Annals of Nuclear Energy 211 (2025) 110892

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

Uncertainty quantification of spent nuclear fuel with multifidelity Monte Carlo

Arnau Albà [a,b], Andreas Adelmann [a], Dimitri Rochman [a,*]

[a] Paul Scherrer Institut, Forschungstrasse 111, 5232 Villigen, Switzerland
[b] ETH Zürich, Rämistrasse 101, 8092 Zurich, Switzerland

ARTICLE INFO

Keywords:
Spent nuclear fuel
Uncertainty quantification
Decay heat
Monte Carlo
Multifidelity Monte Carlo
Machine learning

ABSTRACT

Uncertainty quantification (UQ) of spent nuclear fuel (SNF) is a crucial task that provides predictions and confidence bounds for important quantities of interest, such as decay heat, nuclide content, or k-effective. An accurate estimation of these quantities and their uncertainties is essential to reduce the risks and costs of storing and transporting SNF. The most accurate and robust method employed in reactor physics computations for UQ is Monte Carlo, however it is computationally intensive, to the point where it is unfeasible to carry out UQ for all the high-level waste expected in Switzerland. In this work, Multifidelity Monte Carlo (MFMC) is applied for the first time to the UQ of SNF, and a novel model management strategy to alleviate the training costs of MFMC is introduced. It is shown that MFMC drastically reduces the computational costs of UQ, with speedups between 5 and 1500 with respect to simple Monte Carlo.

1. Introduction

Calculations with reactor neutronics codes have become an essential tool in all phases of the nuclear fuel cycle, and critical decisions are taken based on the results of these calculations. In the case of spent nuclear fuel (SNF), nuclear codes predict quantities of interest a few thousand years into the future, such as nuclide inventory, decay heat (DH), or the canister neutron multiplication factor (k-eff). These predictions are used to determine for instance the type and amount of shielding required, or the number of canisters needed, among other parameters. These decisions ensure that the risks and costs are kept at a minimum during storage, transport, and disposal of SNF. It is therefore of utmost importance that nuclear codes are accurate, and that their uncertainties are well understood. Note that the current work is restricted to neutronic calculations (transport and depletion).

Uncertainty in nuclear calculations occurs due to uncertainties in the simulation inputs, that propagate to output quantities. This effect is modelled by assuming that the uncertain inputs follow a known probability distribution, and the aim of uncertainty quantification (UQ) is to estimate the probability distribution of the output quantities (see Fig. 1). Inputs that have been identified (Vasiliev et al., 2018) as being the major sources of uncertainty in SNF are the design and operational (DO) parameters (e.g. fuel temperature, material density, burnup), and nuclear data (ND) (e.g. energy dependent cross sections, fission spectra). In the nuclear community, two methods have been widely adopted for UQ: linear error perturbation (Cacuci, 2010) and Monte Carlo (MC) methods (Koning and Rochman, 2008; Rochman et al., 2014). The former method is a simple procedure requiring few calculations, but it is heavily biased as it assumes a linear relation between input and output quantities. On the other hand, MC methods are robust and make few assumptions on the simulated model, thus they are the preferred method when accurate UQ is necessary, especially if the output quantities of interest are expected to be non-Gaussian (Rochman et al., 2016). The drawback of MC methods is that, since they rely on repeating a simulation many times with stochastically perturbed input, they often require thousands of calculations, which can be computationally expensive and even prohibitive. The aim of the current paper was to investigate methods that are robust and unbiased, like MC, but with a lower computational cost.

The problem of expensive UQ calculations is well illustrated in the case of Switzerland's deep geological repository for radioactive waste (ENSI, 2020). For the planning of the future repository, calculations will be required to optimise and minimise quantities such as canister k-eff, DH, or dose emission, and these calculations should be accompanied by uncertainty margins (Vasiliev et al., 2019; Herrero et al., 2017a). Considering a 60 year lifespan of the country's nuclear power plants, there will be an expected number of 12 thousand SNF assemblies (Caruso and Pantelias Garcés, 2015), each requiring a UQ calculation. Therefore, assuming that each UQ calculation requires $N \simeq 1000$ simulations of several hours each, the computational costs

* Corresponding author.
E-mail addresses: arnau.albajacas@psi.ch (A. Albà), dimitri-alexandre.rochman@psi.ch (D. Rochman).

https://doi.org/10.1016/j.anucene.2024.110892
Received 18 June 2024; Received in revised form 26 August 2024; Accepted 28 August 2024
Available online 7 September 2024

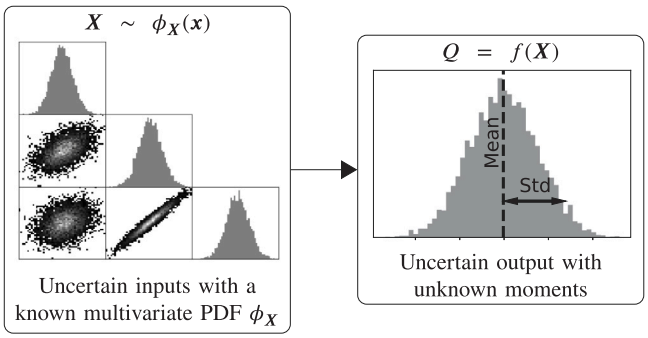

Fig. 1. A visual description of an uncertainty quantification problem. The input follows a known multivariate probability density function (PDF) Φ_X. The simulation model is f, and the output distribution is $Q = f(\boldsymbol{X})$. The objective of UQ is to accurately estimate the moments of Q, such as mean or standard deviation (Std).

of planning the deep geological repository quickly rise to hundreds of millions of CPU hours (Frankl et al., 2021). The problem is further exacerbated if one considers that the assemblies are loaded into canisters, and in principle the k-eff will need to be calculated for every possible canister loading combination (Solans et al., 2021). As a consequence, for a feasible planning of Switzerland's SNF disposal, investigating computationally cheap UQ methods is crucial.

In recent years machine learning (ML) techniques have emerged as a possible contender to replace traditional UQ methods. With this approach, a model $\widetilde{f}$ is trained by fitting to a training set of size N_{tr}, and then $\widetilde{f}$ can be used for UQ. The crux of this approach is that it is only computationally cheaper than simple MC if $N \gg N_{tr}$, i.e. if the training set is smaller than the number of simulations that would be required for simple MC. In Ebiwonjumi et al. (2020, 2021) and Albà et al. (2024), a variety of ML models were successfully used for UQ of spent fuel. These papers used training sets of size $N_{tr} \leq 500$, and were able to train accurate ML models. However, only DO input uncertainties were considered, which amounted to an input size of $d \leq 11$, while ND was not considered, which would have led to an input size $d \gg 1000$. This is a crucial point because ML models suffer from the well-known *curse of dimensionality* (Koppen, 2000), meaning that the size of the required training set grows exponentially with d, and at the very least one requires a training set $N_{tr} > d$. In the case $N_{tr} < d$ the trained model is heavily biased, and likely much less accurate than simple MC. Therefore, if one were to consider the input uncertainties of ND and DO, the training set would have to be much larger than 1000, rendering the ML approach costlier than simple MC. Some exceptions with large input dimensions can however be found in the literature. In Radaideh et al. (2021) a neural network was trained for UQ, considering ND as the source of uncertainty. In this case the ND parameters were limited to 336 (already an introduction of bias by considering only the principal reactions), and $N_{tr} = 1000$ training samples were considered. In Solans et al. (2021) a neural network was trained to predict the k-eff of a canister, where the input was the nuclide content of the SNF assemblies in the canister. Through some clever dimensionality reduction techniques, the number of inputs was reduced to $d = 3300$, and a training set of $N_{tr} = 46\,000$ criticality simulations was used, although in this case an advantage was still obtained since $\widetilde{f}$ was evaluated more than N_{tr} times. In general, however, it is not possible to train an unbiased ML model for UQ that includes ND and DO inputs, such that $N_{tr} < 1000$.

Multifidelity Monte Carlo (MFMC, similar to Multilevel Monte Carlo) (Giles, 2008; Peherstorfer et al., 2018) is a variation on simple MC that has been an active area of research in recent years. Like Monte Carlo, these methods rely on repeating calculations many times with stochastically perturbed inputs, and using the outputs to estimate the moments of the quantities of interest. However, unlike simple MC where only one model f is evaluated, MFMC evaluates multiple models of different accuracies and computational costs, and optimises how many times each model is called. As a result, MFMC estimations are unbiased, and require a lower overall computational cost than simple MC. Recent applications of MFMC for UQ can be found in several fields of research, such as climate modelling (Gruber et al., 2023), stellarator modelling for nuclear fusion (Law et al., 2022), or thermal battery analysis (Adams et al., 2022), although thus far it has not been applied to SNF (with the exception of Lasso Monte Carlo (LMC), discussed in the following paragraph).

In this work MFMC is applied for the first time to the UQ of spent nuclear fuel, and improvements in accuracy and reduction of computational costs are studied. The MFMC algorithm is presented with two variants, both of them employing two levels of fidelity: an expensive physics-based simulation, and a low-fidelity ML model, trained on few samples. An important contribution from this work is the model management strategy that is introduced. The strategy consists in selecting a family of highly correlated UQ problems, such as a family of similar SNF assemblies, and training one single ML model that serves as a low-fidelity approximation for all UQ problems. This novel strategy is the main difference with the LMC method developed in Albà et al. (2023b,a), where many ML models had to be trained.

The remainder of the paper is organised as follows. The theory Sections 2.1 and 2.2 describe simple MC and MFMC respectively, and discuss the expected theoretical advantages. Section 2.3 introduces the model management strategy for MFMC that is specifically aimed at neutronics UQ problems. Finally, Section 3 shows a thorough comparison of simple MC and MFMC when applied to different type of fuel assemblies and quantities of interest.

2. Theory

2.1. Simple Monte Carlo

Let a function $f : \mathbb{R}^d \mapsto \mathbb{R}$ be the *high-fidelity model*, which takes an input $\boldsymbol{x} \in \mathbb{R}^d$ and returns some quantity of interest (QoI) $q = f(\boldsymbol{x})$ as an output. In the context of this paper, f is a physics-based simulation such as a lattice code or a Monte Carlo transport code, the input vector $\boldsymbol{x} \in \mathbb{R}^d$ contains the inputs of the simulations such as reactor design parameters and nuclear data, and the output $q = f(\boldsymbol{x})$ might be the decay heat (DH), the neutron multiplication factor (k-eff), or a nuclide concentration.

