



**HAL**  
open science

# PyNjoy2016: an open source system for producing cross sections libraries for DRAGON5 and SERPENT2

Vivian Salino, Alain Hébert

## ► To cite this version:

Vivian Salino, Alain Hébert. PyNjoy2016: an open source system for producing cross sections libraries for DRAGON5 and SERPENT2. M&C 2023 - The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, American Nuclear Society, Aug 2023, Niagara Falls, Ontario, Canada. irsn-04095482v1

**HAL Id: irsn-04095482**

**<https://irsn.hal.science/irsn-04095482v1>**

Submitted on 11 May 2023 (v1), last revised 11 Jul 2023 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## PyNjoy2016: an open source system for producing cross sections libraries for DRAGON5 and SERPENT2

V. Salino<sup>1,2</sup> and A. Hébert<sup>1</sup>

<sup>1</sup>École Polytechnique de Montréal, Institut de Génie Nucléaire  
P.O. Box 6079, Station Centre-Ville, Montréal, Qc., H3C 3A7, Canada

<sup>2</sup>Institut de Radioprotection et Sûreté Nucléaire (IRSN), PSN-RES/SNC/LN  
BP 17, Fontenay-aux-Roses, 92262, France

[vivian.salino@irsn.fr](mailto:vivian.salino@irsn.fr), [alain.hebert@polymtl.ca](mailto:alain.hebert@polymtl.ca)

### ABSTRACT

PyNjoy2016 was born from the fusion of PyNjoy2012 and NJOY2016. This new software is dedicated to the production of libraries for DRAGON5 and SERPENT2, in DRAGLIB and ACE formats, respectively, for deterministic and Monte-Carlo calculations, from a nuclear data evaluation in ENDF-6 format. Among the various improvements over its predecessor, PyNjoy2016 supports the advanced energy deposition model of SERPENT2. It also includes CPU-parallelized features to facilitate the propagation of uncertainties by Total Monte-Carlo method. For that, it relies either on SANDY code, called on-the-fly, or on presampled random TENDL files. Moreover, it allows *consistent* production of ACE and DRAGLIB files, deriving from the same NJOY computations. The exact deterministic bias of a specific DRAGON5 calculation scheme can therefore be calculated. In contrast, the usual comparisons between deterministic and Monte-Carlo methods are often *not* based on consistently produced libraries, leading to an error in the calculated deterministic bias. Finally, its fully open source nature (BSD license) ensures a wider accessibility than previous versions of PyNjoy.

KEYWORDS: Nuclear data, DRAGLIB, DRAGON5, SERPENT2

### 1. INTRODUCTION

To solve the transport equation, it is necessary to rely on continuous or multigroup nuclear data, respectively for Monte-Carlo and deterministic codes. Such data can be reconstructed from nuclear data evaluations with specialized codes such as NJOY [1].

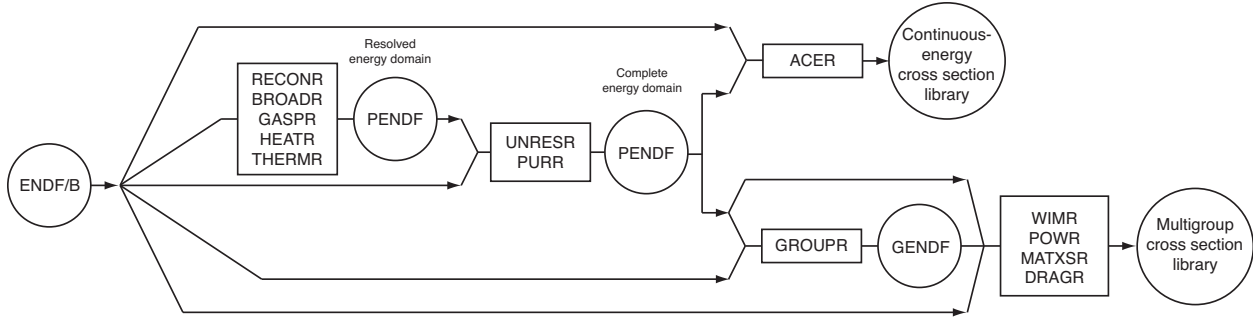
PyNjoy family software are open source Python wrappers, encapsulating calls to the various NJOY modules. It is intended to simplify the use of NJOY and to facilitate the production of complete libraries, for a specific nuclear data evaluation and multigroup energy mesh. In their previous versions, PyNjoy [2] and PyNjoy2012 [3,4] relied respectively on NJOY99 and NJOY2012. As its name suggests, PyNjoy2016 relies on the mature NJOY2016 codebase, with additions such as the DRAGR module. This module produces DRAGLIB files, through which DRAGON5 can access the evaluated nuclear data. PyNjoy2016 can also produce ACE files for Monte-Carlo codes such as SERPENT2.

For the sake of completeness, let us mention that IRSN developed another closed source wrapper for NJOY, GAIA, which is not dedicated to the production of libraries for DRAGON5 nor for SERPENT2 specificities (that will be discussed in section 4), but rather for MCNP and in-house MORET [5] codes.

### 2. PYNJOY2016 DEVELOPMENT AND REGRESSION VERIFICATION

The two components of PyNjoy2016 are :

- (1) functionalities in Python (as suggested by its name), that control the data flow between the different NJOY modules (see Figure 1),



**FIGURE 1 – PyNjoy2016 data flow for producing continuous-energy and multigroup libraries.**

(2) a modified version of NJOY2016. The main modification is the introduction of the DRAGR module. We will see, in this section, that an improvement of THERMR was also mandatory. Finally, some small changes (detailed in section 6) are also included to process famous but incorrect nuclear data files.

