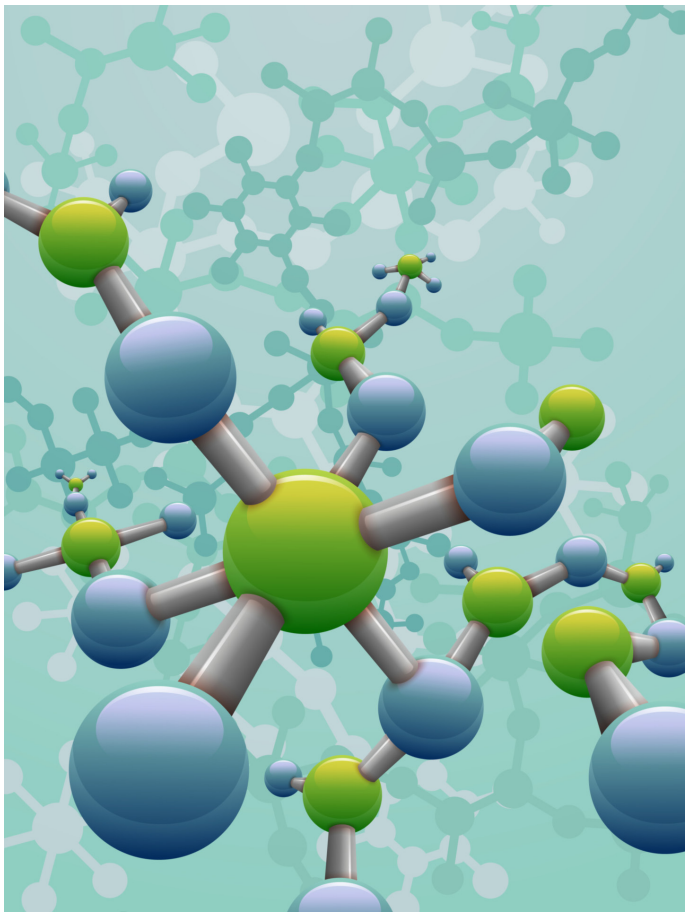
RESEARCH REPORT

VTT-R-00075-25



Development of cross section uncertainty propagation methodology for Monte-Carlo code Serpent 2.2

Authors: Lauri Vaara

Confidentiality: VTT Public

Version: 13.1.2025

Report's title	
Development of cross section uncertainty propagation methodology for Monte-Carlo code Serpent 2.2	
Customer, contact person, address	**Order reference**
Valtion ydinjätehuoltorahasto	SAFER 11/2024
Project name	**Project number/Short name**
Neutronics for Fuel Outside the Reactor Core	137938/NOTCO
Author(s)	**Pages**
Lauri Vaara	16/9
Keywords	**Report identification code**
Uncertainty propagation, Particle transport, Monte-Carlo Calculations, Nuclear data uncertainty	VTT-R-00075-25

Summary

The development and early phase testing of a new nuclear data uncertainty propagation method for Monte-Carlo burnup calculations is presented in this paper. Specifically, the method is aimed at propagating actinide cross section uncertainties from multi-group covariance matrices to nuclide densities in particle transport code Serpent 2.2.

The method was applied to propagate the U238(n,γ)U239 cross section uncertainty in a simple two-dimensional one-depletion-zone Serpent model. A reference solution for the uncertainties was acquired with the established Total Monte-Carlo (TMC) approach, which utilized 200 perturbed U238 files from the JEFF3.3 [4] library. The perturbed files were produced with the SANDY program. The studied responses include the nuclide densities of U238, U239, Pu239, Pu240, Pu241 and Am241.

The developed method showed relatively good performance up to approximately 15 MWd/kgU burnup. The failure to reproduce the TMC results at higher burnups can be associated to various approximations used in the uncertainty propagation framework.

Confidentiality	VTT Public

Espoo 13.1.2025

Written by	**Reviewed by**
Lauri Vaara, Research Scientist	Ville Valtavirta, Senior Scientist

VTT's contact address

VTT Technical Research Centre of Finland Ltd, P.O. Box 1000, FI-02044 VTT, FINLAND

Distribution (customer and VTT)

Customer company/Contact Name, 1 copy
VTT archive, 1 copy

Approval

VTT TECHNICAL RESEARCH CENTRE OF FINLAND LTD

Date:	3.2.2025
Signature:	Häkkinen Silja 912690884 Digitally signed by Häkkinen Silja 912690884 Date: 2025.02.07 15:31:13 +02'00'
Name:	Silja Häkkinen
Title:	Research Team Leader

Contents

1. Introduction

Quantifying uncertainties in nuclide number densities is crucial for accurately estimating decay heat and other characteristics of spent nuclear fuel (SNF). The Serpent Monte Carlo code [1], widely used for particle transport and burnup calculations, currently provides only mean values for nuclide number densities. However, understanding the uncertainties associated with these values is essential for safety assessments, reactor operation, and waste management.

Uncertainty quantification involves propagating input uncertainties, such as those in nuclear data, through the computational process to estimate the variability in outputs. A key aspect of this is the covariance matrix of nuclide number densities, which describes how uncertainties in different nuclides are correlated. These correlations directly affect critical SNF parameters, such as decay heat.

The Total Monte Carlo (TMC) method is often used as a reference solution for estimating the covariance matrix due to its high accuracy. However, TMC's reliance on extensive simulations with perturbed nuclear data demands significant computational resources, making it slow and inefficient for practical applications. This limitation motivates the development of a new, more efficient method.

This work introduces and evaluates an in-house method for estimating the covariance matrix of nuclide number densities. Unlike TMC, the proposed method aims to achieve efficiency by leveraging iterative updates to key sensitivity parameters during the burnup process. The in-house method is benchmarked against TMC to ensure accuracy, with a focus on assessing its capability to predict decay heat uncertainties effectively.

By addressing the computational challenges of uncertainty quantification, this study aims to provide a robust and practical framework for SNF analysis. The outcome will enhance the ability to assess safety margins and optimize fuel cycle operations while significantly reducing computational demands.

2. Theoretical background

As of the current implementation, the Serpent Monte-Carlo particle transport code only computes the first moments of nuclide number densities, specifically the mean values. To derive uncertainty estimates for decay heat and other characteristics of spent fuel, the second moment of the joint probability distribution is required, namely the covariance matrix of the nuclide vector $\mathbf{N}$.