Assume now that the input is a continuous multivariate random variable $\boldsymbol{X}$ that represents the uncertainty in the input. In the case of nuclear physics, the probability density function (PDF) of $\boldsymbol{X}$ generally consists of a combination of normal and lognormal distributions, along with the appropriate covariance matrix such that physical constraints are satisfied. Since the input of f is a random variable, the output $Q = f(\boldsymbol{X})$ is also a random variable, and the goal of UQ is to estimate certain properties of the output distribution. This work specifically considers the estimation of the first three central moments, namely the mean $\mu_1[Q] = \mathbb{E}[Q]$, variance $\mu_2[Q] = \mathbb{E}[(Q - \mathbb{E}[Q])^2]$, and third moment $\mu_3[Q] = \mathbb{E}[(Q - \mathbb{E}[Q])^3]$, although the presented methods can in principle be used for computing any higher-order moments. From the first three central moments, one is usually also interested in calculating the standard deviation $\sigma = \sqrt{\mu_2}$ and skewness $\gamma = \frac{1}{\sqrt{\mu_2}^3}\mu_3$. A visual description of a UQ problem can be found in Fig. 1.

A standard brute-force method for estimating the output moments is Monte Carlo uncertainty propagation (Koning and Rochman, 2008; Rochman et al., 2016) (from hereon referred to as *simple MC*), which relies on computing a large number of output samples to approximate

the distribution. The simple MC algorithm consists of the following steps:

Algorithm 1 Simple Monte Carlo for estimating the i-th moment $\mu_i[Q]$

Require: Input distribution $\boldsymbol{X}$ with known PDF, an initial integer N, a desired error ε

1: **while** $\sqrt{\text{MSE}\left[h_i\right]} > \varepsilon$ **do**
2: Sample N input vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ from i.i.d. random variables distributed as $\boldsymbol{X}$
3: Compute the set of outputs $\{q\}_N = q_1, ..., q_N = f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_N)$
4: Compute the estimator $h_i\left[\{q\}_N\right]$ (eq. (2))
5: Increase N by ΔN
6: **end while**

where, in steps 2 and 3, it is understood that the samples from the previous iteration can be reused, and only the ΔN new samples need to be sampled and evaluated at each iteration. The most expensive step of the algorithm is step 3, the N evaluations of f, while the rest of the steps have negligible costs. Therefore the method is said to have a cost N, and N depends on how fast the mean squared error (MSE) of the estimator converges. The MSE of an estimator is defined as

$$\text{MSE}\left(h_i\left[\{Q\}_N\right]\right) = \mathbb{E}\left[\left(h_i\left[\{Q\}_N\right] - \mu_i[Q]\right)^2\right] \quad \forall\, i \in \mathbb{Z}^+, \tag{1}$$

where $\mathbb{Z}^+$ are all the positive integers. The estimators employed in the algorithm are the *h-estimators*, which are the unbiased estimators with the smallest variance (Halmos, 1946). For the first three moments they are given by

$$\begin{aligned} h_1\left[\{q\}_N\right] &= \frac{1}{N}\sum_{i=1}^{N} q_i\,, \\ h_2\left[\{q\}_N\right] &= \frac{1}{N-1}\sum_{i=1}^{N}\left(q_i - h_1\right)^2\,, \\ h_3\left[\{q\}_N\right] &= \frac{N}{(N-2)(N-1)}\sum_{i=1}^{N}\left(q_i - h_1\right)^3\,, \end{aligned} \tag{2}$$

and they have an error of order $\sqrt{\text{MSE}} \sim \frac{1}{\sqrt{N}}$. Appendix A discusses in more detail the convergence of these estimators and how to calculate the MSE.

The advantages of simple MC are that it is unbiased, non-intrusive, simple to implement, and has a dimension-independent convergence (i.e. MSE does not depend on d). The main disadvantage of the method is that the convergence rate of $1/\sqrt{N}$ is generally regarded as being slow, since a large number of simulations N is required to reach a small enough error. For example, assuming that Q is normally distributed, a relative error of 2% for the $\sqrt{h_2}$ estimator would require $N = 1250$. From the SNF UQ literature one finds that in Frankl et al. (2021) $N = 200$ was used which gives an error of 5%, and in Shama et al. (2021) $N = 625$ was used which gives an error of 3%. Since f is usually a computationally expensive model, it is undesirable (sometimes impossible) to evaluate it many times.

2.2. Multifidelity Monte Carlo

Multifidelity Monte Carlo (MFMC) (Peherstorfer et al., 2018) is a method of combining evaluations of multiple models to calculate unbiased estimates of the moments $\mu_i[Q]$, at a reduced computational cost compared to simple MC. In this work MFMC is implemented with two levels of fidelity: the same *high-fidelity* model f considered in the previous section, and a computationally cheap *low-fidelity* approximation $\widetilde{f}$.

The low-fidelity model is a function $\widetilde{f} : \mathbb{R}^d \mapsto \mathbb{R}$, which takes the same inputs as f, and returns an output $\widetilde{q}$. When the input is replaced by a random variable $\boldsymbol{X}$, the low-fidelity output is also a random variable $\widetilde{Q} = \widetilde{f}(\boldsymbol{X})$. This model is a computationally cheap approximation of f, an example of which might be a simulation with coarse grids or large timesteps, a reduced order model, or a data-fit machine learning model. In the remainder of the paper the cost of evaluating $\widetilde{f}$ is considered negligible with respect to the cost of evaluating f. This assumption is written as follows:

Assumption 2.1. Let f be a high-fidelity model with a computational cost of ω, and $\widetilde{f}$ a low-fidelity model with cost $\widetilde{\omega}$. It is assumed that

$$\left|\frac{\widetilde{\omega}}{\omega}\right| \ll 1.$$

Given these two models, MFMC estimates the moments of $Q = f(\boldsymbol{X})$ with the following algorithm:

Algorithm 2 Multifidelity Monte Carlo for estimating the i-th moment $\mu_i[Q]$

Require: Input distribution $\boldsymbol{X}$ with known PDF, integers N and M, with $N \ll M$, a desired error ε

Require: Low-fidelity model $\widetilde{f}$ following assumption 2.1

1: Sample M input vectors $\boldsymbol{z}_1, ..., \boldsymbol{z}_M$ from i.i.d. random variables distributed as $\boldsymbol{X}$
2: Compute M low-fidelity outputs $\{\widetilde{p}\}_M = \widetilde{p}_1, ..., \widetilde{p}_M = \widetilde{f}(\boldsymbol{z}_1), ..., \widetilde{f}(\boldsymbol{z}_M)$
3: **while** $\sqrt{\text{MSE}\left[h_{\text{MF},i}\right]} > \varepsilon$ **do**
4: Sample N input vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ from i.i.d. random variables distributed as $\boldsymbol{X}$
5: Compute N high-fidelity outputs $\{q\}_N = q_1, ..., q_N = f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_N)$
6: Compute N low-fidelity outputs $\{\widetilde{q}\}_N = \widetilde{q}_1, ..., \widetilde{q}_N = \widetilde{f}(\boldsymbol{x}_1), ..., \widetilde{f}(\boldsymbol{x}_N)$
7: Compute the multifidelity estimator $h_{\text{MF},i}\left[\{q\}_N, \{\widetilde{q}\}_N, \{\widetilde{p}\}_M\right]$ (eq. (3) or eq. (4))
8: Increase N by ΔN
9: **end while**

where it is assumed that $N \ll M$ remains true throughout the execution of the algorithm. There are two possibilities for the multifidelity estimators, the MFMC estimators

$$h_{\text{MF},i}\left[\{q\}_N, \{\widetilde{q}\}_N, \{\widetilde{p}\}_M\right] = h_i\left[\{q\}_N\right] - h_i\left[\{\widetilde{q}\}_N\right] + h_i\left[\{\widetilde{p}\}_M\right] \quad \forall\, i \in \mathbb{Z}^+, \tag{3}$$

and the α-MFMC estimators

$$\begin{aligned} &h_{\alpha\text{-MF},i}\left[\{q\}_N, \{\widetilde{q}\}_N, \{\widetilde{p}\}_M\right] = h_i\left[\{q\}_N\right] - h_i\left[\alpha_N \cdot \{\widetilde{q}\}_N\right] \\ &+ h_i\left[\alpha_N \cdot \{\widetilde{p}\}_M\right] \quad \forall\, i \in \mathbb{Z}^+, \\ &\text{with} \quad \alpha_N = \frac{1}{h_2\left[\{\widetilde{q}\}_N\right]}\frac{1}{N-1}\sum_{j=1}^{N}\left(q_j - h_1\left[\{q\}_N\right]\right)\left(\widetilde{q}_j - h_1\left[\{\widetilde{q}\}_N\right]\right). \end{aligned} \tag{4}$$

This formulation of multifidelity estimators is based on the *multilevel Monte Carlo* method presented in Krumscheid et al. (2020). The α-MFMC formulation (Eq. (4)) adds a *control coefficient* α_N to the expression as is done in the multifidelity literature (Peherstorfer et al., 2016, 2018). The coefficient α_N is in fact a sample estimation of $\frac{\text{Cov}[Q,\widetilde{Q}]}{\mu_2[\widetilde{Q}]}$. These estimators have a convergence rate of $\sqrt{\text{MSE}} \sim \frac{1}{\sqrt{N}}$, the same as simple MC, however the additional benefit of MFMC is that the error can be controlled by the choice of $\widetilde{f}$. As a general heuristic rule, the closer $\widetilde{f}$ is to f, the smaller the MSE will become.

Note that although alg. 2 has more steps than simple MC (alg. 1), the expensive step (step 5) of the algorithm is the same: evaluating the high-fidelity model N times. The remaining part of the algorithm has negligible costs in comparison (Assumption 2.1). Therefore alg. 2 has a computational cost of N, and N depends on how fast the MSE of the estimators converges.