During the transition from PyNjoy2012 to PyNjoy2016, the bulk of the work was checking thoroughly for regressions. This was performed on a depleting UOX pincell<sup>1</sup> with a 172-groups mesh and JEFF-3.1.1 nuclear data. On this case, a  $k_{\text{eff}}$  discrepancy of 16 pcm, at most, was found between PyNjoy2012 and PyNjoy2016, as can be seen on line (A) of table 1.

Most of this discrepancy was identified as being due to thermal scattering. As a matter of fact, NJOY uses an insufficient discretization of the incident energies, in the `calcem` subroutine of THERMR :

- NJOY99.0 used a 59 points discretization, covering up to 5.1 eV.
- In 2004, this discretization was extended to 10 eV and refined, reaching 117 points, in NJOY99.93 [8]. In 2016, it was slightly adjusted to 118 points in NJOY2012.78. This discretization is still used in the latest versions of NJOY2016.
- In 2006, Yamamoto and Sugimura [9] demonstrated that the 59-point discretization brought large errors even on extremely simple systems, such as an infinite homogeneous mixture of uranium 238 and hydrogen. They also showed that 175 points were not sufficient and that the errors due to the discretization became acceptable at 349 points. The discretization they propose covers up to 5.1 eV, as does the one of NJOY99.0. Based on these results, Hébert, in 2008, started to use this 349-point discretization to produce DRAGLIB with NJOY99, encapsulated in PyNjoy.
- In the transition from PyNjoy to PyNjoy2012, Hébert merged the two previous discretizations. He used the 349-point discretization, to which he added the 8 points above 5.1 eV from the discretization of NJOY2012.78, up to 10 eV, thus leading to a 357-point discretization.

**TABLE 1 – Effective multiplication factor ( $k_{\text{eff}}$ ) discrepancies versus PyNjoy2012, in  $\Delta \ln k_{\text{eff}}$  (pcm). The exact  $k_{\text{eff}}$  values behind these discrepancies can be found in ref. [6] (appendix D).**

Burnup (MW·day per ton of initial heavy metal)	0	36,8	1000	2500	5000
(A) NJOY2016 with 118 points in THERMR (default)	15,9	15,5	4,7	8,2	10,5
(B) NJOY2016 with 357 points in THERMR	-0,4	-0,4	-0,2	-0,3	-0,3
(C) NJOY2016 with 357 points in THERMR and backporting the universal physical constants from NJOY2012	0	0	0	0	0
(D) NJOY99 with 357 points in THERMR	0	0	0	0	0

1. Non-regression TCDR04 case of DRAGON5, revision 1834, originally described in ref. [7].

When using the latter 357-point discretization in PyNoy2016, the discrepancies with PyNjoy2012 are reduced to fractions of pcm, as indicated on line (B) of table 1. Note that Monte-Carlo codes are also affected by this THERMR deficiency, as can be seen on Figure 1. Even though it is smaller here, this error can amount to hundreds of pcm [9] in more thermalized systems. Therefore, we strongly recommend an increase of this discretization, without waiting for a complete overhaul of THERMR : meanwhile, NJOY users are getting a biased answer.

The residual discrepancies were found to be induced by the update of the universal physical constants, as can be seen on line (C) of table 1. Furthermore, not a single digit could be found to be different (with `diff`) between the DRAGLIB files produced by PyNjoy, PyNjoy2012 or PyNjoy2016, as illustrated on lines (C) and (D). This implies that no DRAGON5 calculation could be different, largely beyond that pincell case.

In conclusion, the objective is therefore achieved. The extensive comparisons now show discrepancies that are either duly justified (B option) or nil. The version of PyNjoy2016 openly available online [10] corresponds to the (B) option, retaining the most up-to-date physical constants. Very small deviations are therefore expected to occur, compared to PyNjoy2012.

### 3. CONSISTENCY AS AN INDISPENSABLE INGREDIENT FOR ASSESSING DETERMINISTIC BIASES

Since an ACE file does not depend on the approximations commonly used in deterministic methods (such as multigroup meshes or various dilutions necessary for self-shielding models), it is tempting to think that an evaluation of nuclear data can only result in a single ACE file. However, the possible ACE files are far from unique. They depend, in a non-negligible way, on the version of NJOY and the options used. Comparing deterministic and Monte-Carlo results obtained from any ACE files (for example, those provided with the code) therefore implies inconsistencies, since the version of NJOY and the options used are almost inevitably different from those used for the deterministic library. Discrepancies therefore appear as a result of these differences. This error is most often incorrectly attributed to the deterministic code, on account of its deterministic bias, which is then inaccurately evaluated. The vast majority of comparisons between Monte-Carlo and deterministic codes available in the public and industrial literature are affected by these inconsistencies, with a few notable exceptions, such as the CRISTAL V2 criticality-safety package [11].

To avoid this pitfall, it is essential to produce both types of library at the same time, which is a notable effort. Fortunately, PyNjoy has had a rudimentary ability to produce ACE files since its inception [12] (excluding for instance thermal scattering laws). Bringing this dormant feature back into use has resulted in maximum consistency : the same run of NJOY, in the same version and with the same options, produces both multigroup (DRAGLIB) and continuous-energy (ACE) libraries, as illustrated on Figure 1.

