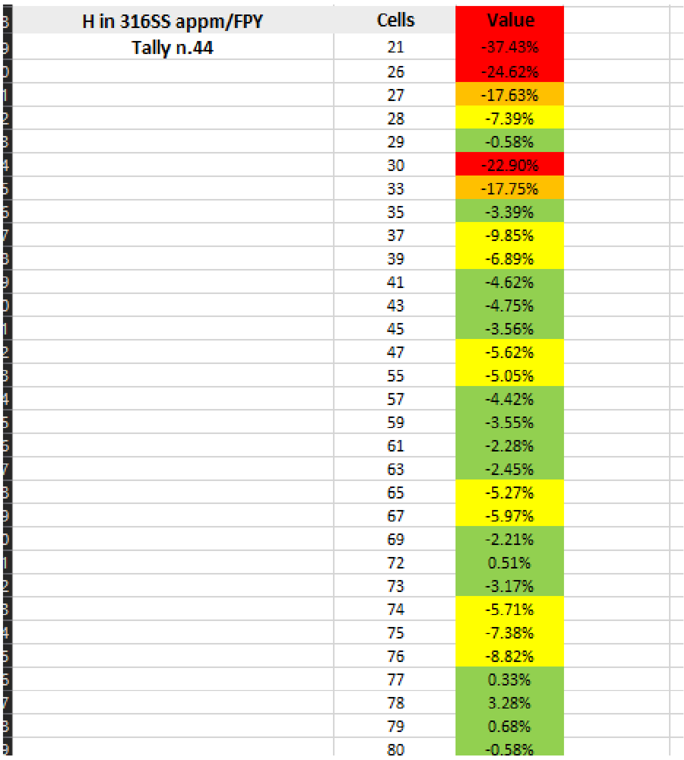
Figure D.3: Example of extract of the consistency checks sheet from the excel output of the Sphere Leakage benchmark



LIBRARY:	32c_Vs_31c	Target library Vs Reference library (Reference-Target)/Reference																	
SPHERE LEAKAGE % COMPARISON RECAP																			
ZAID		TALLIES																	
Zaid N.	Symbol	Neutron Flux (Coarse energy bins) Tally n.12						Gamma Flux (Coarse energy bins) Tally n. 22											
		1e-06 [MeV] [t12]	0.1 [MeV] [t12]	1.0 [MeV] [t12]	10.0 [MeV] [t12]	20.0 [MeV] [t12]	Total [t12]	0.01 [MeV] [t22]	0.1 [MeV] [t22]	1.0 [MeV] [t22]	5.0 [MeV] [t22]	20.0 [MeV] [t22]	Total [t22]	T production	He ppm production	DPA production	Neutron heating F6	Gamma heating F6
1001	H-1	-0.02%	-0.08%	-0.12%	0.04%	0.03%	0.00%		-0.05%	0.01%	-0.01%	0.00%	-0.01%			-0.01%	0.01%	-0.01%
1002	H-2	-0.01%	0.00%	0.00%	0.00%	0.00%	Identical		-2.12%	0.58%	0.09%	0.13%	0.14%	-0.02%		0.00%	Identical	0.06%
1003	H-3	-0.05%	0.00%	-0.02%	0.00%	0.02%	0.00%									-0.01%	0.00%	
2003	He-3		-0.03%	-0.03%	0.02%	0.02%	0.02%		-4.75%	-0.29%	0.29%	-0.56%	-0.44%	0.00%	0.00%	0.00%	0.00%	0.12%
2004	He-4	-0.14%	-0.01%	-0.03%	0.01%	0.06%	-0.01%									-0.02%	0.00%	
3006	Li-6		0.01%	0.00%	-0.13%	Identical	-0.06%		0.02%	0.01%	0.00%	-0.15%	0.01%	0.02%	0.01%	0.01%	0.01%	-0.01%
3007	Li-7	0.00%	0.01%	-0.01%	0.00%	-0.09%	0.00%		0.02%	-0.01%	-0.02%	-1.59%	0.00%	-0.01%	-0.01%	0.00%	-0.01%	-0.01%
4009	Be-9	0.00%	0.06%	0.12%	0.43%	-0.02%	0.01%		-0.07%	-0.08%	0.01%	-0.05%	-0.05%	0.04%	0.04%	0.02%	0.03%	-0.03%
5010	B-10		5.29%	-4.33%	1.02%	-4.04%	-0.62%		-1.06%	-0.92%	-0.34%	2.60%	-0.89%	5.17%	0.49%	-0.06%	-0.64%	-0.50%
5011	B-11	-0.43%	1.01%	2.86%	4.08%	17.44%	-0.18%		0.61%	0.62%	1.28%	4.48%	1.00%	39.56%	16.60%	0.24%	0.60%	1.46%
6012	C-12	0.03%	-0.02%	0.03%	-0.14%	-0.45%	0.02%		-0.12%	-0.07%	-0.12%	1.61%	-0.10%		-0.09%	0.04%	0.01%	-0.04%
6013	C-13	0.03%	0.05%	0.07%	-0.01%	-0.16%	0.03%		-0.05%	-0.04%	-0.09%	-0.12%	-0.07%	0.03%	-0.04%	0.02%	0.01%	-0.04%
7014	N-14	-39.05%	0.00%	0.00%	0.00%	0.00%	0.00%	-0.70%	-0.35%	0.27%	-0.07%	-0.03%	-0.02%	0.00%	0.00%	0.00%	0.00%	-0.01%
7015	N-15	9.61%	0.00%	0.00%	0.00%	0.00%	Identical			0.18%	0.06%	-0.04%	0.00%	0.00%	0.00%	0.00%	Identical	-0.01%
8016	O-16	69.35%	0.00%	-0.01%	0.00%	0.00%	0.00%		3.53%	-0.55%	-0.12%	-0.01%	-0.10%		6.59%	-0.16%	2.56%	-0.10%
8017	O-17	-7.10%	0.00%	Identical	0.00%	0.00%	Identical			0.04%	0.06%	-0.11%	0.02%	0.00%	0.00%	Identical	0.00%	-0.05%
8018	O-18	-10.14%	-0.80%	-0.08%	-0.03%	0.04%	-0.22%	-325.79%	-514.73%	-178.84%	-35.31%	-210.55%	-42.76%		-91.74%	-35.81%	-12.19%	-47.58%
9019	F-19	-8.20%	0.00%	0.00%	0.00%	0.00%	Identical		-0.14%	0.01%	0.04%	-0.55%	0.01%	0.00%	0.00%	0.00%	0.00%	-0.01%
11023	Na-23	2.82%	-0.04%	0.03%	0.03%	-0.02%	0.00%		0.03%	-0.01%	0.01%	0.01%	0.00%		0.00%	0.00%	0.00%	0.00%
12024	Mg-24	0.01%	-0.01%	0.03%	0.02%	-0.02%	0.00%	60.34%	-0.06%	0.02%	0.01%	-0.04%	0.01%		0.00%	0.01%	0.00%	-0.01%
12025	Mg-25	0.28%	-0.04%	0.04%	0.02%	-0.04%	0.00%	22.57%	-0.01%	0.01%	-0.01%	-0.01%	0.00%		0.00%	0.00%	0.00%	0.00%
12026	Mg-26	0.87%	-0.03%	0.00%	0.06%	-0.02%	0.00%	1.06%	0.02%	0.00%	0.00%	-0.02%	0.00%		0.00%	-0.01%	0.00%	0.01%
13027	Al-27	0.02%	0.00%	0.01%	0.01%	-0.01%	0.00%		0.00%	0.04%	0.00%	-0.03%	0.02%	Identical	Identical	0.00%	Identical	0.00%
14028	Si-28	0.03%	-0.01%	-0.01%	0.08%	-0.02%	0.00%	36.97%	-0.04%	-0.03%	-0.01%	-0.05%	-0.03%		0.00%	0.00%	0.00%	0.00%
14029	Si-29	-0.25%	-0.02%	-0.01%	0.06%	-0.03%	-0.01%	31.49%	-0.04%	0.02%	0.00%	0.06%	0.01%		0.00%	0.00%	0.00%	0.01%
14030	Si-30	0.07%	-0.01%	-0.02%	0.06%	-0.02%	0.00%	-2.45%	0.06%	Identical	0.00%	0.00%	0.00%		0.00%	0.00%	0.00%	0.00%