The speedup from using multifidelity Monte Carlo instead of simple MC can be calculated as the ratio of MSEs, with the following formula:

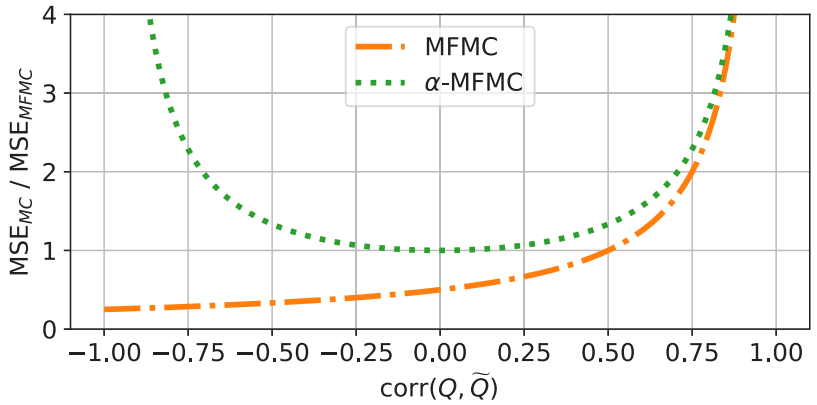

Fig. 2. Approximate speedup from using multifidelity methods instead of simple MC, calculated with Eq. (6).

$$\text{speedup MFMC for } i\text{th moment} \simeq \frac{\text{MSE}\left(h_i\left[\{Q\}_N\right]\right)}{\text{MSE}\left(h_{\text{MF},i}\left[\{Q\}_N, \{\widetilde{Q}\}_N, \{\widetilde{P}\}_M\right]\right)},$$
$$\text{speedup } \alpha\text{-MFMC for } i\text{th moment} \simeq \frac{\text{MSE}\left(h_i\left[\{Q\}_N\right]\right)}{\text{MSE}\left(h_{\alpha\text{-MF},i}\left[\{Q\}_N, \{\widetilde{Q}\}_N, \{\widetilde{P}\}_M\right]\right)}, \tag{5}$$

Note that the speedup is independent of the desired error ε, and of N, since the ratio of MSEs is approximately independent of N (see more details in Appendix A). The speedup depends on the choice of low-fidelity model $\widetilde{f}$, and on the order i of the estimated moment. A Python library (Albà, 2024) provided with this work contains the necessary formulas for the estimators and for calculating their MSE, where the formulas are based on the work in Krumscheid et al. (2020) and Amela et al. (2019).

As an example, consider the speedup from calculating the first moment μ_1, which under certain approximations (Appendix A) can be written as

$$\frac{\text{MSE}\left[h_1\right]}{\text{MSE}\left[h_{\text{MF},1}\right]} \simeq \frac{1}{2}\frac{1}{1-\text{Corr}[Q,\widetilde{Q}]} \quad \text{and}$$
$$\frac{\text{MSE}\left[h_1\right]}{\text{MSE}\left[h_{\alpha\text{-MF},1}\right]} \simeq \frac{1}{1-\text{Corr}^2[Q,\widetilde{Q}]}. \tag{6}$$

From these error ratios (plotted in Fig. 2) it can be seen that the speedup is indeed independent of N, and only depends on the correlation between the high- and low-fidelity models. More specifically, MFMC has a smaller error than simple MC when $\text{Corr}[Q,\widetilde{Q}] > 0.5$, and α-MFMC is always equal or better than simple MC. For the other estimators (variance, skewness, ...) the expressions for the MSE are complicated by many dependencies on higher order moments and correlations, so it is not possible to write them in a simple expression as in Eq. (6). A full discussion on the MSE and the optimal choice of $\widetilde{f}$ is provided in the Appendix A.

2.3. Model management for nuclear computations

The MFMC (Eq. (3)) and α-MFMC (Eq. (4)) methods require a low-fidelity model that satisfies two conditions: it needs to be a good approximation of the high-fidelity model f, and it needs to be computationally cheap (Assumption 2.1). A clear choice is then to use a supervised machine learning model since, once trained, it can be executed extremely fast, in the order of milliseconds, whereas physics-based simulations generally require from a few minutes to several hours.

The drawback of machine learning models is that they require a large amount of input–output samples called the *training set*, and evaluating N_{tr} training samples may incur high computational costs. Consider a UQ problem with its associated high-fidelity model f. Then, to carry out algorithm 2 one would firstly need N_{tr} evaluations of f to train the low-fidelity model $\widetilde{f}$, and then N further evaluations of f for the algorithm itself. In this work the costs of training were reduced by reusing old simulations from similar problems to train the low-fidelity model. In neutronics computations it is often the case that studied systems come in *family* of models that share similarities, such as a family of fuel assemblies with similar burnup histories, or a group of disposal canisters with similar content. In such cases one can train a low-fidelity model $\widetilde{f}$ with simulations from one model, and it will serve as a *cheap approximation* for all of the models in the family.

To formally describe the *model management* strategy, consider the example of L spent fuel assemblies, that share similar design characteristics and burnup histories. Each assembly is simulated by a different input file, giving place to a family of models $f_1, \dots, f_L$, written as

$$f_k : \mathbb{R}^d \mapsto \mathbb{R}, \quad \text{for } k = 1, 2, \dots, L,$$
$$\boldsymbol{X} \mapsto Q_k = f_k(\boldsymbol{X}),$$

Although each output $Q_1, \dots, Q_L$ is different, since the assemblies are similar it is likely that the outputs are highly correlated. For the first assembly, since there is no low-fidelity model available, UQ is performed with simple MC and a large number of samples N_1. After the first UQ process, these N_1 samples are used as a training set for $\widetilde{f}$. Then, for every subsequent assembly in the family, MFMC and α-MFMC can be applied with alg. 2 and $\widetilde{f}$, at a much smaller computational cost $N_2, N_3, \dots \ll N_1$.

The full algorithm for a family of similar UQ problems is described in algorithm 3.

Algorithm 3 Model Management for Multifidelity Monte Carlo

Require: Input distribution $\boldsymbol{X}$, desired error ε
Require: A family of L UQ problems, with their high-fidelity models $f_1, f_2, ..., f_L$
1: Carry out simple MC (alg. 1) for f_1
2: With the N_1 input-output samples from the previous step, train a low-fidelity model $\widetilde{f}$
3: **for** $k = 2, ..., L$ **do**
4: Carry out MFMC and α-MFMC (alg. 2) for f_k
5: **end for**

3. Applications and results

3.1. Design, operational, and nuclear data uncertainties

In this first example UQ was carried out for three SNF assemblies, to characterise their decay heat at different points in time, and their

Table 1
Characteristics of the three studied SNF assemblies. The FA at the bottom, C20, was used for training the low-fidelity model with $N = 500$.

FA name	Burnup [MWd/kgU]	Fissile content [% of ^{235}U]	Total cooling time between cycles [days]	Nº of cycles
C12	36.4	3.1	583	4
F32	51.0	3.2	1059	6
C20	35.7	3.1	2068	4

Table 2
Design and operational (DO) variables perturbed in the input of CASMO5 simulations, with their corresponding standard deviation taken from the SFCOMPO evaluation guide (Steering Committee for Nuclear Energy, 2016). A full correlation (Corr = 1) was assumed between fuel temperature and power. Fuel radius was correlated to fuel density, such that the total fuel mass was kept constant.

Parameter	Standard deviation (1σ)
Cladding radius	33.33 μm
Fuel density	0.67%
Fuel radius	(see caption)
Enrichment	0.0167%
Power	1.67%
Water temperature	2 K
Fuel temperature	50 K
Boron content	10 ppm

nuclide content at end of life (EoL). The high-fidelity models were CASMO5 (Rhodes et al., 2006) lattice calculations, which can simulate the full in-core cycles of a fuel assembly from fresh fuel to EoL, and return the characteristics of the SNF assembly as an output.

The studied assemblies were three uranium oxide (UO_2) samples irradiated in the Swedish pressurised water reactor (PWR) Ringhals-2, which were part of the Clab report (Sturek et al., 2006). The CASMO5 input files were prepared based on the description of the design and irradiation history given in the report, following the same conventions used in previous studies (Shama et al., 2022; Albà et al., 2024), namely using cycle-averaged quantities for each burnup step (power, boron concentration, fuel temperature), and imposing reflecting boundary conditions and a quarter-symmetry assumption. Each CASMO5 simulation required in the order of 10 min to run. The three fuel assemblies (FAs) shared similar design (15×15 layout of UO_2 pins) and operational parameters, with their largest differences being in the burnup and the number of cooling days in-between cycles (see Table 1 and more details in Sturek et al. (2006)).

There were $d = 15\,565$ perturbed inputs given to CASMO5, which can be divided into two types: design and operational variables (DO), and nuclear data (ND). DO variables include the characteristics of the fresh fuel assembly, reactor design parameters, and operational history, which altogether amount to 8 inputs. These parameters were perturbed by following the procedure described in Shama et al. (2021), where the inputs are sampled from a multivariate normal distribution, with the standard deviations and correlations from the SFCOMPO evaluation guide (Steering Committee for Nuclear Energy, 2016). The DO uncertainty parameters are given in Table 2. The remaining 15 557 perturbed parameters were nuclear data values, which are tabulated physical quantities that appear in the neutron transport equation. In this work the perturbed ND were neutron elastic and inelastic cross-sections, neutron capture cross sections, fission cross section, fission yields, fission spectrum, and neutron multiplicity (nu-bar). The ND parameters were given for many nuclides and, for energy-dependent reactions, for each of the 19 discrete energy groups (0 to 20 MeV) used by CASMO5. The PSI in-house code Shark-X (Wieselquist et al., 2013; Aures et al., 2017; Leray et al., 2017) was used for ND random sampling, which uses a multivariate normal distribution and a covariances based on ENDF/B-VII.1 (Chadwick et al., 2011).

Two output QoIs were considered, namely the decay heat at 500 days of cooling, which is relevant for the interim storage and transport of the spent fuel, and the assembly-averaged concentration of ^{155}Gd, which was chosen due to its skewed output distribution, to demonstrate the applicability of MFMC to higher order moments. A comparison of these quantities can be found in Fig. 3, where it is clear that the three output distributions have different moments.