For example, SERPENT2 yields 100 pcm difference in the effective multiplication factor, for each of the assemblies present at the first start-up of Tihange-1 [13], between using :

- the ACE files produced with PyNjoy2016, or
- the ACE files delivered by default with SERPENT2,

based on JEFF-3.1.1 in both cases. To our knowledge, this order of magnitude is rather common.

### 4. ENERGY DEPOSITION MODEL IN SERPENT2

In SERPENT2, by default, energy is only deposited at fission sites, incorporating a constant and crude estimate of the energy deposited by the other reactions such as  $(n,\gamma)$ , thus leading to spatially inaccurate results [14].

When comparing codes, the most common attitude regarding this approximate model is to ignore the difference with the model of the compared code [15,16,17]. Here again, a deterministic code is often blamed for this discrepancy, erroneously labeled as deterministic bias. Another attitude, already less common, is to level down the compared code [18], making it follow this simplified SERPENT2 model. Here and as Ramamoorthy [12], a solution to this problem has been sought : without this, the result of the Monte-Carlo method can no longer be considered as a reference result.

Fortunately, SERPENT2 also features an advanced energy deposition model, but for that, it relies on mixed ACE/ENDF-6 files that are not widely available. PyNjoy2016 has the ability to produce such ACE/ENDF-6 files. As illustrated in Figure 2, a correct energy deposition model is a prerequisite for the correct computation of power distributions. Any depletion comparison will also be affected : since burnup is an amount of energy deposited (MW-day) per unit mass (ton), it is important to correctly estimate the deposited energy. If it is incorrectly estimated, a drift will appear during depletion. This remark is even more important in the presence of a strong absorber such as control rods or burnable absorbers.

Finally, the unresolved resonance range has no tangible impact on our typical use cases (water reactors), but is nevertheless properly taken into account in PyNjoy2016 for potential future uses on systems with larger proportions of fast or epithermal neutrons.

## 5. PRODUCTION OF LIBRARIES FOR TOTAL MONTE-CARLO

The reference solution for uncertainty propagation is the Total Monte-Carlo method, as it avoids the Gaussian and response linearity approximations employed by perturbation theory. More generally, the sampling approach chosen has the following advantages.

- It takes place at the level of the ENDF-6 files. All implicit effects (prior to self-shielding) can be taken into account. Distinctions between implicit and explicit effects are even irrelevant, which evacuates the need for approximations and *ad hoc* treatments [19]. Furthermore, a perturbation at the multigroup library level would induce approximations [20], here avoided, to differentiate the contributions within a combined scattering matrix, summing the elastic, inelastic, (n,2n) and (n,3n) contributions. These complications are accentuated when these libraries are delivered, non-reproducible, in a proprietary and binary format.
- Each sample can be propagated indistinctly through any other physics, such as stationnary or transient thermal-hydraulics, fuel heat transfer, etc. No additional modifications or approximations are required.
- All isotopes can be sampled at the same time. This allows to capture the effect of interactions, between isotopes, of their respective uncertainties.

For such an uncertainty propagation, PyNjoy2016 has the capability to produce randomly sampled ACE and DRAGLIB files on multiple CPUs (in parallel), taking as a basis either :

- presampled TENDL files in ENDF-6 format, sampled upstream of its nuclear physics models, which leads to continuously varying uncertainties that are also not necessarily Gaussian ;
- ENDF-6 files sampled Gaussianly within multigroup covariance matrices. For that purpose, SANDY [21] is called on-the-fly. This path is more adventurous, as these ENDF-6 covariance formats are more heterogeneous than the homogeneous TENDL formats (derived from a single code chain).

Examples of cross sections and their uncertainties are shown in Figures 3, 4 and 5. Also, an example of such an uncertainty propagation is presented in a companion paper [22].

## 6. SHALL WE MERGE DRAGR INTO NJOY2016 MAIN BRANCH?

Since "the mission of NJOY is to take basic data from the nuclear data library and convert it into the forms needed for applications" [1], we believe the DRAGR module should be integrated into NJOY2016 main branch. Without such an inclusion, the NJOY community may become increasingly divided, having more different versions being used, instead of benefiting from a community effect.

To enable this objective to be reached, we strive to meet the highest development standards for contributors. Therefore, we added a non-regression test similar to those of NJOY2016, but specific to DRAGR. This test :

- uses the same ENDF-6 files already used in the non-regression testing of NJOY2016 ; it thus does not burden its repository ;
- is as short as possible. It does not include all the isotopes of an evaluation (typically between 300 and 600 isotopes), however all possible use cases are covered. The nine used isotopes are :  $^1\text{H}$  bounded in light water,  $^{10}\text{B}$ ,  $^{58}\text{Fe}$ ,  $^{239}\text{Pu}$ ,  $^{240}\text{Pu}$ ,  $^{83}\text{Kr}$ ,  $^{119}\text{Sn}$ ,  $^{90}\text{Zr}$  and  $^{109}\text{Ag}$ .

Also, DRAGR is thoroughly documented, in a  $\text{\LaTeX}$  section in NJOY documentation format [23].

Finally, it is understandable that some small changes cannot be merged into NJOY2016, such as the few (sixteen) lines useful to process famous but incorrect nuclear data files. More specifically, these adaptations in NJOY are required to process existing evaluations with format errors (in MODER and RECONR) or resulting in negative, unphysical cross sections in the unresolved resonance range (in UNRESR and PURR). Such a low number of lines can anyway live outside NJOY2016's repository.