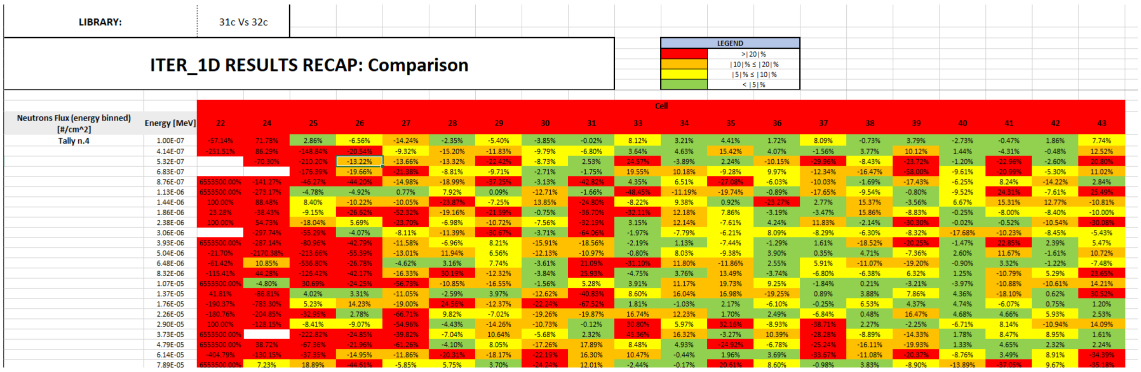
Figure D.4: Example of extract of the comparison sheet from the excel output of the Sphere Leakage benchmark

D.2 Examples from general computational benchmark



H in 316SS appm/FPY	Cells	Value
Tally n.44	21	-37.43%
	26	-24.62%
	27	-17.63%
	28	-7.39%
	29	-0.58%
	30	-22.90%
	33	-17.75%
	35	-3.39%
	37	-9.85%
	39	-6.89%
	41	-4.62%
	43	-4.75%
	45	-3.56%
	47	-5.62%
	55	-5.05%
	57	-4.42%
	59	-3.55%
	61	-2.28%
	63	-2.45%
	65	-5.27%
	67	-5.97%
	69	-2.21%
	72	0.51%
	73	-3.17%
	74	-5.71%
	75	-7.38%
	76	-8.82%
	77	0.33%
	78	3.28%
	79	0.68%
	80	-0.58%

Figure D.5: Example of extract of the comparison sheet from the excel output of a general computational benchmark. Tally single binning.