Before progressing into nuclear physics, the uncertainty propagation formula is derived. This formula is the main tool on which rest of the mathematical framework relies on.

First, consider a general multi-variable function $f : \mathbb{R}^{1\times l} \to \mathbb{R}^{1\times k}$, for which the partial derivatives with respect to the input parameters are well-behaving throughout the domain of the variables. Perturbing the input parameters with the perturbation vector $\delta\mathbf{X}$ results in a response perturbation $\delta\mathbf{Y}$, which can be approximated using the first-order Taylor expansion

$$\mathbf{Y} + \delta\mathbf{Y} \approx f(\mathbf{X}) + \mathbf{J_f}\delta\mathbf{X} \tag{2.1}$$

where $\mathbf{J_f}$ is the Jacobian matrix of the function f. The approximation holds well if the response is approximately linear, or if the perturbations are small with respect to the magnitude of their respective variables.

Next, consider the $\mathbf{X}$ a vector of random variables, which is fully described by its multivariate probability distribution. The first moment of this distribution is the mean vector $\mathbb{E}[\mathbf{X}]$. The perturbation $\delta\mathbf{X}$ is defined as

the deviation from the mean vector

$$\delta\mathbf{X} = \mathbf{X} - \mathbb{E}[\mathbf{X}]. \tag{2.2}$$

The response perturbation $\delta\mathbf{Y}$ is defined in a similar manner

$$\delta\mathbf{Y} = \mathbf{Y} - \mathbb{E}[\mathbf{Y}]. \tag{2.3}$$

Next, consider the covariance matrix of $\mathbf{Y}$. The covariance of $\mathbf{Y}$ is by definition

$$\text{cov}(\mathbf{Y}) = \mathbb{E}\big[(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])^T\big]. \tag{2.4}$$

Substituting expression 2.3 into 2.4, yields

$$\text{cov}(\mathbf{Y}) = \mathbb{E}[\delta\mathbf{Y}\,\delta\mathbf{Y}^T]. \tag{2.5}$$

Using $\delta\mathbf{Y} \approx \mathbf{J_f}\delta\mathbf{X}$, the above expression can be rewritten as

$$\begin{aligned} \text{cov}(\mathbf{Y}) &\approx \mathbb{E}[\mathbf{J_f}\delta\mathbf{X}\,(\mathbf{J_f}\delta\mathbf{X})^T] && (2.6)\\ &\approx \mathbb{E}[\mathbf{J_f}\delta\mathbf{X}\,\delta\mathbf{X}^T\mathbf{J_f}^T]. && (2.7)\\ &\approx \mathbf{J_f}\mathbb{E}[\delta\mathbf{X}\,\delta\mathbf{X}^T]\mathbf{J_f}^T. && (2.8) \end{aligned}$$

Since $\text{cov}(\mathbf{X}) = \mathbb{E}[\delta\mathbf{X}\,\delta\mathbf{X}^T]$, we arrive at the final expression for the covariance of $\mathbf{Y}$:

$$cov(\mathbf{Y}) \approx \mathbf{J_f}\, cov(\mathbf{X})\, \mathbf{J_f}^T. \tag{2.9}$$

Equation 2.9 is the uncertainty propagation formula, that is widely used in this research work. The equation shows how the covariance of the output vector $\mathbf{Y}$ is related to the covariance of the input variable vector $\mathbf{X}$ through the Jacobian matrix of the transformation f. The search for the uncertainty in number densities, decay heat and other SNF characteristics is hence narrowed down to finding the expression for the Jacobian matrix of the response. For certain cases it is a straightforward process, as seen in the following section.

2.1 Decay heat uncertainty

The decay heat *DH* of the spent nuclear fuel at time t is the product of the number density, decay rate and the energy release per decay - vectors. In mathematical terms this statement can be written as

$$DH(t) = \mathbf{N}(t) \cdot (\mathbf{E} \odot \lambda), \tag{2.10}$$

where $\mathbf{N}(t)$ is the number density vector at time t, $\mathbf{E}$ the average energy release per decay per nuclide and λ the decay constant vector. The $\odot$ symbol denotes the Hadamard product, indicating element-wise multiplication between $\mathbf{E}$ and λ. The time evolution of $\mathbf{N}(t)$ is governed by the Bateman depletion equation

$$\mathbf{N}(t) = e^{\mathbf{A}t}\mathbf{N}_0 \tag{2.11}$$

where $\mathbf{A}$ is the decay matrix consisting of microscopic decay rates, and $\mathbf{N}_0 = \mathbf{N}(\text{t=0})$ is the number density vector at time zero. Combining equations 2.10 and 2.11 together we obtain the equation for the time evolution of the decay heat

$$DH(\mathbf{N_0}, t) = e^{\mathbf{A}t}\mathbf{N}_0 \cdot (\mathbf{E} \odot \lambda) \tag{2.12}$$

We are interested in investigating how the uncertainty in $\mathbf{N}_0$ affects the decay heat uncertainty as a function of t. Since the right-hand side in 2.12 maps a vector to a scalar, the Jacobian matrix for the system is a vector $\mathbf{J}_{DH} \in \mathbb{R}^{1\times k}$, and the uncertainty propagation formula 2.9 reduces to

$$\sigma^2_{DH}(\mathbf{N_0}, t) = \mathbf{J}_{DH} Cov(\mathbf{N}_0)\mathbf{J}^T_{DH}. \tag{2.13}$$

where σ^2_{DH} is the variance of the decay heat. To calculate the Jacobian $\mathbf{J}_{DH}$ for the right hand side of the equation 2.12, we introduce the gradient operator $\nabla_{\mathbf{N}_0} = \left[\frac{\partial}{\partial N^1_0}, \frac{\partial}{\partial N^2_0}, ...\right]$, which consists of partial derivative operators with respect to the number density of each nuclide at time zero. By the definition of the Jacobian, we have

$$\mathbf{J}_{DH} = \nabla_{\mathbf{N_0}} DH(\mathbf{N_0}, t) \tag{2.14}$$

Substituting equation 2.12 into 2.14, yields

$$\begin{aligned}
\mathbf{J}_{DH} &= \nabla_{\mathbf{N}0}\left(e^{\mathbf{A}t}\mathbf{N}_0 \cdot (\mathbf{E} \odot \lambda)\right) && (2.15)\\
&= \nabla_{\mathbf{N}0}\mathbf{N}^T_0\left(e^{\mathbf{A}t}\right)^T \cdot (\mathbf{E} \odot \lambda) && (2.16)\\
&= \mathbb{I}\left(e^{\mathbf{A}t}\right)^T (\mathbf{E} \odot \lambda) && (2.17)\\
&= (\mathbf{E} \odot \lambda)^T e^{\mathbf{A}t} && (2.18)
\end{aligned}$$