## 7. CONCLUSIONS

During the course of a Ph.D. [6], we developed a software dedicated to the production of :

- multigroup libraries in DRAGLIB format, for the DRAGON5 deterministic code ;
- continuous-energy ACE files, supporting the deposition energy model of SERPENT2.

DRAGON5 deterministic bias can now be precisely obtained by consistently comparing its results with SERPENT2. Such a consistency is a rare feature provided by the simultaneous library production : the same NJOY2016 run, in the same version and with the same options are used to produce both libraries.

The open nature of NJOY2016 implies that PyNjoy2016 can be fully open source (BSD license) and is therefore easily available [10]. It is usefully complementing the state-of-the-art DRAGON5 deterministic lattice code, also open source and, in that sense, unique.

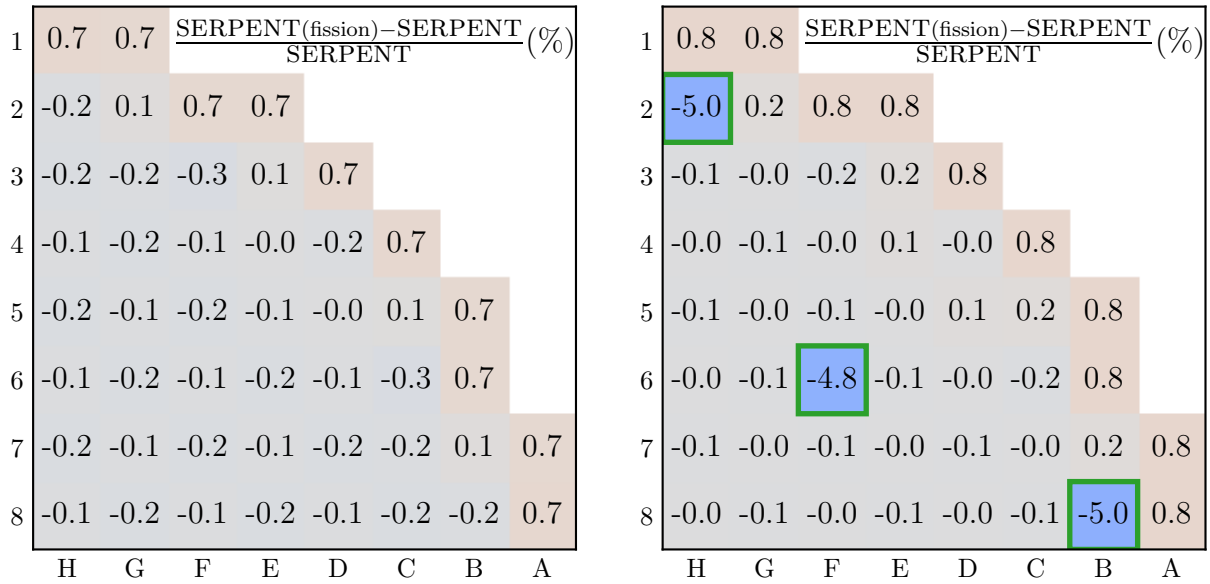
## ACKNOWLEDGEMENTS

The authors are grateful to Dimitri Rochman (PSI) for his randomly sampled advices, Riku Tuominen (VTT) for sharing with us his `runnjoy_kermas.pl` Perl script that enables SERPENT2 precise energy deposition model and Paul Romano (ANL) for his fruitful remarks on THERMR.