LIBRARY: 31c Vs 32c

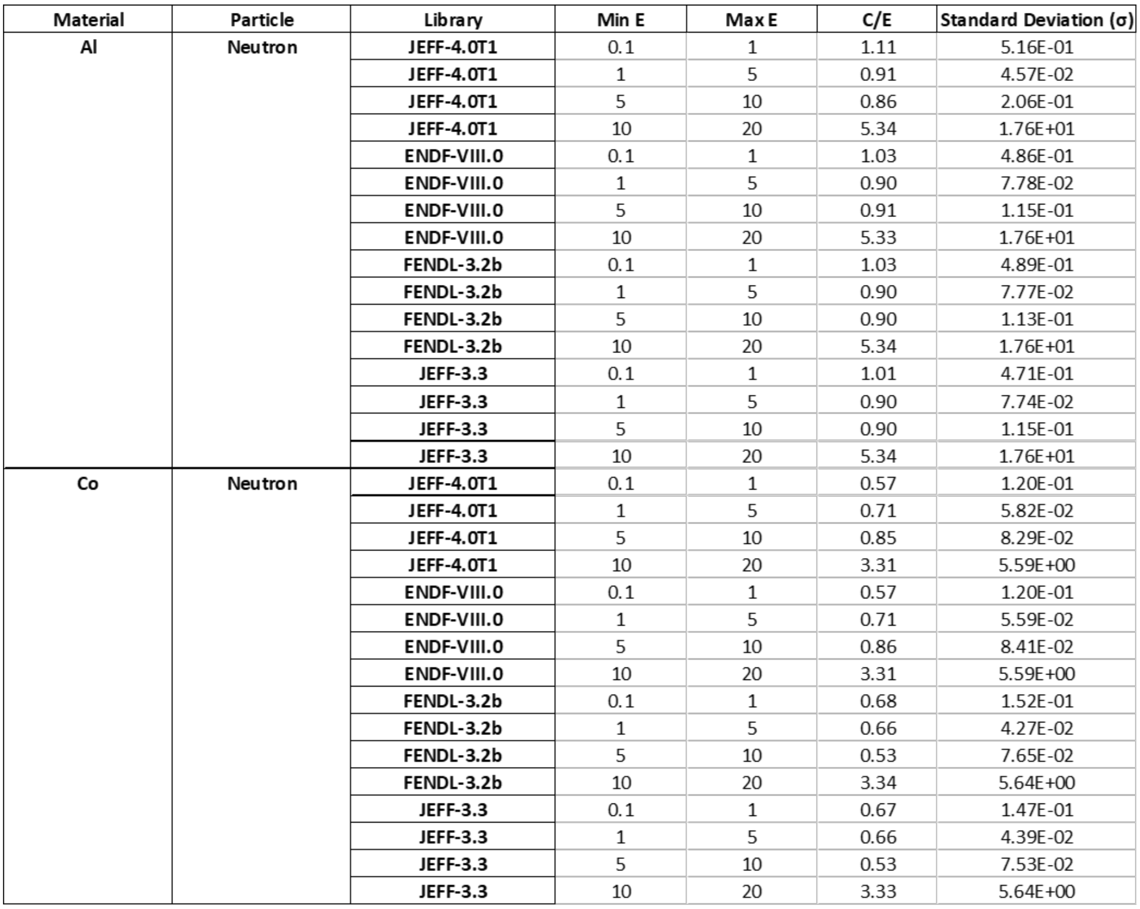
ITER_1D RESULTS RECAP: Comparison

LEGEND	
🟥	>|20|%
🟧	|10|% ≤ |20|%
🟨	|5|% ≤ |10|%
🟩	< |5|%

		Cell																			
Neutrons Flux (energy binned) [#/cm^2]	Energy [MeV]	22	24	25	26	27	28	29	30	31	33	34	35	36	37	38	39	40	41	42	43
Tally n.4	1.00E-07	-57.14%	71.78%	2.86%	-6.56%	-14.24%	-2.35%	-5.40%	-3.85%	-0.02%	8.12%	3.21%	4.41%	1.72%	8.09%	-0.73%	3.79%	-2.73%	-0.47%	1.86%	7.74%
	4.14E-07	-251.51%	86.29%	-148.84%	-20.54%	-9.32%	-15.20%	-11.83%	-9.79%	-6.80%	3.64%	4.63%	15.42%	4.07%	-1.56%	3.77%	10.12%	1.44%	-4.31%	-0.48%	12.52%
	5.32E-07		-70.30%	-210.20%	-13.22%	-13.66%	-13.32%	-22.42%	-8.73%	2.53%	24.57%	-3.89%	2.24%	-10.15%	-29.96%	-8.43%	-23.72%	-1.20%	-22.96%	-2.60%	20.80%
	6.83E-07			-176.39%	-19.66%	-21.38%	-8.81%	-9.71%	-2.71%	-1.75%	19.55%	10.18%	-9.28%	9.97%	-12.34%	-16.47%	-58.00%	-9.61%	-20.99%	-5.30%	11.02%
	8.76E-07	6553500.00%	-141.27%	-46.27%	-44.20%	-14.98%	-18.99%	-37.25%	-3.13%	-42.82%	4.35%	6.51%	-27.08%	-6.03%	-10.03%	-1.69%	-17.43%	-6.25%	8.24%	-14.22%	2.84%
	1.13E-06	6553500.00%	-273.17%	-4.78%	-4.92%	0.77%	7.92%	0.09%	-12.71%	-1.66%	-48.45%	-11.19%	-19.74%	-0.89%	-17.65%	-9.54%	-0.80%	-9.52%	24.31%	-7.61%	25.49%
	1.44E-06	100.00%	88.48%	8.40%	-10.22%	-10.05%	-23.87%	-7.25%	13.85%	-24.80%	-8.22%	9.38%	0.92%	-23.27%	2.77%	15.37%	-3.56%	6.67%	15.31%	12.77%	-10.81%
	1.86E-06	23.28%	-38.43%	-9.15%	-26.62%	-52.32%	-19.16%	-21.59%	-0.75%	-36.70%	-32.11%	12.18%	7.86%	-3.19%	-3.47%	15.86%	-8.83%	-0.25%	-8.00%	-8.40%	-10.00%
	2.38E-06	100.00%	54.73%	-18.04%	5.69%	-23.70%	-6.98%	-10.72%	-7.56%	-32.19%	3.15%	12.14%	-7.61%	4.24%	11.83%	-2.14%	-30.30%	-0.02%	-0.52%	-10.54%	-30.08%
	3.06E-06		-297.74%	-55.29%	-4.07%	-8.11%	-11.39%	-30.67%	-3.71%	-64.06%	-1.97%	-7.79%	-6.21%	8.09%	-8.29%	-6.30%	-8.32%	-17.68%	-10.23%	-8.45%	-5.43%
	3.93E-06	6553500.00%	-287.14%	-80.96%	-42.79%	-11.58%	-6.96%	8.21%	-15.91%	-18.56%	-2.19%	1.13%	-7.44%	-1.29%	1.61%	-18.52%	-20.25%	-1.47%	22.85%	2.39%	5.47%