As described in algorithm 3, simple MC was firstly carried out for one SNF assembly, since initially there was no low-fidelity model available. In this case the first model f_1 (see alg. 3) was the assembly C20, and $N_1 = 500$ was used. These N_1 input–output samples of C20 were then used to train a Lasso machine learning model $\widetilde{f}$ (Tibshirani, 1996) (see full details of the training procedure in Appendix B). The training of the Lasso model took less than one minute on a single CPU core and, once trained, it had an execution time of $\widetilde{\omega} \sim 10^{-4}$ s. Considering that the CASMO5 simulations had a cost of $\omega \sim 10$ min $\sim 10^3$ s, the ratio of costs was $\frac{\widetilde{\omega}}{\omega} \sim 10^{-7}$, thus Assumption 2.1 was satisfied. Note that a different Lasso model was trained for each QoI, and each training required under one minute.

With the now available low-fidelity model $\widetilde{f}$, UQ with multifidelity Monte Carlo was performed for the two other assemblies C12 and F32 (corresponding to high-fidelity models f_2 and f_3 in alg. 3). The multifidelity methods and the calculations of the MSE were all performed with the accompanying Python library (Albà, 2024), and $M = 6000$ (see alg. 2, and a schematic description in Fig. 4). Fig. 5 shows the running average plots for C12 with all three algorithms. In this case MFMC and α-MFMC are clearly superior to simple MC, as the predictions converge towards the true value faster. For these plots the *ground truth* was computed with a simple MC calculation of 6000 samples.

Since the presented methods are stochastic, rather than plotting the running average of one seed as in Fig. 5, it is more relevant to look at the average performance of the methods. Accordingly, the three methods were applied 50 times with different seeds for the input sampling, and the average errors are plotted in Figs. 6 and 7. These plots confirm that all three methods have a convergence rate of $1/N$, and that on average α-MFMC is the most accurate method, followed by MFMC and simple MC in this order. This same behaviour is observed in all plots, although the degree of improvement between methods varies. The different improvement is simply due to the fact that the MSE depends on higher order moments and higher order correlations of f and $\widetilde{f}$, which differ among assemblies and QoIs.

For completeness, UQ was carried out for several other QoIs, for assemblies C12, F32. In this case the methods were applied 50 times with different seeds, for a fixed computational cost of $N = 10$, and only the mean and standard deviation were computed, as the skewness was zero for most quantities. The results (Fig. 8) show the same trends observed above, with α-MFMC consistently being the most accurate method. The MSE of each method, assembly, and QoI was calculated and used to compute the speedups with Eq. (5). As can be seen in Fig. 9 there is a large spread of speedups due to the different moments and correlations of each QoI. Table 3 shows that α-MFMC always has a speedup larger than 1, ranging from a factor of 6 to 1553, with the largest speedup coming from the prediction of ^{137}Cs content. Since α-MFMC is always the most accurate method, MFMC will no longer be investigated in the remainder of the paper, and only α-MFMC and simple MC will be compared.

3.2. Application to mixed oxide fuel

In the previous section the three studied assemblies had similar design, burnup history, and type of fuel, thus it was to be expected that their CASMO5 models f_1, f_2, f_3 would be highly correlated. To further investigate the applicability of α-MFMC, in the sequel eight different assemblies are considered, coming from different reactors, with different designs, and with two types of fuel, namely UO_2 and mixed oxide (MOX). A summary of the eight assemblies is shown on Table 4. All assemblies were simulated using CASMO5, and in this case only nuclear data was perturbed (unlike the previous Section 3.1 where

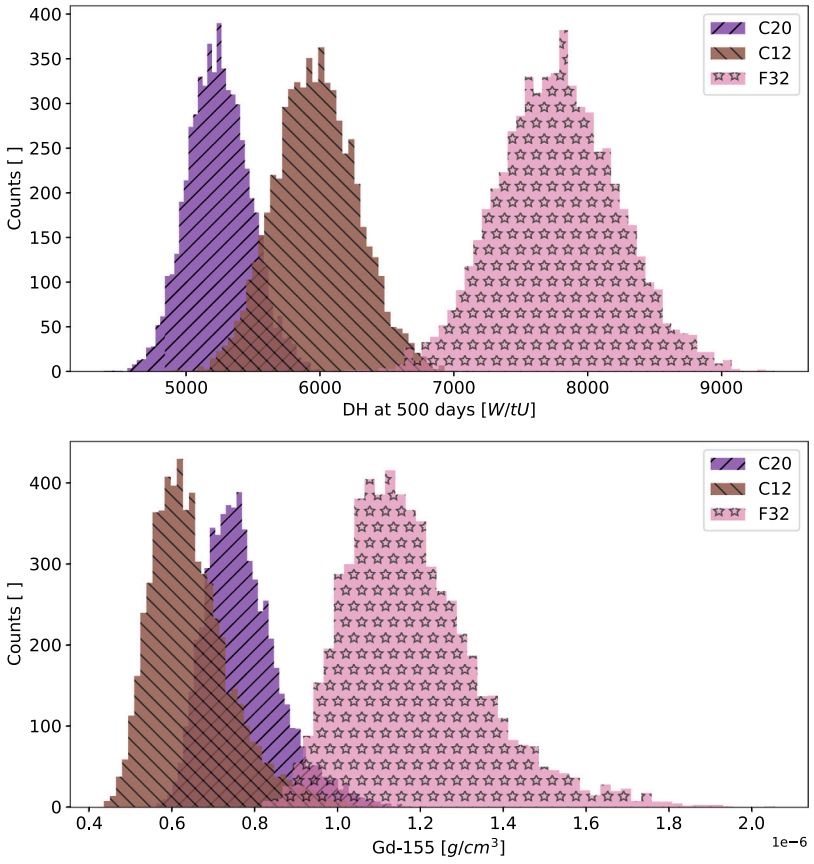

Fig. 3. Histograms of the quantities of interest studied for three different SNF assemblies. These plots were generated with 6000 CASMO5 simulations, each with randomly perturbed inputs (DO and ND). The outputs of the three assemblies are clearly different in regards to mean, variance, and skewness.

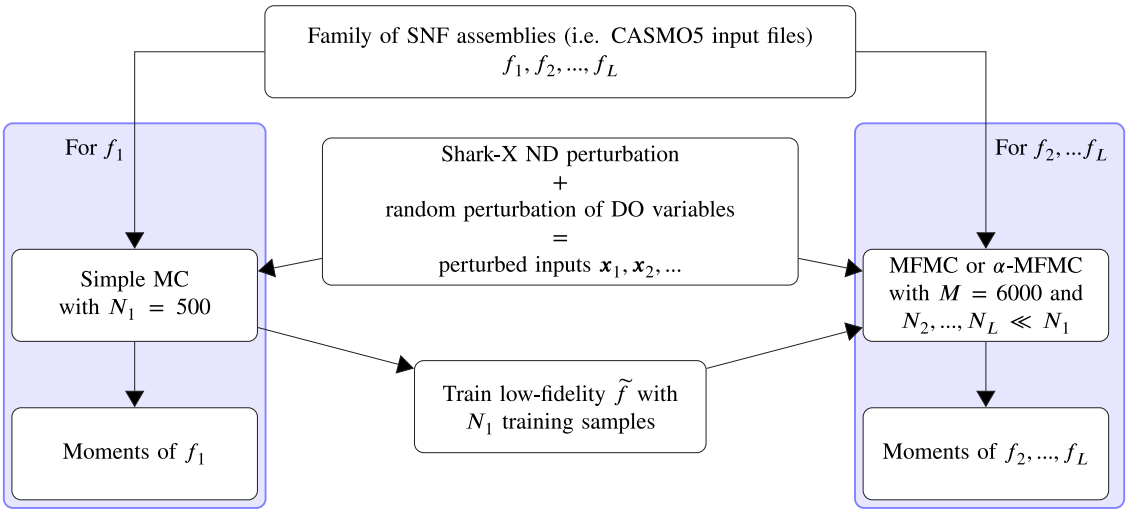

Fig. 4. Visual description of the model management strategy for applying MFMC or α-MFMC to a family of SNF assemblies.

ND and DO were perturbed), thus there were $d = 15\,557$ uncertain input parameters. The studied QoIs were the DH at 2 and 50 years of cooling, and the concentration of ^{239}Pu and ^{235}U at EoL.

Two low-fidelity models were trained $\widetilde{f}_{\text{C20}}$ and $\widetilde{f}_{\text{BM5}}$, for the UO_2 assembly C20 and the MOX assembly BM5 respectively. Each model was trained with $N = 500$ simulations and the Lasso method (as described in the Appendix B). The CASMO5 simulations of the UO_2 assemblies were relatively fast to run, averaging at 10 min each, since they used few burnup steps with cycle-averaged quantities. The simulations of MOX assemblies used many small and detailed burnup steps, thus requiring between 2 and 3 h per simulation. For the remaining six assemblies, UQ was carried out with $N = 10$, using three methods: simple MC, α-MFMC with $\widetilde{f}_{\text{C20}}$, and α-MFMC with $\widetilde{f}_{\text{BM5}}$. The results are plotted in Fig. 10.