Substituting 2.18 into 2.13, we obtain the decay heat variance as a function of cov($\mathbf{N_0}$) and time t. The uncertainty is obtained from the square root of the variance.

$$\sigma^2_{DH}(\mathbf{N_0}, t) = (\mathbf{E} \odot \lambda)^T e^{\mathbf{A}t} Cov(\mathbf{N_0})(e^{\mathbf{A}t})^T(\mathbf{E} \odot \lambda) \tag{2.19}$$

Equation 2.19 can be used to compute decay heat uncertainty, given that the covariance for the nuclide density vector at the end of the irradiation is known. It is important to note that in the derivation the decay matrix was considered to be free of uncertainty (ie. decay data uncertainties were ignored). This means that the equation only captures the time-evolution of the uncertainty from the initial nuclide density vector.

In a similar fashion one can show that the time-evolution of $Cov(\mathbf{N})$ is governed by the equation 2.20

$$Cov(\mathbf{N}_t) = e^{\mathbf{A}t} Cov(\mathbf{N_0})(e^{\mathbf{A}t})^T \tag{2.20}$$

The above equations are useful only in the case that the $Cov(\mathbf{N}_0)$ is available. In the following section two possible methods for the computation of $Cov(\mathbf{N}_0)$ during the particle transport simulation are discussed.

2.2 Number density uncertainty propagation through a burnup calculation

Having established Total Monte-Carlo approach for uncertainty propagation in Serpent [5], the calculation of $Cov(\mathbf{N})$ is already available. The Total Monte-Carlo is based on running hundreds of burnup calculations with perturbed nuclear data inputs, to produce hundreds of result sets, from which variances and covariances can be estimated. The perturbed inputs are produced with the TALYS-based T6 program, and the covariance evaluation based stochastic sampling tool SANDY [2].

From the results set, the non-biased covariance estimate for nuclides N_i and N_j can be calculated with

$$\mathrm{Cov}(N_i, N_j) = \frac{1}{n-1}\sum_{k=1}^{n}(N_{ik} - \bar{N}_i)(N_{jk} - \bar{N}_j) \tag{2.21}$$

The main benefit of this approach is its simplicity, and the readiness of combining dozens of reaction channel uncertainties into the output uncertainty.

The major drawback of the Total Monte-Carlo approach for the generation of $Cov(\mathbf{N})$ is the large computational demand. To achieve satisfactory statistical accuracy for the covariance estimates, the burnup calculation will have to be repeated hundreds of times. This struggle has motivated the development of an in-house uncertainty propagation method, which aims to reduce the computational burden of the Total Monte-Carlo. As of writing, this method is still in early stages, but somewhat promising preliminary results have been obtained.

2.2.1 Basic idea

The method is based on iteratively updating the Jacobian matrix for nuclide densities, with respect to nuclear data. Since cross section uncertainties are given in multi-group covariances, we can apply the uncertainty propagation formula 2.9 to calculate $Cov(\mathbf{N})$, if we are able to find expression for the Jacobian matrix of the system.

To achieve this goal, we introduce the gradient operator, which applies the full derivative operator on every system parameter with respect to every nuclear data parameter

$$\nabla_\sigma = \left(\frac{d}{d\sigma_1}, \frac{d}{d\sigma_2}, \frac{d}{d\sigma_3}, \ldots\right). \tag{2.22}$$

The burnup calculation progresses iteratively from a particle transport step to a depletion step, by first solving the reaction rate estimates with the transport simulation. The reaction rate information is used to compute a new number density vector for the subsequent transport step, for each depletion zone. For simplicity, we consider only one depletion zone and drop the cell dependency from the equations.

The number density in the depletion zone is updated from transport step n to transport step n+1 with the Bateman equation

$$\mathbf{N_{n+1}} = e^{\mathbf{A}_n t}\mathbf{N}_n \tag{2.23}$$

where $\mathbf{A}_n$ is the transmutation matrix, containing both decay constants and microscopic neutron induced reaction rates. For ease of viewing we define matrix $\mathbf{M}_n \equiv e^{\mathbf{A}_n t}$, and apply operator 2.22 on equation 2.23

$$\nabla_\sigma \mathbf{N}_{n+1} = \nabla_\sigma(\mathbf{M}_n\mathbf{N}_n) \tag{2.24}$$

$$= \mathbf{M}_n\nabla_\sigma\mathbf{N}_n + \nabla_\sigma\mathbf{M}_n\mathbf{N}_n \tag{2.25}$$

The $\nabla_\sigma\mathbf{N}$ terms are analogous to the jacobian of the system, consisting of full derivatives of each nuclide density, with respect to each nuclear data parameter

$$\nabla_\sigma\mathbf{N} = \begin{pmatrix} \frac{dN_1}{d\sigma_1} & \frac{dN_1}{d\sigma_2} & \cdots & \frac{dN_1}{d\sigma_n} \\ \frac{dN_2}{d\sigma_1} & \frac{dN_2}{d\sigma_2} & \cdots & \frac{dN_2}{d\sigma_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dN_m}{d\sigma_1} & \frac{dN_m}{d\sigma_2} & \cdots & \frac{dN_m}{d\sigma_n}. \end{pmatrix} \tag{2.26}$$

The boundary condition for the system is that at time zero we must have $\nabla_\sigma\mathbf{N_1} = \mathbf{0}$, because no depletion has occurred at that point in time. Hence equation 2.25 reduces to

$$\nabla_\sigma\mathbf{N}_2 = \nabla_\sigma\mathbf{M}_1\mathbf{N}_1 \tag{2.27}$$

for the first depletion step. The second depletion step can then be written as

$$\nabla_\sigma\mathbf{N}_3 = \mathbf{M}_2\nabla_\sigma\mathbf{N}_2 + \nabla_\sigma\mathbf{M}_2\mathbf{N}_2 \tag{2.28}$$

$$= \mathbf{M}_2(\nabla_\sigma\mathbf{M}_1\mathbf{N}_1) + \nabla_\sigma\mathbf{M}_2\mathbf{N}_2. \tag{2.29}$$

Notice how the $\nabla_\sigma\mathbf{N}_1 = 0$ condition enables us to recursively solve the subsequent $\nabla_\sigma\mathbf{N}_{n+1}$ terms, if expressions for $\mathbf{M}_n, \mathbf{N}_n$ and $\nabla_\sigma\mathbf{M}_n$ are found. The first two are already calculated by Serpent during the burnup calculation, meaning that if we are able to find expression for $\nabla_\sigma\mathbf{M}_n$, we can propagate the $\nabla_\sigma\mathbf{N}$ through the whole burnup calculation.