	5.04E-06	-21.70%	-2170.38%	-213.66%	-55.39%	-13.01%	11.94%	6.56%	-12.13%	-10.97%	-0.80%	8.03%	-9.38%	3.90%	0.35%	4.71%	-7.36%	2.60%	11.67%	-1.61%	10.72%
	6.48E-06	-61.42%	10.85%	-536.80%	-26.78%	-4.62%	3.16%	7.74%	-3.61%	21.09%	-31.10%	11.80%	-11.86%	2.55%	5.91%	-11.07%	-19.20%	-0.90%	3.32%	-1.22%	-7.48%
	8.32E-06	-115.41%	44.28%	-126.42%	-42.17%	-16.33%	30.19%	-12.32%	-3.84%	25.93%	-4.75%	3.76%	13.49%	-3.74%	-6.80%	-6.38%	6.32%	1.25%	-10.79%	5.29%	23.65%
	1.07E-05	6553500.00%	-4.80%	30.69%	-24.25%	-56.73%	-10.85%	-16.55%	-1.56%	5.28%	3.91%	11.17%	19.73%	9.25%	-1.84%	0.21%	-3.21%	-3.97%	-10.88%	-10.61%	14.21%
	1.37E-05	41.81%	-86.81%	4.02%	3.31%	-11.05%	-2.59%	3.97%	-12.62%	-40.83%	8.60%	16.04%	16.98%	-19.25%	0.89%	3.88%	7.86%	4.36%	-18.10%	0.62%	30.52%
	1.76E-05	-190.37%	-783.30%	5.23%	14.23%	-19.00%	24.56%	-12.37%	-22.24%	-67.52%	1.81%	-1.03%	2.17%	-6.10%	-0.25%	6.53%	4.37%	4.74%	-6.07%	0.75%	1.20%
	2.26E-05	-180.76%	-204.85%	-32.95%	2.78%	-66.71%	9.82%	-7.02%	-19.26%	-19.87%	16.74%	12.23%	1.70%	2.49%	-6.84%	0.48%	16.47%	4.68%	4.66%	5.93%	2.53%
	2.90E-05	100.00%	-128.15%	-8.41%	-9.07%	-54.96%	-4.43%	-14.26%	-10.73%	-0.12%	30.80%	5.97%	32.16%	-8.93%	-38.71%	2.27%	-2.25%	-6.71%	8.14%	-10.94%	14.09%
	3.73E-05	6553500.00%		-222.82%	-24.85%	-39.82%	-7.04%	10.64%	-5.68%	2.32%	45.36%	16.32%	-3.27%	10.39%	-28.28%	-8.89%	-14.33%	1.78%	8.47%	8.95%	1.61%
	4.79E-05	6553500.00%	38.72%	-67.36%	-21.96%	-61.26%	-4.10%	8.05%	-17.26%	17.89%	8.48%	4.93%	-24.92%	-6.78%	-25.24%	-16.11%	-19.93%	1.33%	4.65%	2.32%	2.24%
	6.14E-05	-404.79%	-130.15%	-37.35%	-14.95%	-11.86%	-20.31%	-18.17%	-22.19%	16.30%	10.47%	-0.44%	1.96%	3.69%	-33.67%	-11.08%	-20.37%	-8.76%	3.49%	8.91%	-34.39%
	7.89E-05	6553500.00%	7.23%	18.89%	-44.61%	-5.85%	5.75%	3.70%	-24.24%	12.01%	-2.44%	-0.17%	20.61%	8.60%	-0.98%	3.83%	-8.90%	-13.89%	-37.05%	9.67%	-35.18%

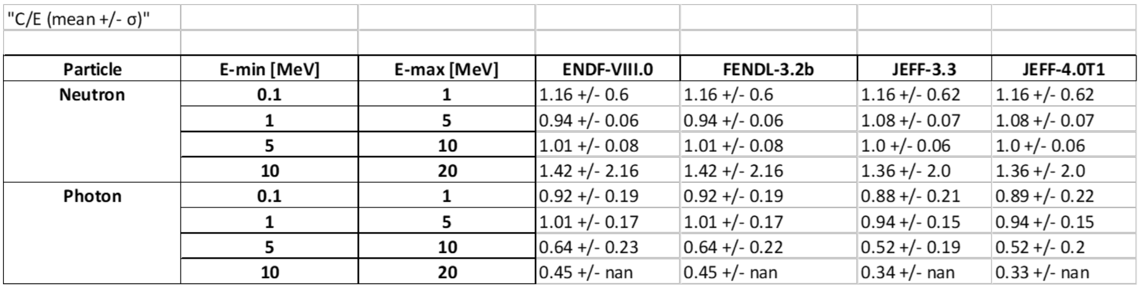
Figure D.6: Example of extract of the comparison sheet from the excel output of a general computational benchmark. Tally double binning.