## REFERENCES

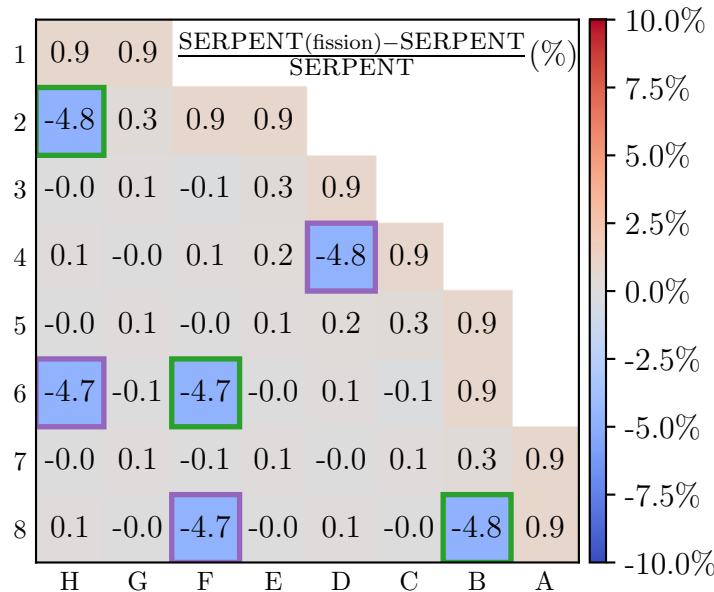
- [1] A. C. Kahler, D. W. Muir, R. M. Boicourt, J. L. Conlin, and W. Haeck. "The NJOY Nuclear Data Processing System, Version 2016." Technical Report LA-UR-17-20093, Los Alamos National Laboratory (2018). <https://github.com/njoy/NJOY2016-manual/raw/master/njoy16.pdf>.
- [2] A. Hébert. "A nuclear data library production system for advanced lattice codes." *International Conference on Nuclear Data for Science and Technology 2007*. <https://nd2007.edpsciences.org/articles/ndata/pdf/2007/01/ndata07253.pdf>.
- [3] A. Hébert. "PyNjoy-2012 : A system for producing cross-section libraries for the DRAGON lattice code." *International Conference on Nuclear Data for Science and Technology 2016*. [https://www.researchgate.net/publication/308566629\\_PyNjoy-2012\\_A\\_system\\_for\\_producing\\_cross-section\\_libraries\\_for\\_the\\_DRAGON\\_lattice\\_code](https://www.researchgate.net/publication/308566629_PyNjoy-2012_A_system_for_producing_cross-section_libraries_for_the_DRAGON_lattice_code).
- [4] <http://merlin.polymtl.ca/pynjoy2012.htm>.
- [5] W. Monange and A. Bardelay. "New capabilities of the MORET 6 Monte Carlo neutron transport code." In *International Conference on Physics of Reactors 2022 (PHYSOR 2022)*. Pittsburgh, United States of America (2022).
- [6] V. Salino. *Incertitudes et ajustements de données nucléaires au moyen de méthodes déterministes, probabilistes et de mesures effectuées sur des réacteurs à eau sous pression*. Ph.D. thesis, École Polytechnique de Montréal (2022). [https://publications.polymtl.ca/10545/1/2022\\_VivianSalino.pdf](https://publications.polymtl.ca/10545/1/2022_VivianSalino.pdf).
- [7] R. D. Mosteller, R. C. Little, L. D. Eisenhart, and J. Chao. "Benchmark calculations for the Doppler coefficient of reactivity." *Nuclear Science and Engineering*, **volume 107 :3** (1991). <https://www.osti.gov/biblio/6017499>.
- [8] M. Mattes and J. Keinert. "Thermal Neutron Scattering Data for the Moderator Materials H<sub>2</sub>O, D<sub>2</sub>O and ZrH<sub>x</sub> in ENDF-6 Format and as ACE Library for MCNP(X) Codes." Technical Report INDC(NDS)-0470, IAEA (2005). <https://www-nds.iaea.org/publications/indc/indc-nds-0470.pdf>.

- [9] A. Yamamoto and N. Sugimura. “Improvement on multi-group scattering matrix in thermal energy range generated by NJOY.” *Annals of Nuclear Energy*, volume 33(6) (2006). <https://www.sciencedirect.com/science/article/pii/S0306454906000089>.
- [10] <https://github.com/IRSN/PyNjoy2016>.



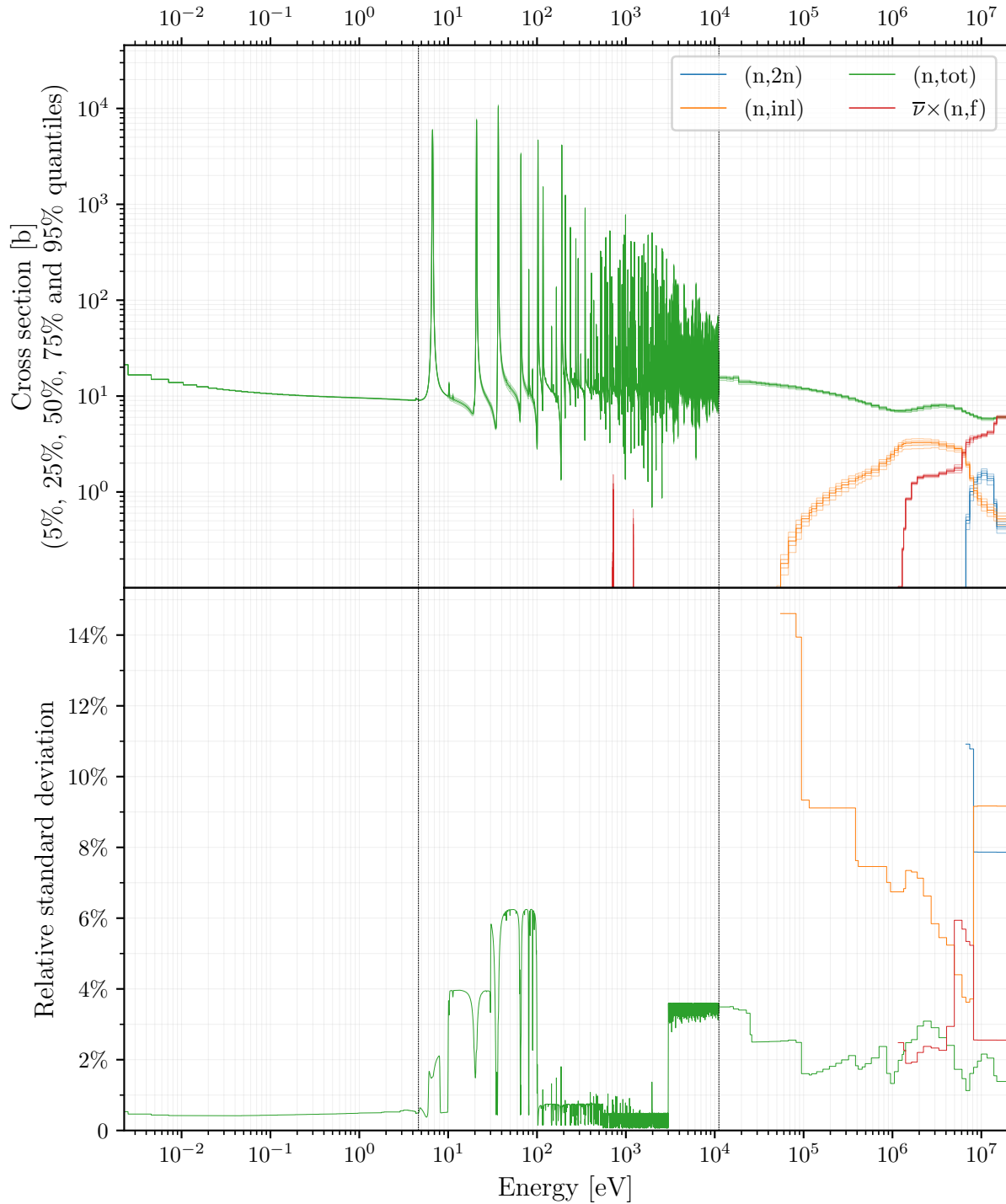
(a) All rods out.