2.2.2 Computation of $\nabla_\sigma \mathbf{M}_n$

Let us begin by applying the ∇_σ operator element-wise on the elements of $\mathbf{M}_n$

$$\nabla_\sigma \mathbf{M}_n = \begin{pmatrix} \nabla_\sigma M_{11} & \nabla_\sigma M_{12} & \cdots & \nabla_\sigma M_{1m} \\ \nabla_\sigma M_{21} & \nabla_\sigma M_{22} & \cdots & \nabla_\sigma M_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \nabla_\sigma M_{m1} & \nabla_\sigma M_{m2} & \cdots & \nabla_\sigma M_{mm} \end{pmatrix} \tag{2.30}$$

The elements of $\mathbf{M}$ depend on nuclear data via the elements of the transmutation matrix $\mathbf{A}$. The diagonal elements of $\mathbf{A}$ are functions of the off-diagonal elements, hence it is enough to investigate the off-diagonal elements. We can expand the $\nabla_\sigma M_{l,k}$ by applying common derivative rules, omitting the diagonal elements

$$\nabla_\sigma M_{l,k} = \sum_{r \neq s}^{m \, x \, m} \frac{\partial M_{l,k}}{\partial A_{r,s}} \nabla_\sigma A_{r,s}. \tag{2.31}$$

Substituting equation 2.31 into 2.30 yields

$$\nabla_\sigma \mathbf{M}_n = \begin{pmatrix} \sum_{r \neq s}^{m \times m} \frac{\partial M_{1,1}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} & \sum_{r \neq s}^{m \times m} \frac{\partial M_{1,2}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} & \cdots & \sum_{r \neq s}^{m \times m} \frac{\partial M_{1,m}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} \\ \sum_{r \neq s}^{m \times m} \frac{\partial M_{2,1}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} & \sum_{r \neq s}^{m \times m} \frac{\partial M_{2,2}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} & \cdots & \sum_{r \neq s}^{m \times m} \frac{\partial M_{2,m}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{r \neq s}^{m \times m} \frac{\partial M_{m,1}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} & \sum_{r \neq s}^{m \times m} \frac{\partial M_{m,2}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} & \cdots & \sum_{r \neq s}^{m \times m} \frac{\partial M_{m,m}}{\partial A_{r,s}} \nabla_\sigma A_{r,s} \end{pmatrix} = \sum_{r \neq s}^{m \times m} \frac{\partial \mathbf{M}_n}{\partial A_{r,s}} \otimes \nabla_\sigma A_{r,s} \tag{2.32}$$

where $\otimes$ is the kronecker product. Substituting 2.32 into equation 2.25 yields

$$\nabla_\sigma \mathbf{N}_{n+1} = \mathbf{M}_n \nabla_\sigma \mathbf{N}_n + \sum_{r \neq s}^{m \times m} \frac{\partial \mathbf{M}_n}{\partial A_{r,s}} \otimes \nabla_\sigma A_{r,s} \mathbf{N}_n \tag{2.33}$$

which can be rewritten as

$$\nabla_\sigma \mathbf{N}_{n+1} = \mathbf{M}_n \nabla_\sigma \mathbf{N}_n + \sum_{r \neq s}^{m \times m} \frac{\partial \mathbf{M}_n}{\partial A_{r,s}} \mathbf{N}_n \nabla_\sigma A_{r,s}. \tag{2.34}$$

The $\mathbf{N}_n \nabla_\sigma A_{r,s}$ term represents the outer product between vectors $\mathbf{N}_n$ and $\nabla_\sigma A_{r,s}$. We can further expand the $\nabla_\sigma A_{r,s}$ by considering $A_{r,s}$ a function of nuclear data and the number density vector

$$\nabla_\sigma A_{r,s} = \frac{\partial A_{r,s}}{\partial \overline{\sigma}} + \left(\frac{\partial A_{r,s}}{\partial \mathbf{N}_n} \right)^T \nabla_\sigma \mathbf{N}_n \tag{2.35}$$

with this, we arrive at the final expression for the $\nabla_\sigma \mathbf{N}_{n+1}$

$$\nabla_\sigma \mathbf{N}_{n+1} = \mathbf{M}_n \nabla_\sigma \mathbf{N}_n + \sum_{r \neq s}^{m \times m} \frac{\partial \mathbf{M}_n}{\partial A_{r,s}} \mathbf{N}_n \left[\frac{\partial A_{r,s}}{\partial \overline{\sigma}} + \left(\frac{\partial A_{r,s}}{\partial \mathbf{N}_n} \right)^T \nabla_\sigma \mathbf{N}_n \right]. \tag{2.36}$$

The $\frac{\partial \mathbf{M}_n}{\partial A_{r,s}}$ terms can be solved with simple forward perturbation. By omitting the decay constants, the matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \begin{pmatrix} -\sum_{i=2}^{m} A_{i1} & A_{12} & A_{13} & \cdots & A_{1m} \\ A_{21} & -\sum_{\substack{i=1 \\ i\neq 2}}^{m} A_{i2} & A_{23} & \cdots & A_{2m} \\ A_{31} & A_{32} & -\sum_{\substack{i=1 \\ i\neq 3}}^{m} A_{i3} & \cdots & A_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & A_{m3} & \cdots & -\sum_{i=1}^{m-1} A_{im} \end{pmatrix} \tag{2.37}$$