Of the three methods, α-MFMC with $\widetilde{f}_{\text{C20}}$ is the most accurate one in all cases, except for the BM3 assembly. For BM3, the best method is α-MFMC with $\widetilde{f}_{\text{BM5}}$. This result suggests that UO_2 assemblies have highly correlated DH and nuclide content, despite their differences in burnup history and design, but they are weakly correlated to MOX

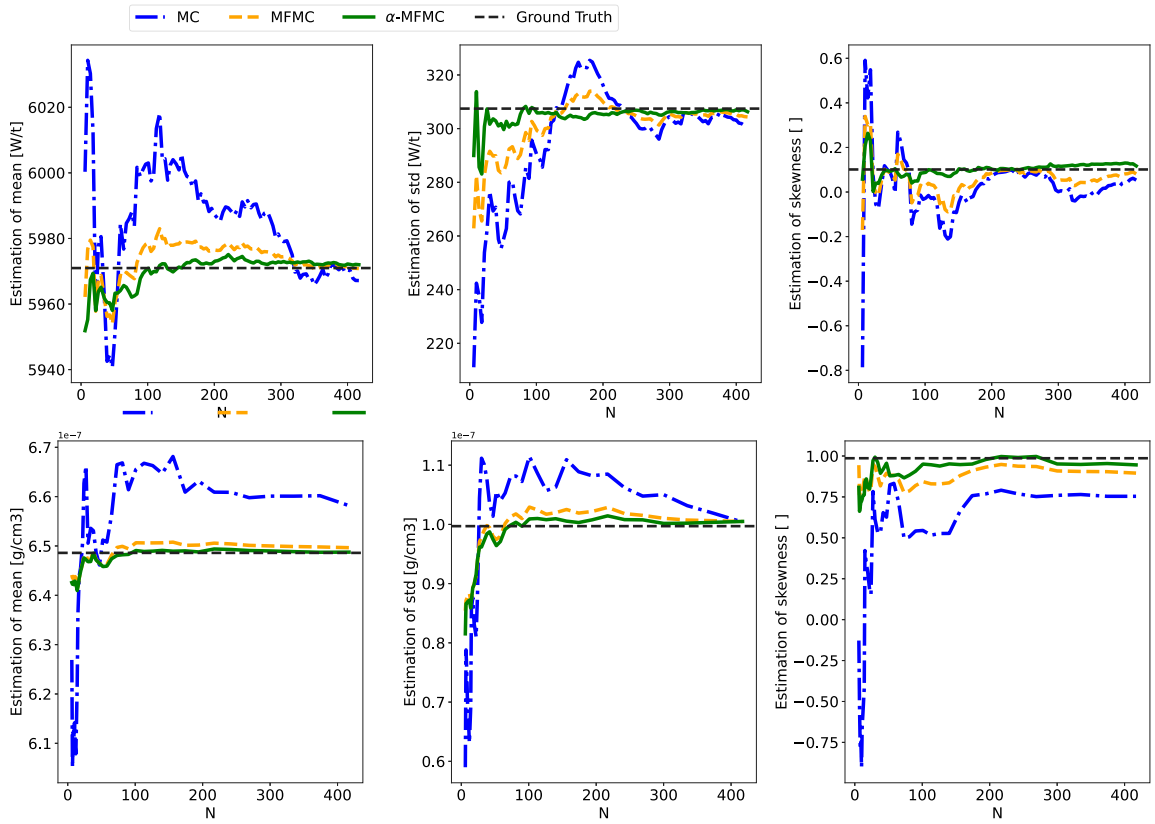

Fig. 5. Estimation of the mean, standard deviation, and skewness as the computational cost N increases. The estimated QoIs were the DH at 500 days of cooling (top plots), and the ^{155}Gd concentration at EoL (bottom plots), for the fuel assembly C12.

Table 3
Speedups from using multifidelity methods instead of simple MC. Speedup was calculated with Eq. (5), and the data from Fig. 8.

Method	MFMC		α-MFMC	
Estimated moment	Mean	Variance	Mean	Variance
Average speedup	56	29	172	52
Median speedup	23	6	48	19
Minimum speedup	2	1	16	6
Maximum speedup	245	629	1553	274
QoI with Min speedup	^{244}Cm, F32		DH 2 years, F32	
QoI with Max speedup	^{235}U, C12		^{137}Cs, C12	

Table 4
Characteristics of the studied SNF assemblies. The six UO_2 samples are described in detail in the Clab report (Sturek et al., 2006), and the two MOX ones (BM3 and BM5) are analysed in Rochman et al. (2021) and Primm (2002). The two bottom lines correspond to assemblies that were used for training models, hence they were simulated $N = 500$ times with perturbed input. The top six assemblies were only used for UQ with $N = 10$. In the table BWR stands for Boiling Water Reactor.

FA name	Reactor type	Fuel type	Burnup [MWd/kgU]	Pin layout	Fissile content [%]
1C5	PWR	UO_2	38.5	17 × 17	3.1 ^{235}U
C12	PWR	UO_2	36.4	15 × 15	3.1 ^{235}U
C42	PWR	UO_2	35.6	15 × 15	3.1 ^{235}U
2074	BWR	UO_2	22.9	8 × 8	2.33 ^{235}U
9329	BWR	UO_2	41.1	8 × 8	2.92 ^{235}U
BM3	PWR	MOX	43.7	14 × 14	3.8 Fissile Pu
C20	PWR	UO_2	35.7	15 × 15	3.1 ^{235}U
BM5	PWR	MOX	58.9	14 × 14	3.6 Fissile Pu

assemblies. This is to be expected due to their vastly different fissile content. As a consequence, UO_2 assemblies can be considered a *family* of UQ problems, that use the same low-fidelity model (see alg. 3). For a family of MOX assemblies, a new low-fidelity model should be trained.

The relation between correlation and speedup can be seen in Fig. 11, which shows the good agreement between the predicted theoretical speedup (Eq. (6)) and the true speedup using α-MFMC (calculated with Eq. (5)). Fig. 11 also shows that all UO_2 assemblies are highly correlated to $\widetilde{f}_{\text{C20}}$ and weakly correlated to $\widetilde{f}_{\text{BM5}}$, and the opposite behaviour for assembly BM3.

3.3. Criticality uncertainty at different cooling times

As a final example UQ was carried out for the neutron multiplication factor k-inf (k-inf when simulation uses reflecting boundaries, k-eff when open boundaries are imposed) in an SNF canister, at different cooling times. For the simulations, a canister containing four exactly equal SNF assemblies was modelled (as was done in Herrero et al. (2017b) for a preliminary study on the loading curves of SNF) with the Monte Carlo transport code OpenMC (Romano et al., 2015). The assembly in the canister was the C20 sample (Table 4), where the pin-wise nuclide content was extracted from CASMO5 simulations. The CASMO5 simulations to obtain the nuclide content were performed with randomly perturbed nuclear data (as in Section 3.1, although here DO variables were not perturbed). This setup gave place to a *family* of UQ problems (see alg. 3), described as follows:

$$f_T : \mathbb{R}^{15\,557} \mapsto \mathbb{R},$$
$$\text{ND} \mapsto \text{k-inf at time } T,$$

for $T = 0, 2, 9, \ldots, 5000$ years. Each model f_T was executed with the following steps:

1. perform a CASMO5 simulation of assembly C20, with perturbed ND,

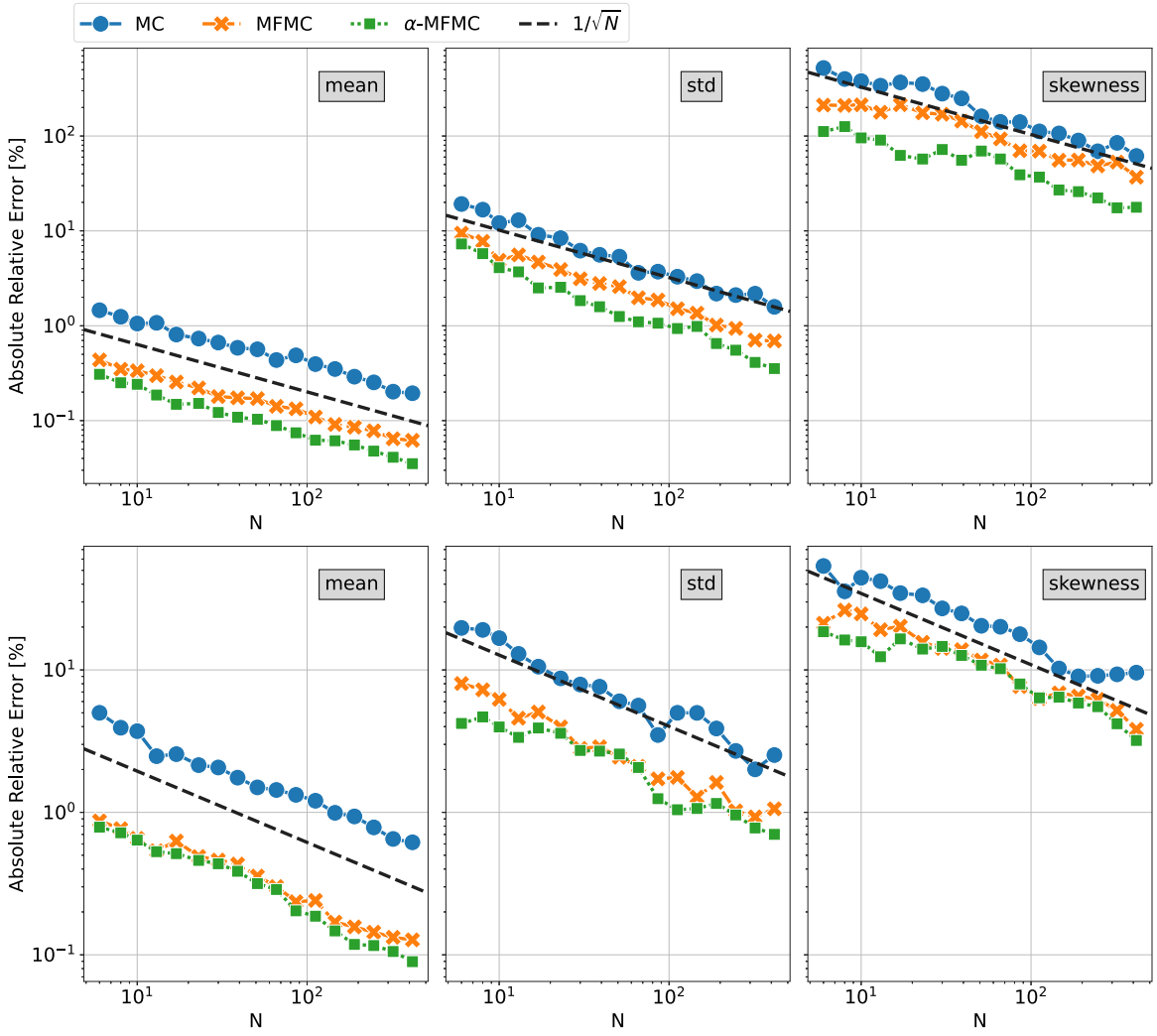

Fig. 6. Error convergence for three UQ methods, applied to the SNF assembly C12. Top plots show error in prediction of DH at 500 days of cooling, and bottom plots for ^{155}Gd concentration. Each UQ method was executed 50 times with different seeds for the random input selection, then the average errors of each method were plotted.

2. read out the nuclide content of the SNF assembly from the CASMO5 output file,
3. with OpenMC, model the 2D canister with 4 identical FAs (Fig. 12), with the nuclide content from the CASMO5 output,
4. perform a decay calculation (Bateman equation) of the canister with OpenMC for T years,
5. perform a transport calculation with OpenMC to compute k-inf.