D.3 Examples from experimental benchmark custom post-processing



Material	Particle	Library	Min E	Max E	C/E	Standard Deviation (σ)
Al	Neutron	JEFF-4.0T1	0.1	1	1.11	5.16E-01
		JEFF-4.0T1	1	5	0.91	4.57E-02
		JEFF-4.0T1	5	10	0.86	2.06E-01
		JEFF-4.0T1	10	20	5.34	1.76E+01
		ENDF-VIII.0	0.1	1	1.03	4.86E-01
		ENDF-VIII.0	1	5	0.90	7.78E-02
		ENDF-VIII.0	5	10	0.91	1.15E-01
		ENDF-VIII.0	10	20	5.33	1.76E+01
		FENDL-3.2b	0.1	1	1.03	4.89E-01
		FENDL-3.2b	1	5	0.90	7.77E-02
		FENDL-3.2b	5	10	0.90	1.13E-01
		FENDL-3.2b	10	20	5.34	1.76E+01
		JEFF-3.3	0.1	1	1.01	4.71E-01
		JEFF-3.3	1	5	0.90	7.74E-02
		JEFF-3.3	5	10	0.90	1.15E-01
		JEFF-3.3	10	20	5.34	1.76E+01
Co	Neutron	JEFF-4.0T1	0.1	1	0.57	1.20E-01
		JEFF-4.0T1	1	5	0.71	5.82E-02
		JEFF-4.0T1	5	10	0.85	8.29E-02
		JEFF-4.0T1	10	20	3.31	5.59E+00
		ENDF-VIII.0	0.1	1	0.57	1.20E-01
		ENDF-VIII.0	1	5	0.71	5.59E-02
		ENDF-VIII.0	5	10	0.86	8.41E-02
		ENDF-VIII.0	10	20	3.31	5.59E+00
		FENDL-3.2b	0.1	1	0.68	1.52E-01
		FENDL-3.2b	1	5	0.66	4.27E-02
		FENDL-3.2b	5	10	0.53	7.65E-02
		FENDL-3.2b	10	20	3.34	5.64E+00
		JEFF-3.3	0.1	1	0.67	1.47E-01
		JEFF-3.3	1	5	0.66	4.39E-02
		JEFF-3.3	5	10	0.53	7.53E-02
		JEFF-3.3	10	20	3.33	5.64E+00

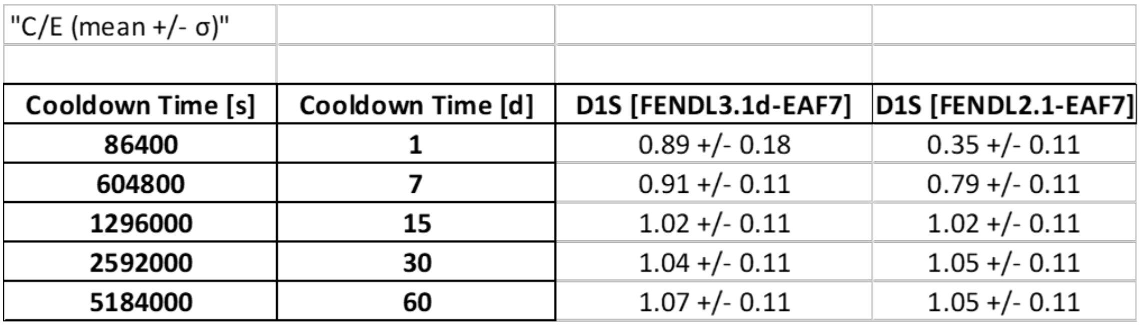
Figure D.7: Example of extract of the global sheet from the excel output of the Oktavian benchmark



"C/E (mean +/- σ)"

Particle	E-min [MeV]	E-max [MeV]	ENDF-VIII.0	FENDL-3.2b	JEFF-3.3	JEFF-4.0T1
Neutron	0.1	1	1.16 +/- 0.6	1.16 +/- 0.6	1.16 +/- 0.62	1.16 +/- 0.62
	1	5	0.94 +/- 0.06	0.94 +/- 0.06	1.08 +/- 0.07	1.08 +/- 0.07
	5	10	1.01 +/- 0.08	1.01 +/- 0.08	1.0 +/- 0.06	1.0 +/- 0.06
	10	20	1.42 +/- 2.16	1.42 +/- 2.16	1.36 +/- 2.0	1.36 +/- 2.0
Photon	0.1	1	0.92 +/- 0.19	0.92 +/- 0.19	0.88 +/- 0.21	0.89 +/- 0.22
	1	5	1.01 +/- 0.17	1.01 +/- 0.17	0.94 +/- 0.15	0.94 +/- 0.15
	5	10	0.64 +/- 0.23	0.64 +/- 0.22	0.52 +/- 0.19	0.52 +/- 0.2
	10	20	0.45 +/- nan	0.45 +/- nan	0.34 +/- nan	0.33 +/- nan

Figure D.8: Example of extract of the single material sheet from the excel output of the Oktavian benchmark



"C/E (mean +/- σ)"

Cooldown Time [s]	Cooldown Time [d]	D1S [FENDL3.1d-EAF7]	D1S [FENDL2.1-EAF7]
86400	1	0.89 +/- 0.18	0.35 +/- 0.11
604800	7	0.91 +/- 0.11	0.79 +/- 0.11
1296000	15	1.02 +/- 0.11	1.02 +/- 0.11
2592000	30	1.04 +/- 0.11	1.05 +/- 0.11
5184000	60	1.07 +/- 0.11	1.05 +/- 0.11