(b) Control rod bank D inserted (green boxes).



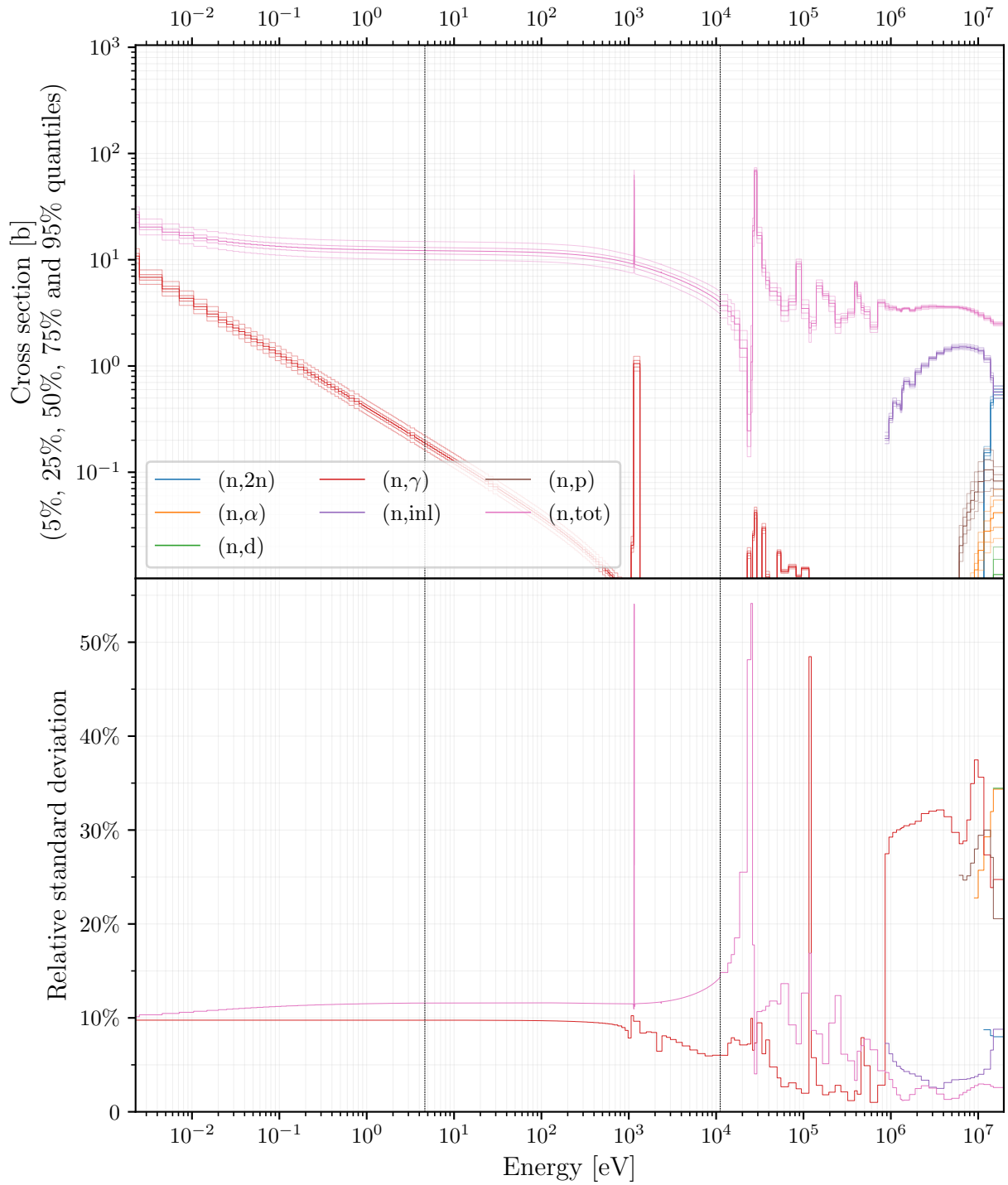
(c) Control rod banks C and D inserted (purple and green boxes, respectively).