Therefore in this example the studied family of models (alg. 3) consisted of SNF canister simulations at different times, and the models were expected to be correlated since they shared the same geometry and nuclide content, only differing due to the decay of nuclides. The uncertainty in each model f_T came from the random ND in step 1. No uncertainties were considered in the OpenMC calculations themselves, which used nominal nuclear data (formatted ACE files distributed with OpenMC based on ENDF/B-VII.1 Chadwick et al. (2011)). The canister model was filled with water and had the dimensions described in Vasiliev et al. (2019) (the only difference is that the original work had assembly boxes made of carbon steel, whereas this work considered boxes made of zircaloy). The OpenMC simulations used reflecting boundaries and an 8-fold symmetry to reduce computational costs. Each simulation required over 40 min of computational time on a single CPU, using six thousand neutron histories with 100 inactive batches and 100 active ones. The statistical uncertainty of the calculated k-inf was $\sigma_{\text{stat}} = (101 \pm 3)$ pcm.

Fig. 13 shows how the mean and variance of k-inf change with time, as the nuclide content decays. To perform UQ with algorithm 3, a low-fidelity model $\widetilde{f}$ was trained on $N = 500$ simulations of the model f_9, with the Lasso method (see full details on the training process in the Appendix B). The choice of using $T = 9$ years for training was based on the correlation matrix (Fig. 14), which suggests that at this time k-inf is strongly correlated to most other times T.

With the low-fidelity model $\widetilde{f}$, the α-MFMC method was applied to compute the k-inf at all other times, with $N = 10$. Since the success of α-MFMC depends on the correlations among models, it was expected that α-MFMC would be the most accurate for calculating k-inf at times $T > 9$ years, and less accurate for times $T < 2$ years due to the weak correlations (Fig. 14). This is indeed in agreement with the results in Fig. 15, where α-MFMC is clearly more accurate than simple MC for all times $T \geq 49$. For $T = 2$ α-MFMC is only slightly better than simple MC, and for $T = 0$ both methods are equivalent. The speedups from using α-MFMC (Table 5) are considerably lower than what was observed for DH and nuclide content (see Fig. 9 and Table 3). This observation is further confirmed by the theoretical speedups calculated from the correlations between models, which are in agreement with the observed speedups (Fig. 16). These results suggest that for cooling times $T \geq 10$ years, a single low-fidelity model can be used for applying α-MFMC for the UQ of k-inf. For shorter cooling times, a new low-fidelity model should be trained. This behaviour is due to the weak correlations between

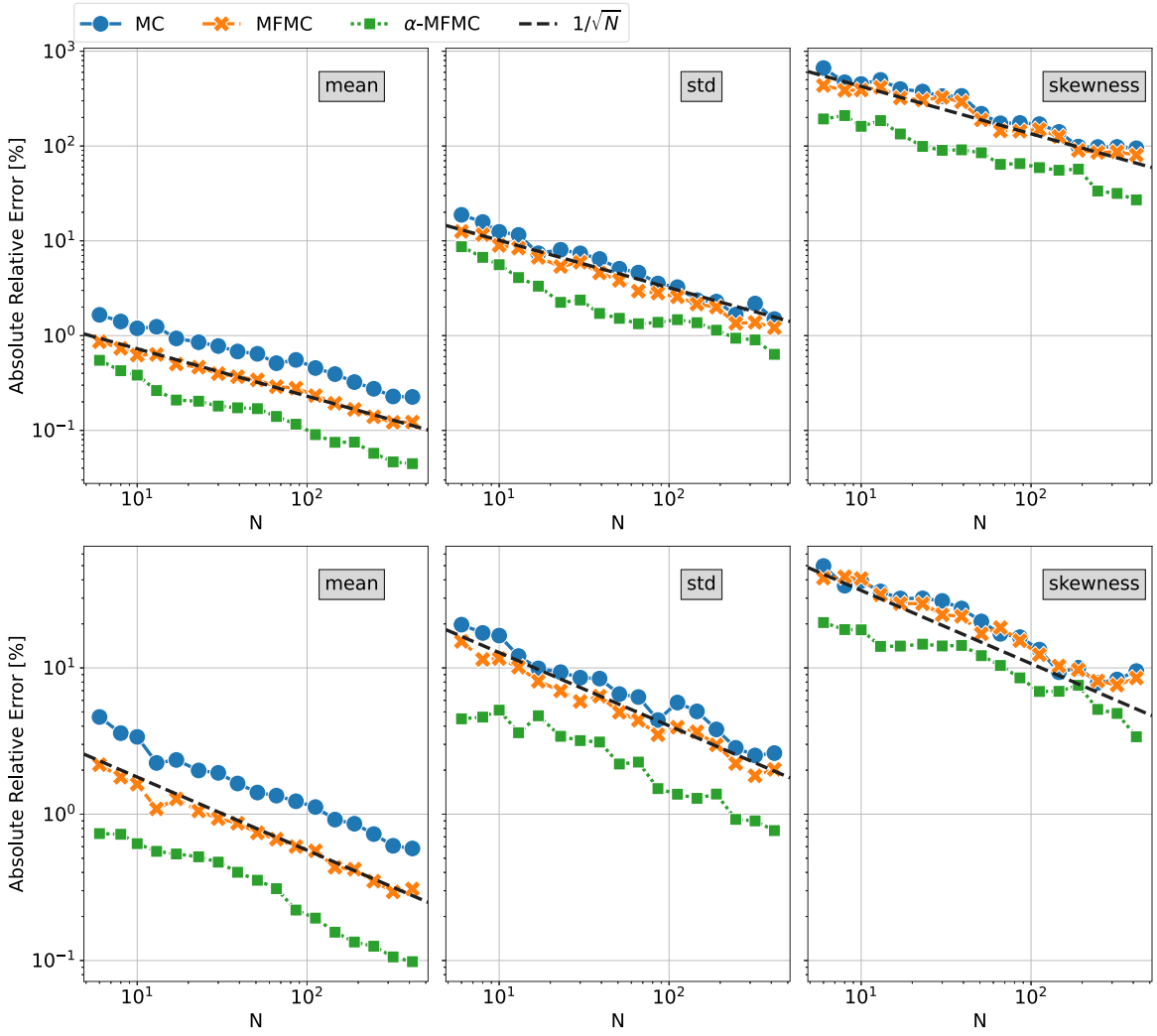

Fig. 7. Error convergence for three UQ methods, applied to the SNF assembly F32. Top plots show error in prediction of DH at 500 days of cooling, and bottom plots for ^{155}Gd concentration. Plots were generated by executing each UQ method 50 times with different seeds for the random input selection.

Table 5
Speedups from using α-MFMC instead of simple MC. Speedup was calculated with Eq. (5), and the data from Fig. 15.

Estimated moment	Mean	Variance
Average speedup	8.5	4.0
Median speedup	9.2	4.0
Minimum speedup	1.9	1.0
Maximum speedup	11.5	6.7
QoI with Min speedup	k-inf at 0 years	
QoI with Max speedup	k-inf at 5002 years	

models with short and long cooling times, likely due to the presence of short-lived nuclides, which dominate the k-inf at short cooling times.

4. Conclusion

The application of multifidelity Monte Carlo as an uncertainty quantification method for spent nuclear fuel was investigated for the first time. Two algorithms were introduced based on existing literature, MFMC and α-MFMC. Both methods optimally combine a high-fidelity model f with a machine learning low-fidelity model $\widetilde{f}$, to estimate the moments of the quantity of interest at a lower cost than simple MC. Additionally, a novel model management strategy was introduced, to alleviate the costs of training $\widetilde{f}$. This approach exploits the similarity of UQ problems, which allows for a single low-fidelity model $\widetilde{f}$ to be reused many times over for multiple problems.

The introduced methods were used to calculate the uncertainty in decay heat and nuclide content of SNF. The considered uncertainties were nuclear data, and design and operation parameters. The α-MFMC algorithm outperformed the other methods (MFMC and simple MC) for all QoIs in terms of accuracy and computational cost. This result was consistent for assemblies coming from different reactors (PWR and BWR) and with different designs and burnup histories. It was found that one low-fidelity model could be used for all UO_2 assemblies, and another model for MOX assemblies, but the models should not be interchanged as these two types of fuel have poorly correlated uncertainties.

The computational speedups from using α-MFMC instead of simple MC, ranged from a factor of 5 to a factor of 1500, depending on the QoI and the studied assembly. The observed speedup was in excellent agreement with the speedup predicted by the theory, and it was found that the speedup depends on how well the uncertainty of a QoI is correlated to the available low-fidelity model.

Finally, the UQ methods were also applied to the uncertainty of k-inf in an SNF canister at different cooling times. In this case a low-fidelity model was trained with criticality simulations at 9 years of cooling, and then used with α-MFMC for calculating the uncertainty at all other cooling times, ranging from 0 to 5000 years. For times larger than