Taking the derivative of **A** with respect to, for instance, A_{13} yields

$$\frac{\partial \mathbf{A}}{\partial A_{13}} = \begin{pmatrix} 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \tag{2.38}$$

with this we define matrix $\bar{\xi}_{i,j}$, which has the number 1 for the off-diagonal element i,j, number -1 at the diagonal element j,j and zero elsewhere. The derivative of $\frac{\partial \mathbf{M}_n}{\partial A_{i,j}}$ can then be calculated numerically with

$$\frac{\partial \mathbf{M}_n}{\partial A_{i,j}} = \lim_{h\to 0} \frac{1}{h} \left(e^{(\mathbf{A}_n + h\bar{\xi}_{i,j})t} - e^{\mathbf{A}_n t} \right). \tag{2.39}$$

Terms $\frac{\partial A_{r,s}}{\partial \bar{\sigma}}$ and $\frac{\partial A_{r,s}}{\partial \mathbf{N}_n}$ are more difficult to evaluate. A perturbation in σ impacts the elements of matrix **A** both directly and indirectly. The direct effect encompasses the immediate change in the microscopic reaction rate from the perturbed cross section. A perturbation in σ or **N** perturbs the neutron flux, which in turn affects every reaction rate in the system. This is the non-direct effect, which is much harder to evaluate, than the direct effect.

To make the distinction between the direct and indirect components more clear, we expand the $\frac{\partial A_{r,s}}{\partial \bar{\sigma}}$ term using the multigroup scalar flux ϕ_g and multigroup microscopic cross sections $\overline{\sigma_{g,r,s}}$ as so

$$\frac{\partial}{\partial \bar{\sigma}} A_{r,s} = \frac{\partial}{\partial \bar{\sigma}} \sum_g \phi_g(\mathbf{r}) \overline{\sigma_{g,r,s}} \tag{2.40}$$

$$= \sum_g \frac{\partial \phi_g(\mathbf{r})}{\partial \bar{\sigma}} \overline{\sigma_{g,r,s}} + \sum_g \phi_g(\mathbf{r}) \frac{\partial \overline{\sigma_{g,r,s}}}{\partial \bar{\sigma}} \tag{2.41}$$

$$= \sum_g \frac{\partial \phi_g(\mathbf{r})}{\partial \bar{\sigma}} \overline{\sigma_{g,r,s}} + \bar{\phi}(\mathbf{r}). \tag{2.42}$$

The first term in 2.42 represents the indirect effect on $A_{r,s}$ from the perturbed flux. The second term is the direct effect, which is equal to the multi-group flux vector $\bar{\phi}$. If we assume the $\frac{\partial \phi_g(\mathbf{r})}{\partial \bar{\sigma}}$ terms small and ignore the indirect effect, we can replace the $\frac{\partial A_{r,s}}{\partial \bar{\sigma}}$ term with the multi-group flux vector $\bar{\phi}$ in equation 2.36.

The $\frac{\partial A_{r,s}}{\partial \mathbf{N}}$ term contains only the indirect component, because $A_{r,s}$ does not explicitly depend on **N**. We apply the same crude approximation, and set the $\frac{\partial A_{r,s}}{\partial \mathbf{N}}$ term to zero. This approximation is bound to fail at higher burnups, when $\nabla_\sigma \mathbf{N}$ differs from zero significantly. However, for the first couple burnup steps this approximation should yield relatively accurate predictions, due to $\nabla_\sigma \mathbf{N}$ being approximately zero. It is worth mentioning here that this sort of approach is definitely not rigorous, and methods for evaluating the indirect effects should be developed in the future.

Substituting the multi-group flux vector into the square brackets in the equation 2.36 yields the following final expression for the uncertainty propagation formula

$$\nabla_\sigma \mathbf{N}_{n+1} = \mathbf{M}_n \nabla_\sigma \mathbf{N}_n + \sum_{r \neq s}^{m \times m} \frac{\partial \mathbf{M}_n}{\partial A_{r,s}} \mathbf{N}_n \bar{\phi}. \tag{2.43}$$

The nuclide density derivatives $\nabla_\sigma \mathbf{N}$ are propagated through the burnup calculation with equation 2.43. The covariance of **N** can then be collapsed with equation 2.44.

$$Cov(\mathbf{N}) = \nabla_\sigma \mathbf{N}_n Cov(\bar{\sigma}) \nabla_\sigma \mathbf{N}_n^T \tag{2.44}$$

The workflow of the developed uncertainty propagation framework is summarized in figure 1.

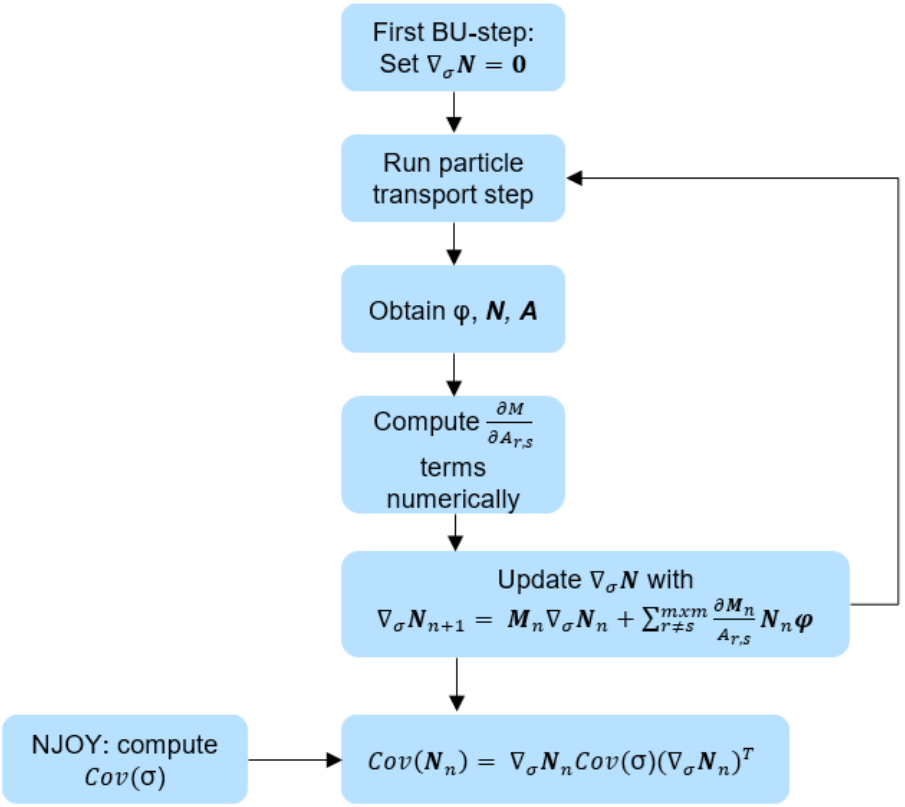


Figure 1. Uncertainty propagation progression

The framework can be used to derive nuclide density uncertainties as a function of burnup. This can be done by applying the equation 2.44 at every burnup point to produce a covariance matrix for every burnup point. The burnup specific nuclide density uncertainties are then acquired by extracting the variance information from the diagonal of the matrices. The uncertainty (absolute standard deviation) is given as

the square root of the variance. With this approach the uncertainties for a given nuclide are obtained as a function of burnup. The burnup specific uncertainties in section 4 have been derived using this approach.