**FIGURE 2 – Relative differences in power distributions between two SERPENT tallies options, tallying fission only or all reactions, on the first start-up of Tihange-1 PWR [6,13]. Produced with JEFF-3.3 nuclear data, SERPENT version 2.1.32 and serpentTools for data extraction [24]. Plotting instructions [25] as well as SERPENT datasets, ACE and outputs are open data [26], made publicly available for reproducibility purposes or any other reuse.**

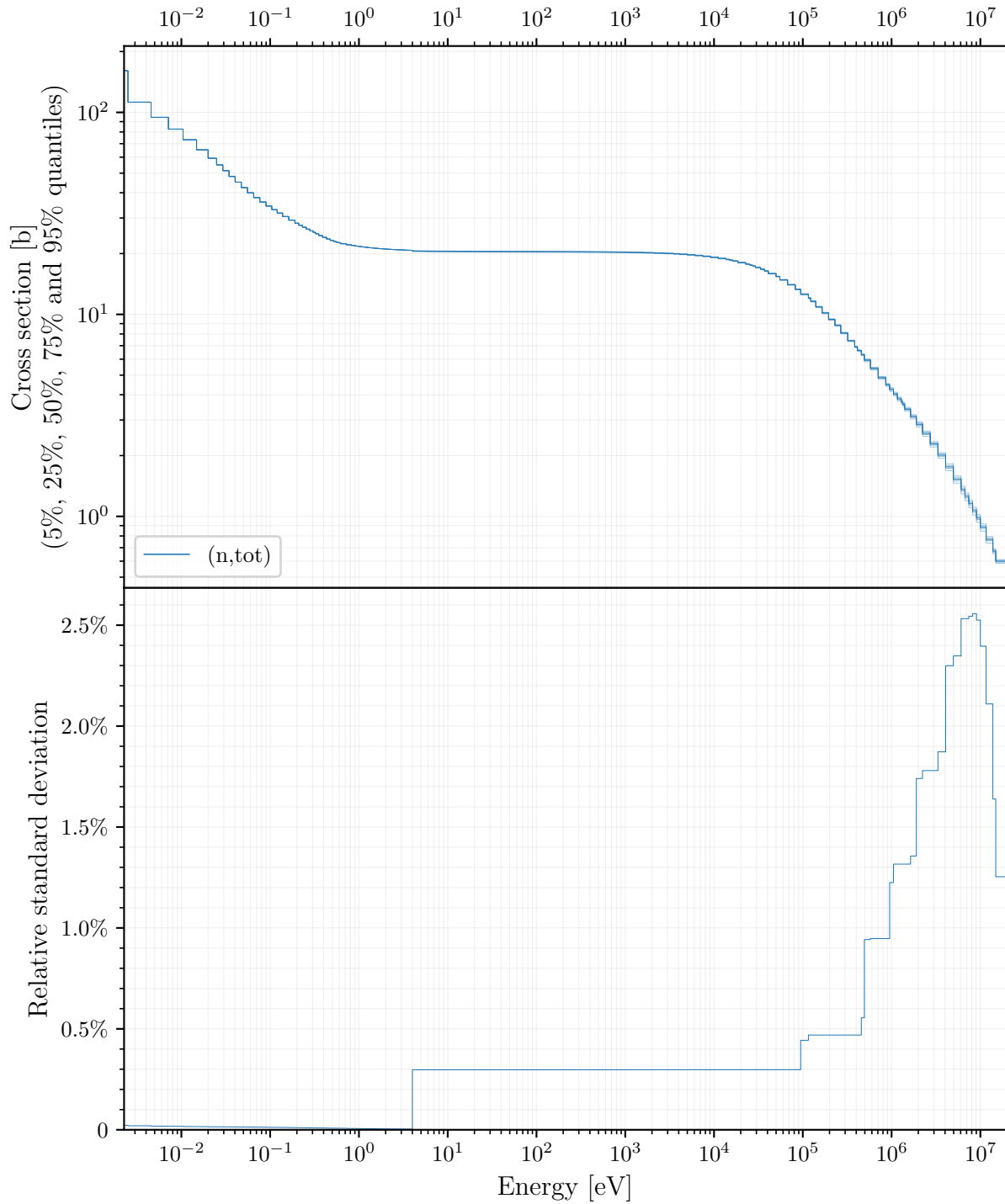


**FIGURE 3 – Uranium 238 cross sections statistics of 300 samples within JEFF-3.3. The temperature is 550 K (276.85°C). The continuous plotted data are those contained in the DRAGLIB files, as used by DRAGON5, allowing it to reach excellent accuracies [27] (computing on-the-fly mathematical probability tables for self-shielding subgroup methods). Plotting instructions are openly available [28], along with the released data itself [26].**





**FIGURE 4 – Iron 56 cross sections statistics from 300 TENDL-2019 files, presampled upstream of its nuclear physics models. The temperature is 550 K (276.85°C). Plotting instructions are openly available [28], along with the released data itself [26].**



**FIGURE 5 – Hydrogen in H<sub>2</sub>O cross sections statistics of 300 samples within JEFF-3.3. The temperature is 573.6 K (300.45°C). At this temperature and at the time of this work, no uncertainties evaluation of the thermal scattering laws  $S(\alpha, \beta)$  of light water was available. This lack explains the zero standard deviation below 4 eV. Since then, an evaluation of its uncertainties has been proposed in TENDL-2021 [29]. Plotting instructions are openly available [28], along with the released data itself [26].**