Figure D.9: Example of extract from the excel output of the FNG benchmark

Appendix E

Plot Atlas output gallery

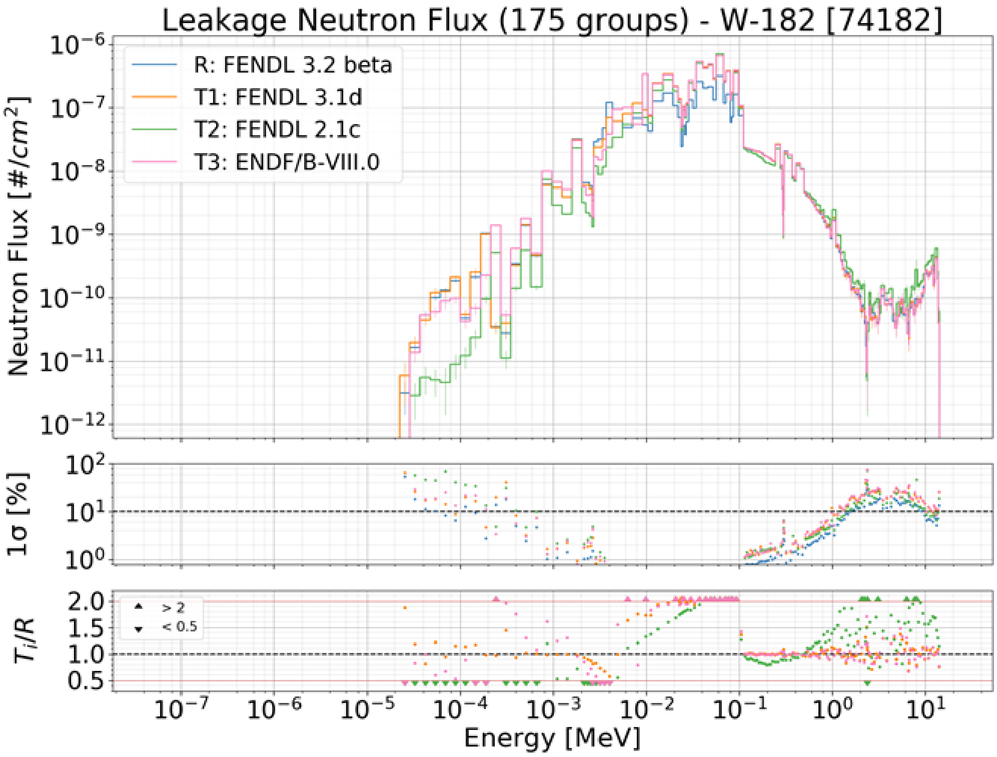


Figure E.1: Example of binned tally plot

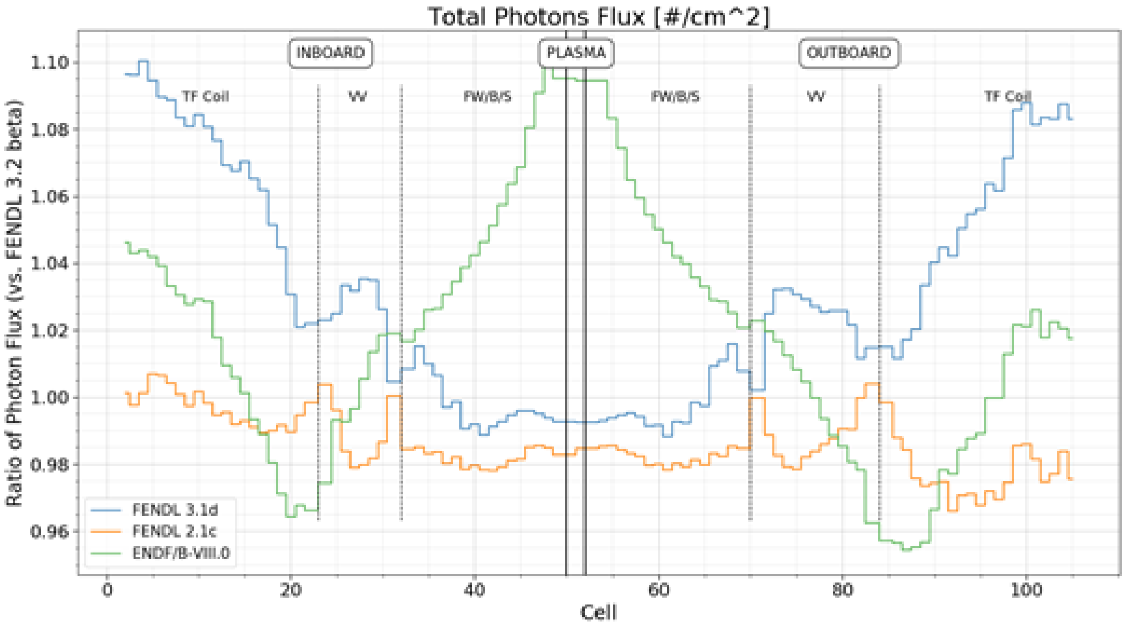


Figure E.2: Example of ratio plot (ITER 1D custom)