The $\frac{\partial A_{r,s}}{\partial \mathbf{N}_n}$ and $\frac{\partial A_{r,s}}{\partial \sigma}$ terms can be evaluated with collision history based sensitivity calculations [6]. These capabilities are already implemented in Serpent and could in principle be used to make the developed uncertainty propagation framework more robust. The integration of these two frameworks was attempted in this research work. However, the obtained results were inconsistent with the reference TMC results, and for that reason they have been omitted from this report. A key priority in the future development of the uncertainty propagation method is the incorporation of the sensitivity calculations to the developed framework.

The framework was derived for a one-depletion-zone system, which is typically not applicable to real world scenarios. With multiple depletion zone models, the progression of the algorithm would have to solve the next iteration of $\nabla \mathbf{N}_{n+1}$ for each depletion zone individually. The difficulty with multiple depletion zones arises from the fact that the depletion is zone A is somewhat dependent on the depletion of zone B. This effect is amplified in burnable absorber rods, where the pin is typically divided into multiple ring-shaped depletion zones. Since these zones are adjacent to one another, their depletion may be strongly correlated for some nuclides. The Serpent's sensitivity calculation capabilities can compute these cross depletion zone terms, but since the number of the terms scales as n^2 where n is the number of depletion zones, it may be infeasible to calculate all of them for larger systems.

3. Modelled system

The developed theory was put into test with a simple one-depletion-zone Serpent model, consisting of one fuel pin and surrounding moderator. Figure 2 visualizes the geometry of the system.

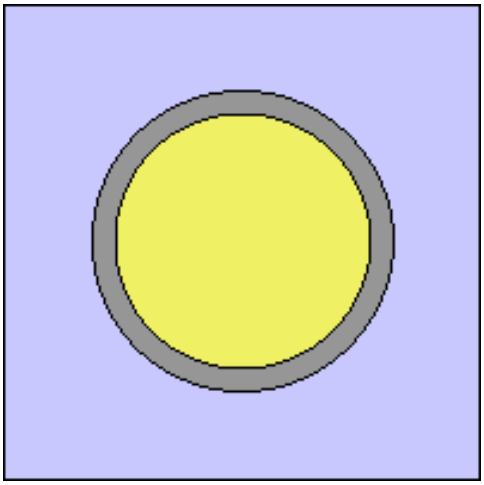


Figure 2. Simulated one-depletion-zone Serpent model.

To simplify the problem, only the uncertainty of the U238(n,g)U239 cross section uncertainty was considered in the calculation scheme. The reference solution for the uncertainty propagation of U238(n,g)U239 was solved with the Total Monte-Carlo approach, utilizing perturbed U238 files. Two hundred of these files were created from the evaluated JEFF3.3 [4] U238 file, using the SANDY [2] program. SANDY operates by applying perturbations to nuclear parameters from evaluated covariances. The perturbed U238 files only

had variations in the neutron capture reaction channel. Other nuclear data within the perturbed U238 files were kept unchanged. For nuclides other than U238, official processed JEFF3.3 files were used, together with JEFF3.1.1 for fission yield and decay data.

The developed uncertainty propagation method requires a multi-group covariance matrix to collapse the total uncertainty. This multigroup covariance matrix for U238(n,g)U239 was produced with the NJOY program [3].

The modelled fuel rod was burned through 42 burnup steps, ranging from zero to approximately 40 MWd/kgU. The depletion step was calculated with constant extrapolation, aka. the "Euler's method". This method was chosen for its simplicity. Higher-order depletion schemes would have introduced unwanted complications to the developed uncertainty propagation method.

4. Results

Figure 3 - 8 show the computed uncertainty estimates for nuclides U238, U239, Pu239, Pu240, Pu241 and Am241. The chosen unit for the uncertainty is the absolute standard deviation for the nuclide densities (bN/cm^3).