- [11] J. Gomit, I. Duhamel, Y. Richet, A. Entringer, C. Magnaud, F. Malouch, and . Carmouze C. “CRISTAL v2 : new package for criticality calculations.” In *Nuclear Criticality Safety Division Topical Meeting (NCS D 2017)*. Nuclear Criticality Safety Division Topical Meeting (NCS D 2017), Carlsbad, United States (2017). <https://hal.archives-ouvertes.fr/hal-02417819>.
- [12] K. Ramamoorthy. *Development and validation of burnup dependent computational schemes for the analysis of assemblies with advanced lattice code*. Ph.D. thesis, École Polytechnique de Montréal (2006). [http://merlin.polymtl.ca/downloads/Thesis\\_karthikeyan.pdf](http://merlin.polymtl.ca/downloads/Thesis_karthikeyan.pdf).
- [13] H. Panek. *Qualification du système Neptune - interprétation d'expériences critiques, calculs de cœur de réacteur de puissance - ébauche d'une nouvelle chaîne de calcul basée sur les codes APOLLO et TORTISE - premiers tests sur cœur de PWR - CEA-N-2092*. Ph.D. thesis, Université Paris-Sud (1980). [https://inis.iaea.org/collection/NCLCollectionStore/\\_Public/11/511/11511367.pdf](https://inis.iaea.org/collection/NCLCollectionStore/_Public/11/511/11511367.pdf).
- [14] R. Tuominen, V. Valtavirta, and J. Leppänen. “New energy deposition treatment in the Serpent 2 Monte Carlo transport code.” *Annals of Nuclear Energy*, **volume 129** (2019). <https://www.sciencedirect.com/science/article/pii/S0306454919300726>.
- [15] A. Canbakan. *Validation d'un nouveau calcul de référence en évolution pour les réacteurs thermiques*. mémoire de maîtrise, École Polytechnique de Montréal (2014). [https://publications.polymtl.ca/1480/1/2014\\_AxelCanbakan.pdf](https://publications.polymtl.ca/1480/1/2014_AxelCanbakan.pdf).
- [16] A. Canbakan and A. Hébert. “Accuracy of a 2-level scheme based on a subgroup method for pressurized water reactor fuel assembly models.” *Annals of Nuclear Energy*, **volume 81** (2015). <https://www.sciencedirect.com/science/article/pii/S0306454915001681>.
- [17] L. Ghasabyan. *Validation of DRAGON5 lattice code for PWR applications using depletion benchmarks by detailed comparison with SERPENT2 Monte Carlo code*. mémoire de maîtrise, École Polytechnique de Montréal (2020). [https://publications.polymtl.ca/5527/1/2020\\_LevonGhasabyan.pdf](https://publications.polymtl.ca/5527/1/2020_LevonGhasabyan.pdf).
- [18] A. Bennett, N. Martin, and M. Schneider. “Validation of APOLLO2-A Against SERPENT 2 on BWR Lattices.” In *M&C 2019*. Portland, United States of America (2019). [https://www.researchgate.net/publication/335987968\\_Validation\\_of\\_APOLLO2-A\\_Against\\_SERPENT\\_2\\_on\\_BWR\\_Lattices](https://www.researchgate.net/publication/335987968_Validation_of_APOLLO2-A_Against_SERPENT_2_on_BWR_Lattices).
- [19] M. Hursin, S. Pelloni, A. Vasiliev, and H. Ferroukhi. “Treatment of the implicit effect in Shark-X.” *Annals of Nuclear Energy*, **volume 138** (2020). <https://www.sciencedirect.com/science/article/pii/S0306454919306887>.
- [20] W. Wieselquist, T. Zhu, A. Vasiliev, and H. Ferroukhi. “PSI Methodologies for Nuclear Data Uncertainty Propagation with CASMO-5M and MCNPX : Results for OECD/NEA UAM Benchmark Phase I.” *Science and Technology of Nuclear Installations* (2013). <https://www.hindawi.com/journals/stni/2013/549793>.
- [21] L. Fiorito, G. Žerovnik, A. Stankovskiy, G. V. den Eynde, and P. Labeau. “Nuclear data uncertainty propagation to integral responses using SANDY.” *Annals of Nuclear Energy*, **volume 101** (2017). <http://www.sciencedirect.com/science/article/pii/S0306454916305278>.
- [22] V. Salino, D. Rochman, E. Dumonteil, F. Malvagi, and A. Hébert. “Nuclear data uncertainties and adjustments using deterministic and Monte-Carlo methods along with PWR measurements.” In *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C)*. Niagara Falls, Canada (2023).
- [23] [http://merlin.polymtl.ca/downloads/njoy12\\_rev.pdf](http://merlin.polymtl.ca/downloads/njoy12_rev.pdf).
- [24] A. E. Johnson, D. Kotlyar, S. Terlizzi, and G. Ridley. “serpentTools : A Python Package for Expediting Analysis with Serpent.” *Nuclear Science and Engineering*, **volume 194** (2020). <https://doi.org/10.1080/00295639.2020.1723992>.
- [25] <https://github.com/IRSN/SalinoPhD/blob/master/Plots/PowDifferenceTih.py>.
- [26] <https://github.com/IRSN/SalinoPhD>.
- [27] A. Hébert. “Development of the Subgroup Projection Method for Resonance Self-Shielding Calculations.” *Nuclear Science and Engineering*, **volume 162** (2009). <https://doi.org/10.13182/NSE162-56>.
- [28] <https://github.com/IRSN/SalinoPhD/blob/master/Plots/Draglib.py>.
- [29] [https://tendl.web.psi.ch/tendl\\_2021/randomTSL.html](https://tendl.web.psi.ch/tendl_2021/randomTSL.html).