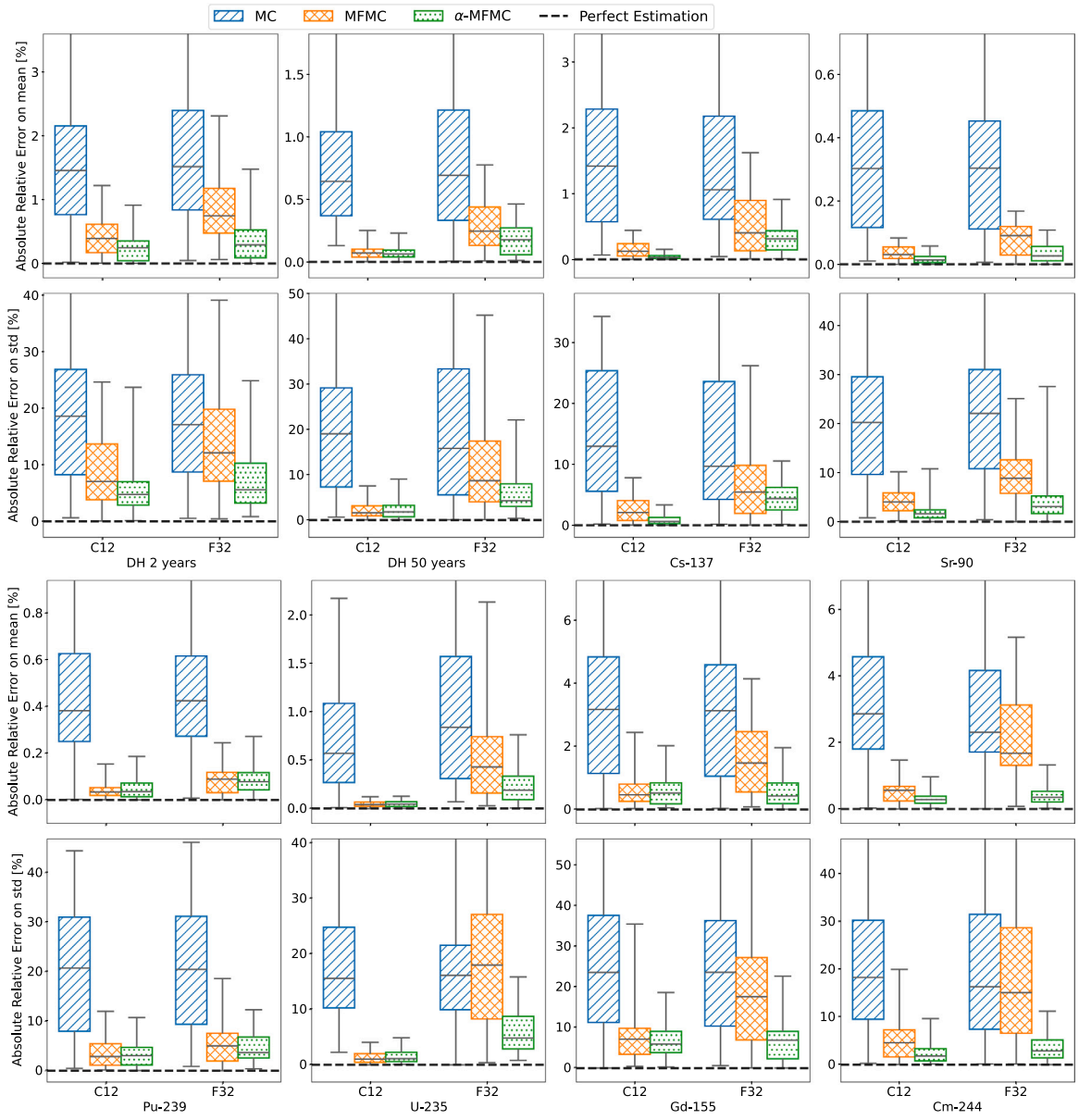

Fig. 8. Relative error of estimations with $N = 10$ samples, using the methods simple MC, MFMC, and α-MFMC. The error was calculated by comparing to the estimation of a simple MC run with $N = 6000$. Each method and QoI was performed 50 times with different seeds. The boxes represent the interquartile range (50% of points are within the box), the line inside the box is the median, and the whiskers represent the 1st and 4th quartile (the 25% smallest and largest errors are within the whiskers).

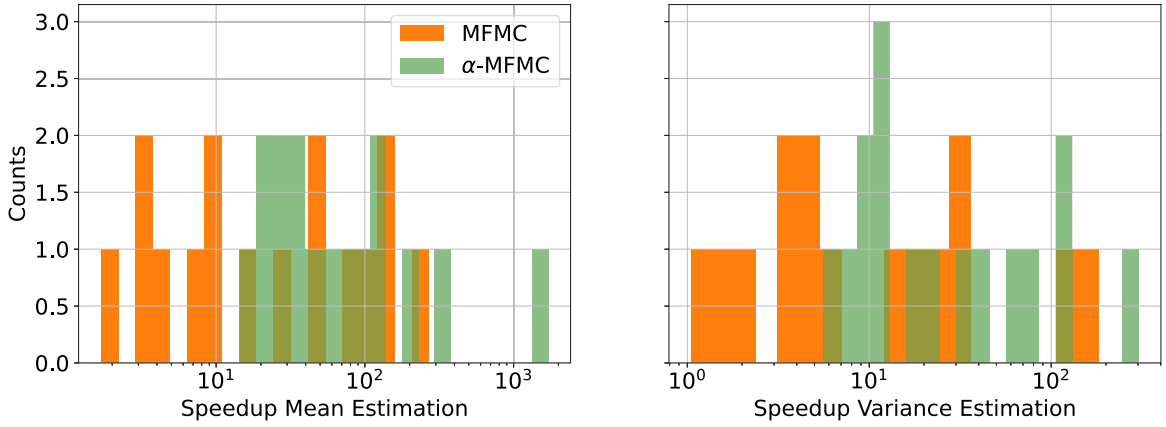

Fig. 9. Speedups from using multifidelity methods instead of simple MC. Speedup was calculated with Eq. (5), and the data from Fig. 8.

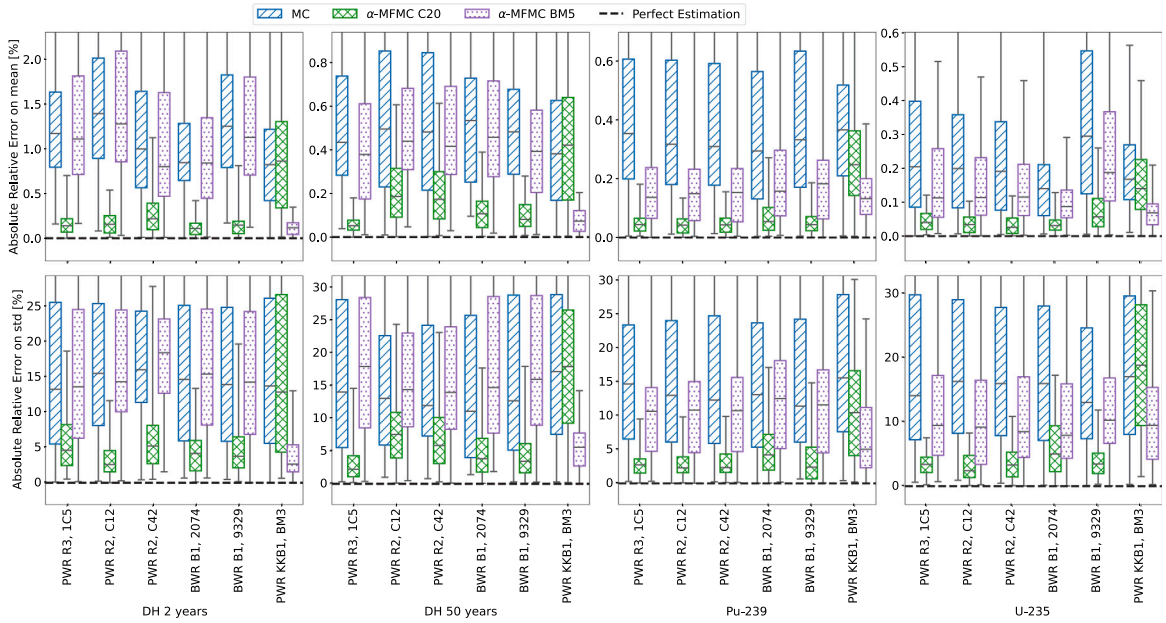

Fig. 10. Errors in the estimation of the mean and standard deviation of several QoIs, applied to different SNF assemblies. The two different α-MFMC methods use different low-fidelity models, one trained on the assembly C20 and the other on assembly BM5. Each method was performed 50 times for each assembly and QoI, with $N = 10$. The boxes represent the interquartile range, the horizontal line in the box is the median, and the whiskers are the 1st and 4th quartiles (the 25% lowest and largest errors are within the whiskers).

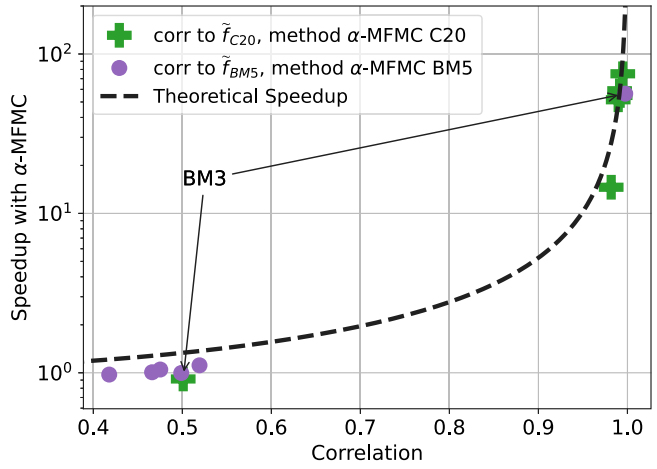

Fig. 11. Speedup with the α-MFMC method for computing the mean DH at 2 years of cooling. The theoretical speedup was calculated with Eq. (6). The x-axis is the correlation between the DH of each assembly and the output of the low-fidelity models $\widetilde{f}_{C20}$ and $\widetilde{f}_{BM5}$. The arrows show the points corresponding to assembly BM3, the only MOX assembly.

10 years, α-MFMC proved to be advantageous over simple MC, with speedups in the vicinity of a factor 10 for the mean estimation, and 5 for the variance estimation. However, for short cooling times, namely under 2 years of cooling, it was found that the uncertainties in k-inf are weakly correlated and thus α-MFMC shows no advantage over simple MC.

The presented methods have shown large reduction in computational costs for the UQ of SNF, but they are in principle applicable to any other UQ problem in the field of neutronic computations. A Python library (Albà, 2024) for MFMC and α-MFMC was provided, to enable further application of the methods.

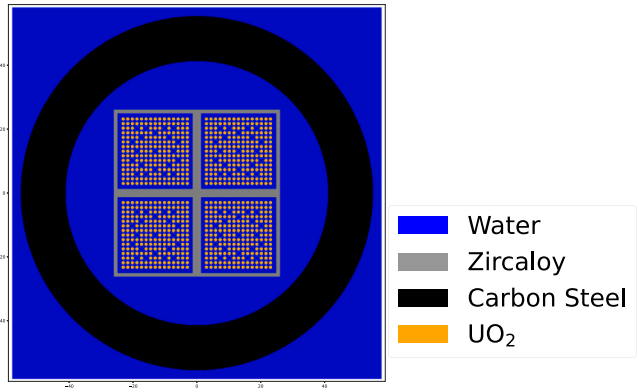

Fig. 12. The 2D canister model used for the OpenMC criticality calculations. The geometry was based on Vasiliev et al. (2019), although in that work the boxes containing the assemblies were made of carbon steel rather than zircaloy.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Arnau Alba reports financial support was provided by swissnuclear. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors thank Swissnuclear for partially sponsoring this work.