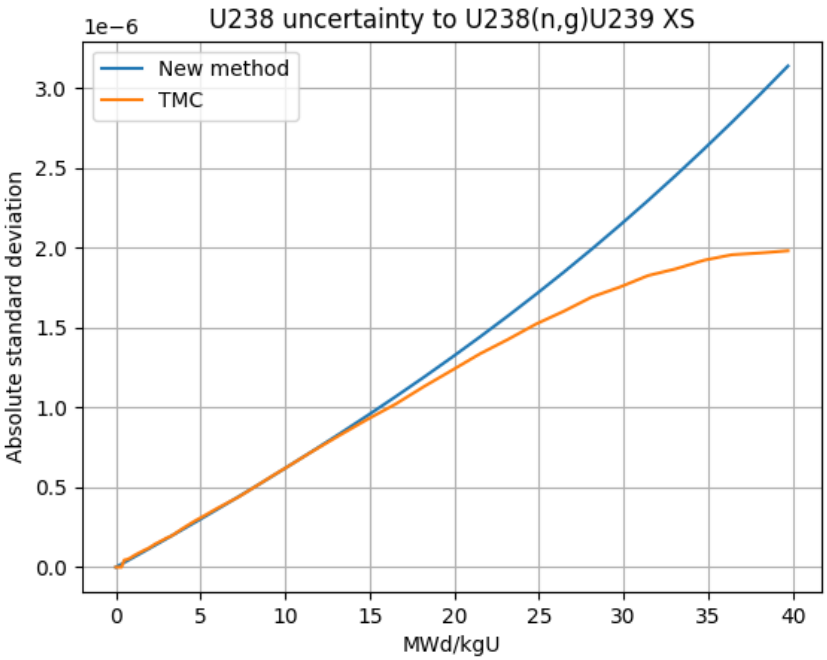


Figure 3. Absolute standard deviation - U238

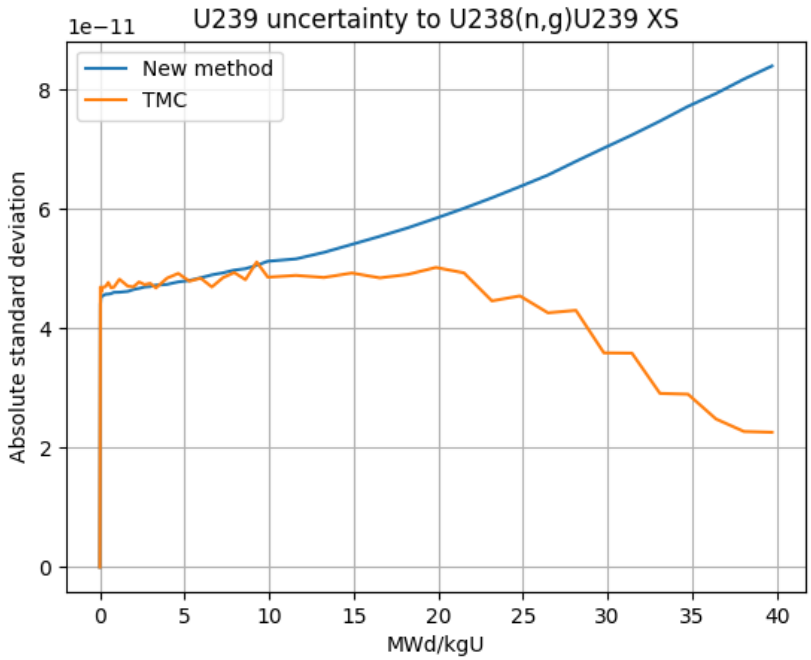


Figure 4. Absolute standard deviation - U239

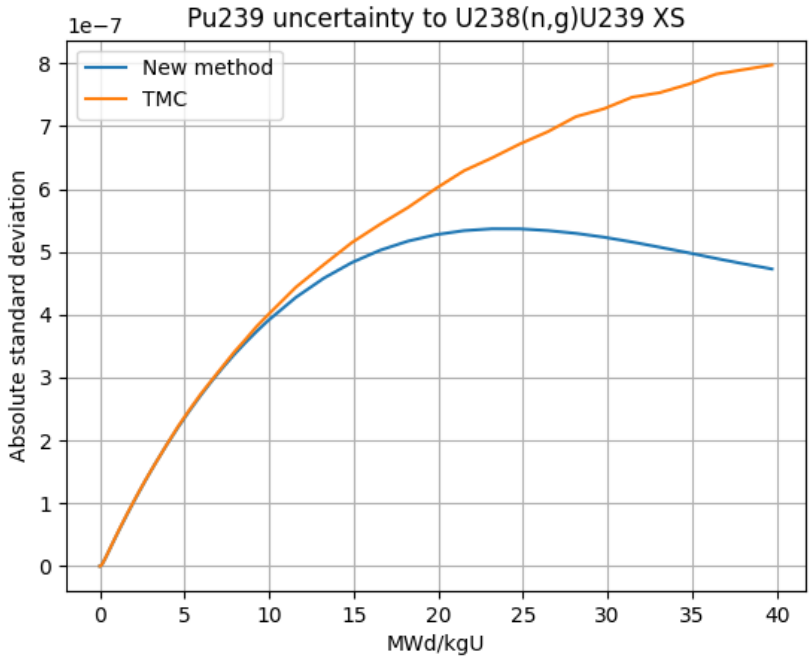


Figure 5. Absolute standard deviation - Pu239

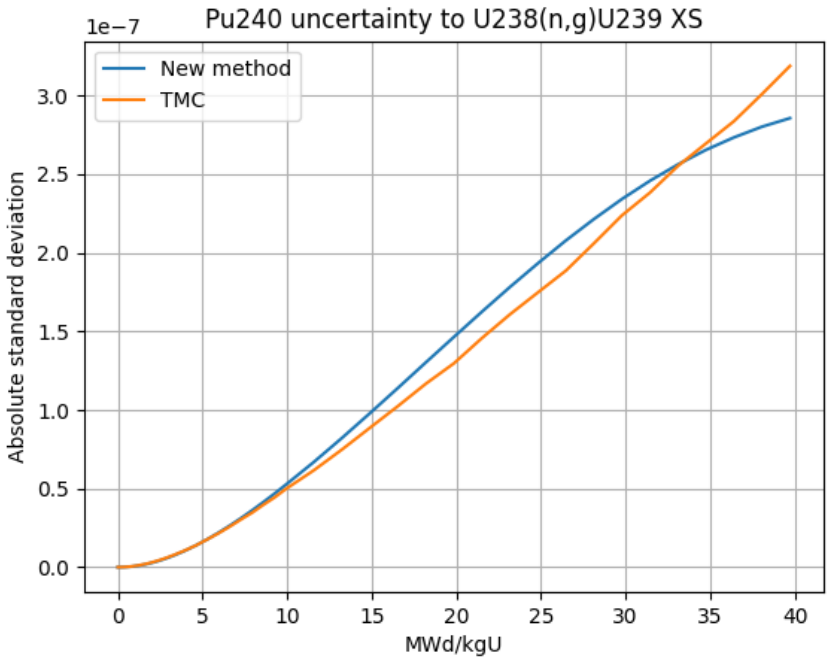


Figure 6. Absolute standard deviation - Pu240

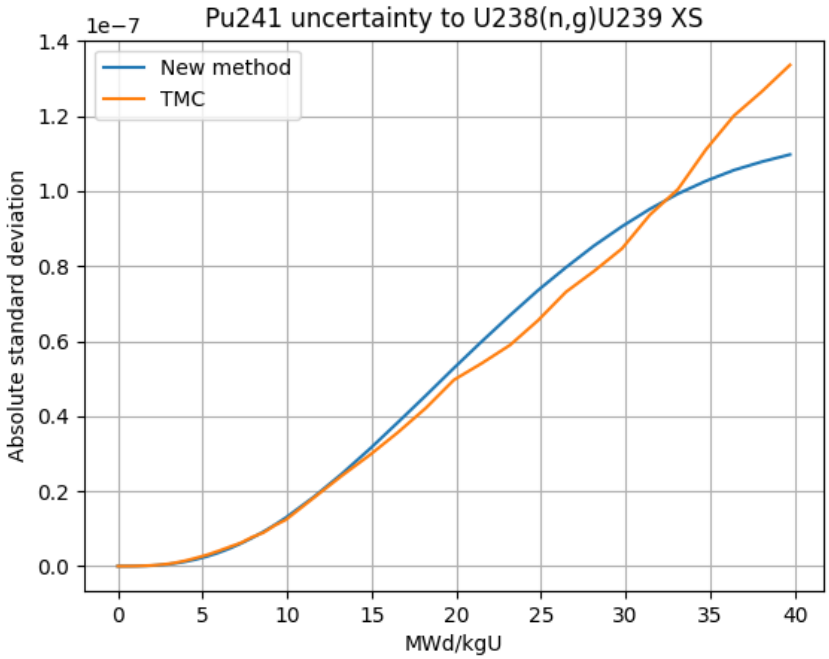


Figure 7. Absolute standard deviation - Pu241

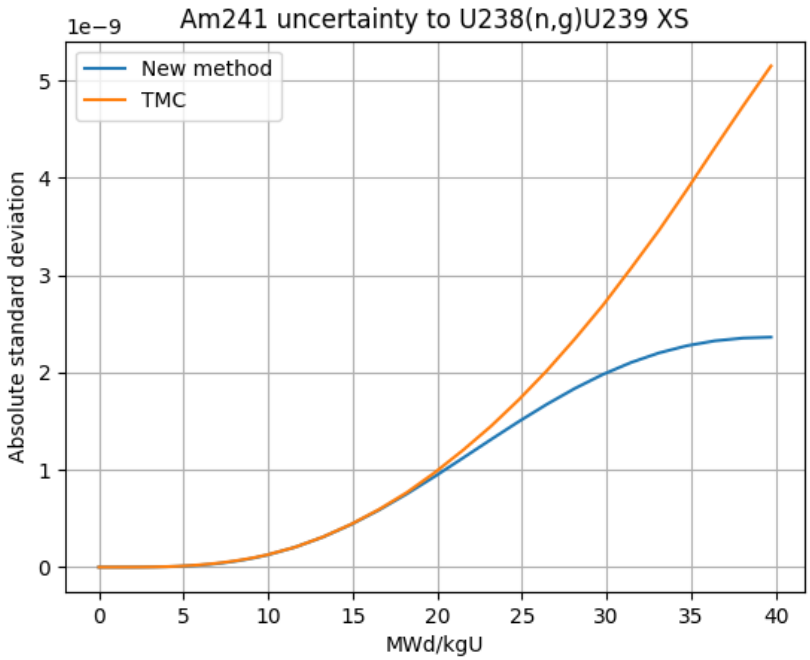


Figure 8. Absolute standard deviation - Am241

5. Conclusions

This research work was aimed at developing a new cross section uncertainty propagation framework for Monte-Carlo particle transport code Serpent 2.2. The developement was motivated by the the computational burden of the Total Monte-Carlo method. The developed method is based on the computation of the Jacobian matrix for nuclide densities with respect to discretized cross section data. It was shown that this matrix can be used to collapse the uncertainty information into a covariance matrix of the nuclide density vector, if covariance matrices for the cross section data are available. The importance of the covariance of the nuclide densities was demonstrated with the derivation of the time-evolution for the decay heat uncertainty. The time-evolution of this uncertainty is straight-forwardly calculated, if the covariance of the nuclide densities is available.

The developed method was evaluated with a simple two-dimensional one-depletion-zone Serpent model. The reference values for the nuclide density uncertainties were obtained with the established Total Monte-Carlo approach. The uncertainty propagation was confined to one nuclear reaction, namely the neutron capture of U238. This reaction was chosen for its importance in reactor physics. The studied responses include the nuclide densities of U238, U239, Pu239, Pu240, Pu241 and Am241.

The developed method exhibits good performance up to approximately 15 MWd/kgU, after which it starts to gradually drift away from the reference TMC solution for some nuclides. This behavior was somewhat expected, since the flux variations from the variations in nuclide densities were ignored in the developed framework. The uncertainty of Pu240 and Pu241 densities were predicted to the highest accuracy, with relatively good agreement to the reference solution all the way up to 40 MWd/kgU.