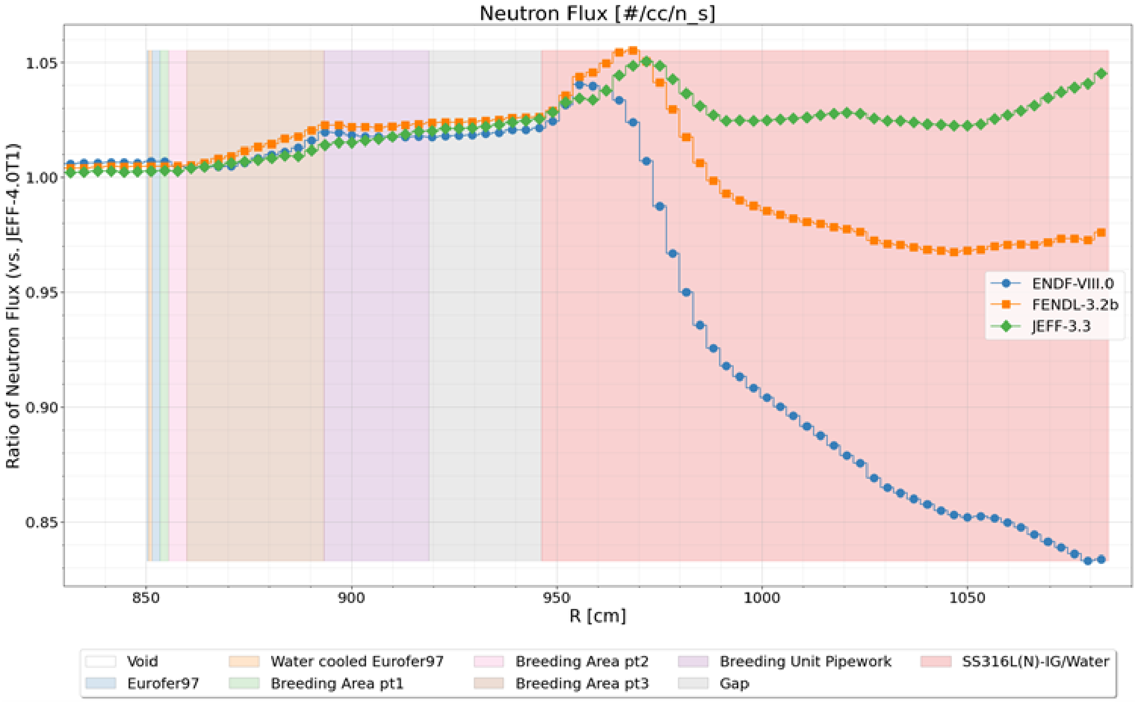


Figure E.3: Example of ratio plot (TBM custom)

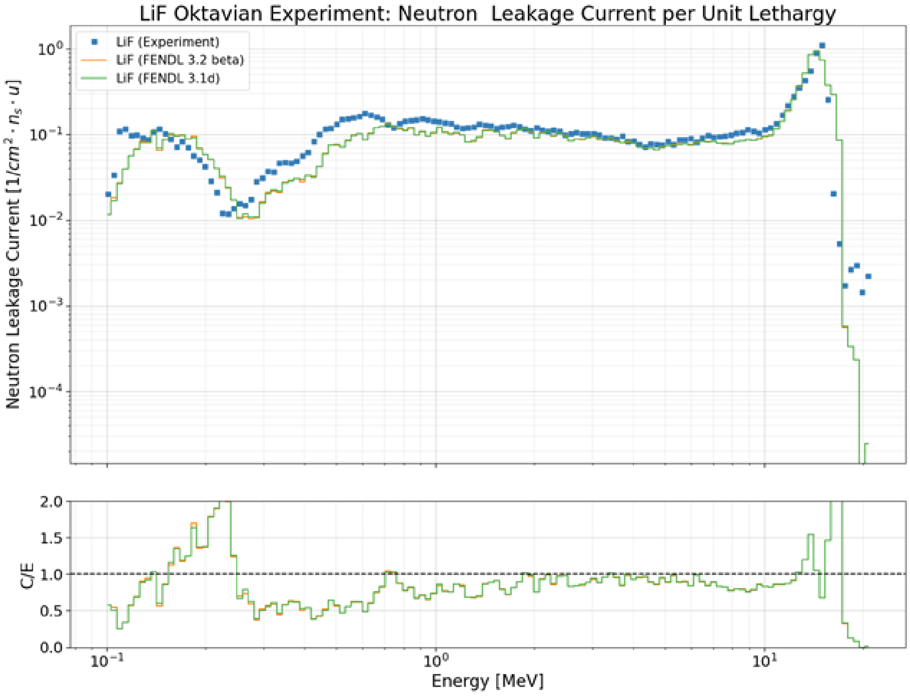


Figure E.4: Example of experimental continuous plot

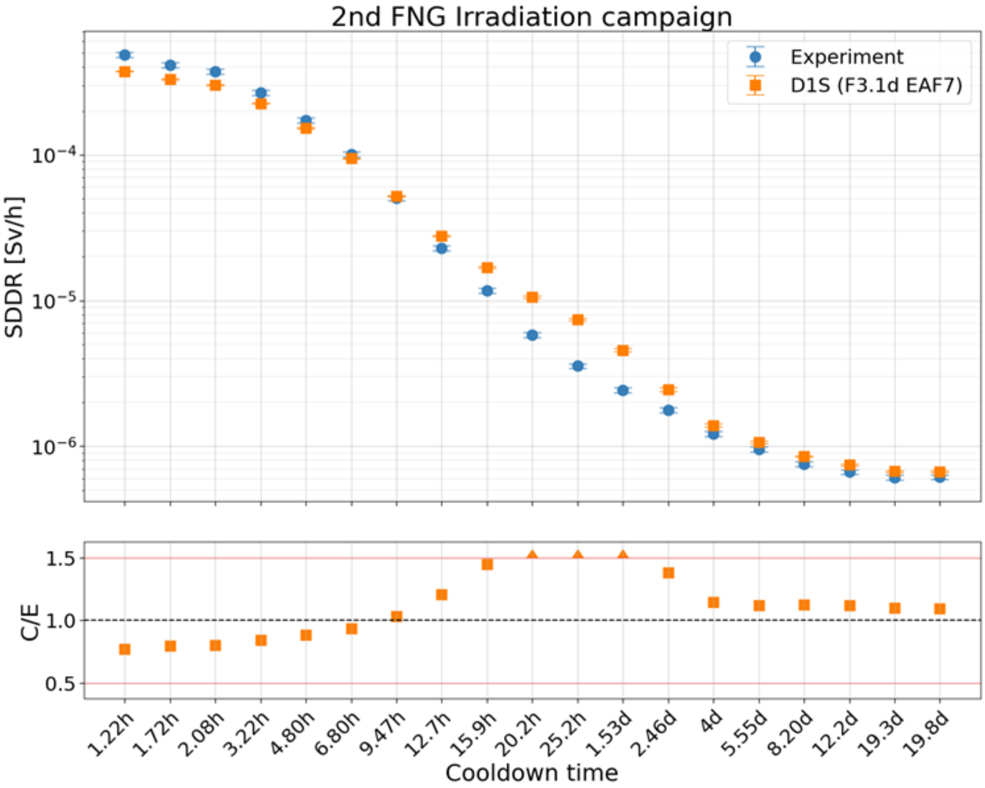


Figure E.5: Example of experimental discreet plot

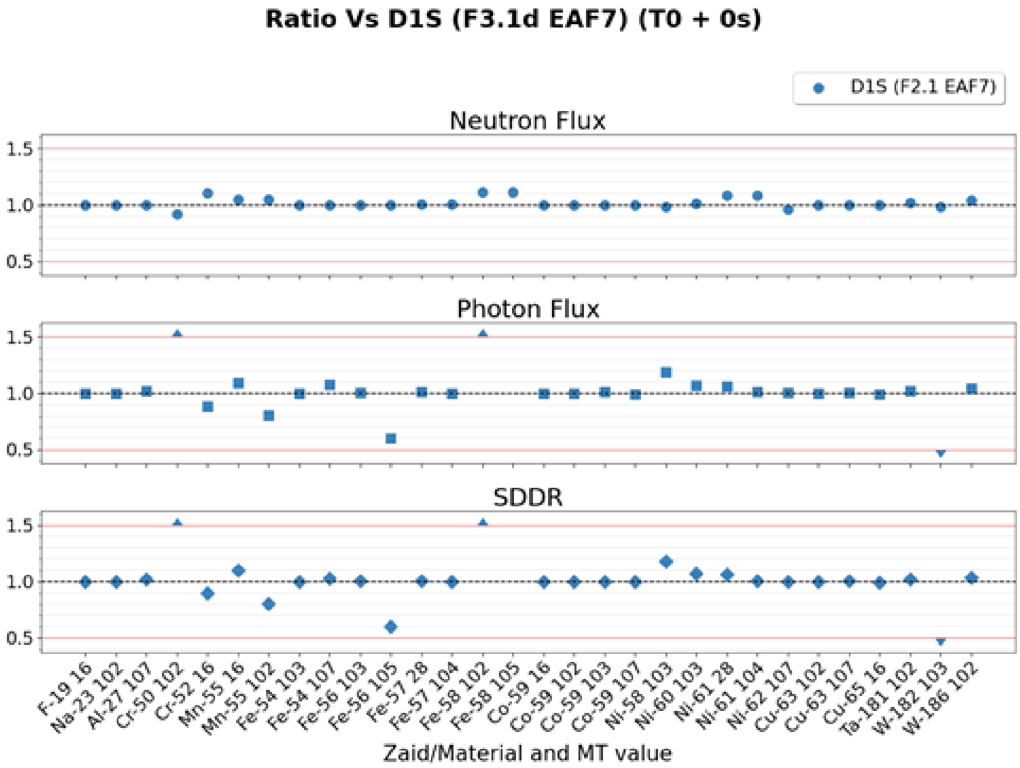


Figure E.6: Example of waves plot

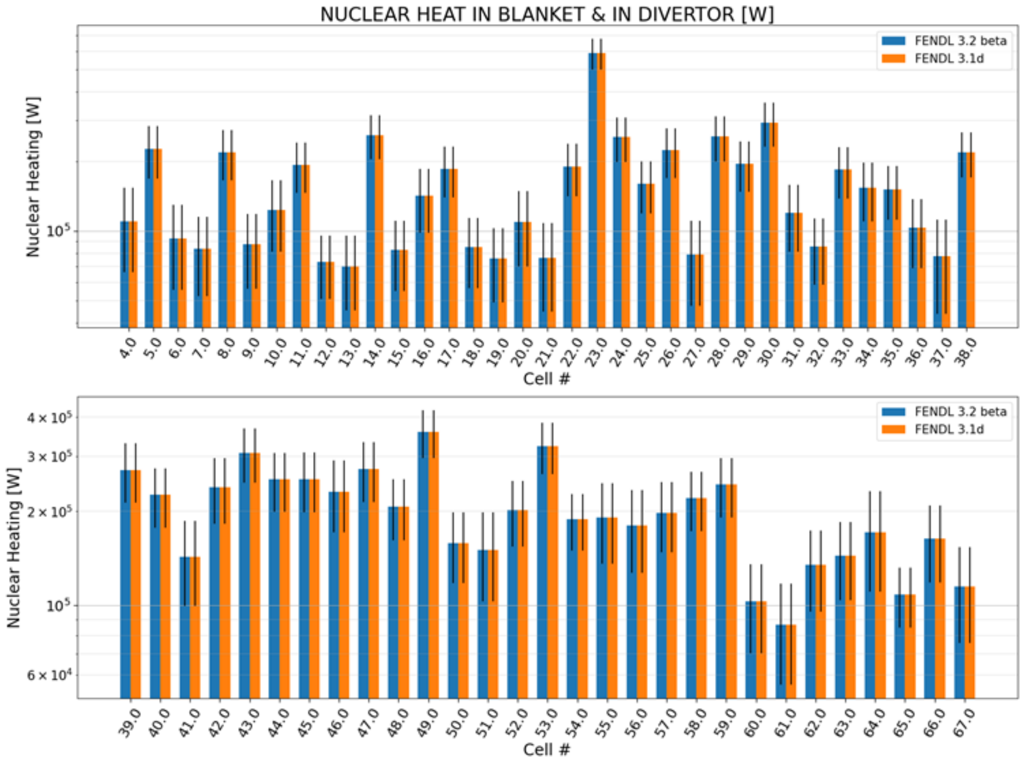


Figure E.7: Example of histogram plot