All in all, the developed method showed promising results, but plenty of development work is required before the method can be put into practical use. The scalability of the method remains an open question. The uncertainties were calculated very fast for the one-depletion-zone model with only one perturbed reaction channel. There is no guarantee that this efficiency remains when more depletion zones are introduced.

Moreover, methods for the evaluation of the indirect terms (flux perturbations) must be derived before the method can be used to predict uncertainties at higher burnups. Before these open questions are answered, a system-wide implementation of the developed uncertainty propagation method is unlikely to happen.

References

[1] Leppänen, J., et al. (2015) *The Serpent Monte Carlo code: Status, development and applications in 2013.* Ann. Nucl. Energy, 82 (2015) 142-150.

[2] L. Fiorito a b, G. Žerovnik c, A. Stankovskiy a, G. Van den Eynde a, P.E. Labeau b. *Nuclear data uncertainty propagation to integral responses using SANDY.* Annals of Nuclear Energy Volume 101, March 2017, Pages 359-366. 2017.

[3] MacFarlane, R. E., Muir, D. W., Boicourt, R. M., Kahler, A. C., Conlin, J. L., Haeck, W. *The NJOY Nuclear Data Processing System, Version 2016*. Los Alamos National Laboratory. Updated for NJOY2016.53 in 2019.

[4] Plompen, A. J. M. , et al., *The joint evaluated fission and fusion nuclear data library, JEFF-3.3*. European Physical Journal A, Volume 56, 181, July 2020, DOI:10.1140/epja/s10050-020-00141-9

[5] Vaara, L. O. T., *Nuclear Data Uncertainty Propagation with Total Monte-Carlo Method*. Master's thesis. Aalto University (2022). Available: https://urn.fi/URN:NBN:fi:aalto-202209045290

[6] Aufiero, M. et al. *A collision history-based approach to sensitivity/perturbation calculations in the continuous energy Monte Carlo code SERPENT*, Ann. Nucl. Energy, 152 (2015) 245-258.