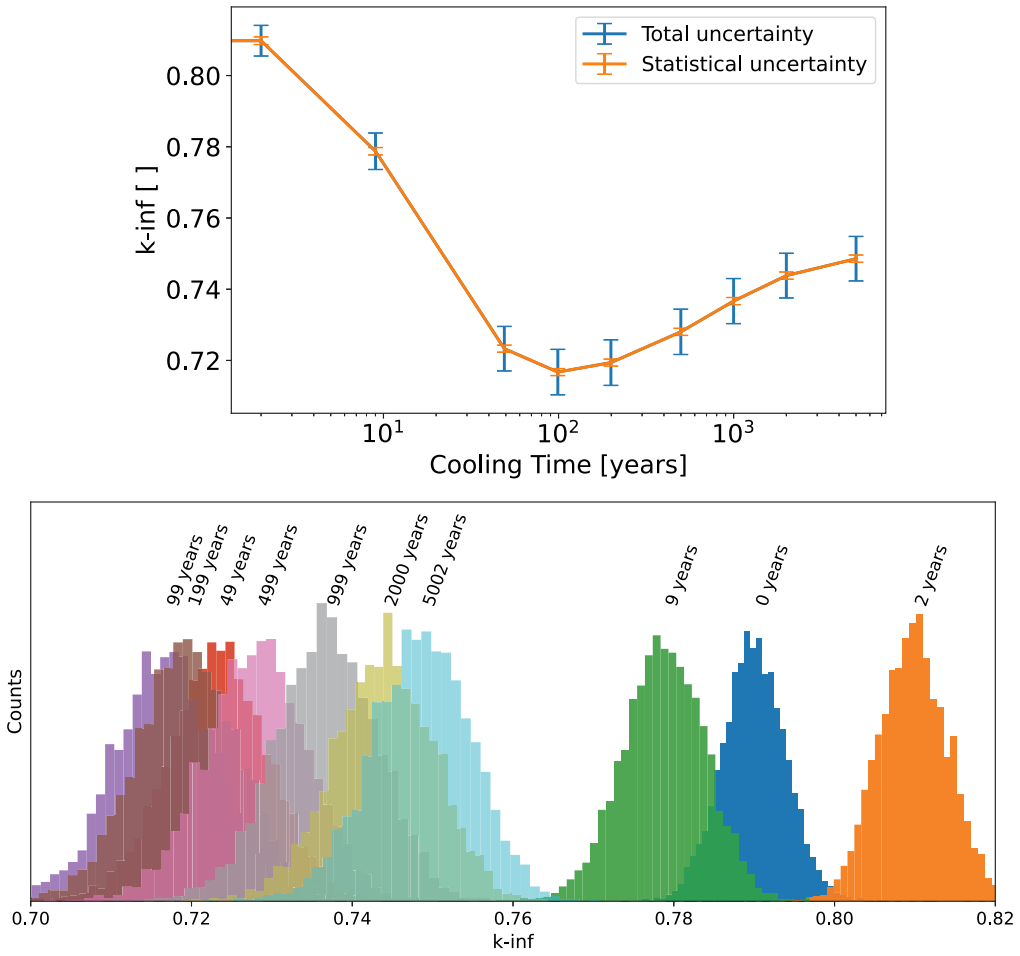

Fig. 13. Calculated k-inf of an SNF canister at different cooling times. The histograms were generated with 5300 simulations with random nuclear data.

Appendix A. Error analysis

For a given estimator $h_i\left[\{q\}_N\right]$ of a moment μ_i, the MSE (defined in Eq. (1)) can be split into a *bias* and *variance* parts:

$$\mathrm{MSE}\left(h_i\left[\{Q\}_N\right]\right) = \underbrace{\left(\mathbb{E}\left(h_i\left[\{Q\}_N\right]\right) - \mu_i\right)^2}_{\text{bias}} + \underbrace{\mathrm{Var}\left(h_i\left[\{Q\}_N\right]\right)}_{\text{variance}}, \tag{7}$$

with $\{Q\}_N = Q_1, Q_2, \ldots, Q_N$ a set of N i.i.d. random variables.

The simple MC estimators (Eq. (2)) and the MFMC estimators (Eq. (3)) are unbiased, hence

$$\mathbb{E}\left(h_i\right) = \mu_i[Q] \quad \Rightarrow \quad \mathrm{MSE}\left(h_i\right) = \mathrm{Var}\left(h_i\right),$$

and

$$\mathbb{E}\left(h_{\mathrm{MF},i}\right) = \mu_i[Q] \quad \Rightarrow \quad \mathrm{MSE}\left(h_{\mathrm{MF},i}\right) = \mathrm{Var}\left(h_{\mathrm{MF},i}\right).$$

where the expressions $h_i\left[\{Q\}_N\right]$ and $h_{\mathrm{MF},i}\left[\{Q\}_N, \{\widetilde{Q}\}_N, \{\widetilde{P}\}_M\right]$ have been shortened to h_i and $h_{\mathrm{MF},i}$ for the sake of notation. Recall that the random variables Q and $\widetilde{Q}$ are correlated since they are obtained from the same input, i.e. a random variable $\boldsymbol{X}$ is used to compute $Q = f(\boldsymbol{X})$ and $\widetilde{Q} = \widetilde{f}(\boldsymbol{X})$, whereas $\widetilde{P}$ is computed with $\widetilde{P} = \widetilde{f}(\boldsymbol{Z})$ where $\boldsymbol{Z}$ is independent and identically distributed as $\boldsymbol{X}$. In the following paragraphs, the MSE of each estimator is discussed in detail, as well as the bias introduced by the α-MFMC estimators (Eq. (4)).

A.1. Simple MC

The variance of the simple MC estimators for the first two moments is

$$\begin{aligned}
\mathrm{MSE}\left(h_1\left[\{Q\}_N\right]\right) &= \mathrm{Var}\left(h_1\left[\{Q\}_N\right]\right) = \frac{1}{N}\mu_2(Q), \\
\mathrm{MSE}\left(h_2\left[\{Q\}_N\right]\right) &= \mathrm{Var}\left(h_2\left[\{Q\}_N\right]\right) = \frac{1}{N}\left\{\mu_4(Q) - \frac{N-3}{N-1}\mu_2^2(Q)\right\},
\end{aligned} \tag{8}$$

where $\mu_4 = \mathbb{E}\left[(Q - \mathbb{E}[Q])^4\right]$ is the fourth order moment. In Krumscheid et al. (2020) the expressions of the MSE up to order 4 are derived, and it is shown that they all have a convergence rate of $\mathrm{MSE}\left(h_i\right) = \mathcal{O}(1/N)$.

A.2. MFMC

For the MFMC estimators the variance is given by

$$\begin{aligned}
\mathrm{MSE}\left(h_{\mathrm{MF},i}\left[\{Q\}_N, \{\widetilde{Q}\}_N, \{\widetilde{P}\}_M\right]\right) = &\underbrace{\mathrm{Var}\left(h_i\left[\{Q\}_N\right] - h_i\left[\{\widetilde{Q}\}_N\right]\right)}_{=\mathcal{O}(1/N)} \\
&+ \underbrace{\mathrm{Var}\left(h_i\left[\{\widetilde{P}\}_M\right]\right)}_{=\mathcal{O}(1/M)},
\end{aligned}$$

and thus $\mathrm{MSE}(h_{\mathrm{MF},i}) = \mathcal{O}\left(\max\left(\frac{1}{N}, \frac{1}{M}\right)\right)$. With Assumption 2.1 one can increase M at negligible costs, which guarantees that $M \gg N$.

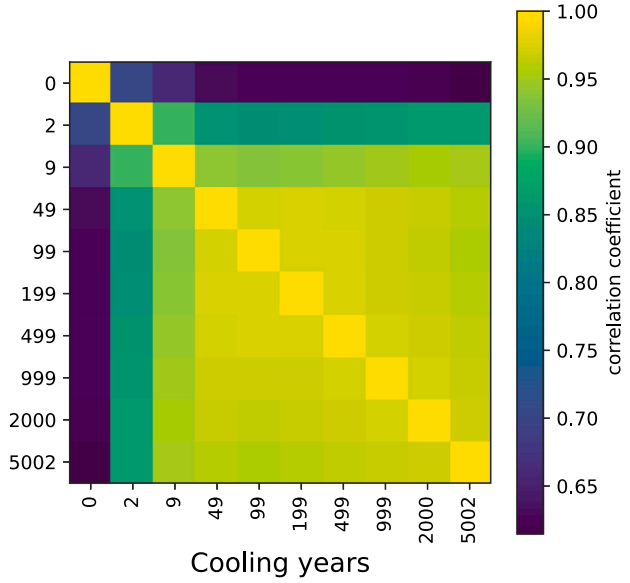

Fig. 14. Correlation between k-inf at different times, generated with 5300 simulations. Based on this plot, the model at $T = 9$ years was chosen for training a low-fidelity model $\widetilde{f}$.

Therefore

$$\text{Assumption 2.1} \quad \Rightarrow \quad \text{MSE}\left(h_{\text{MF},i}\left[\{Q\}_N, \{\widetilde{Q}\}_N, \{\widetilde{P}\}_M\right]\right) = \text{Var}\left(h_i\left[\{Q\}_N\right] - h_i\left[\{\widetilde{Q}\}_N\right]\right) = \mathcal{O}\left(\frac{1}{N}\right),$$

and the MSE of MFMC becomes independent of M. With this assumption, the MSEs (as derived in Krumscheid et al. (2020)) for the first two moments are

$$\begin{aligned}
\text{MSE}\left(h_{\text{MF},1}\left[\{Q\}_N, \{\widetilde{Q}\}_N, \{\widetilde{P}\}_M\right]\right) &= \frac{1}{N}\left\{\mu_2(Q) + \mu_2\left(\widetilde{Q}\right) - 2\text{Cov}\left(Q, \widetilde{Q}\right)\right\}, \\
\text{MSE}\left(h_{\text{MF},2}\left[\{Q\}_N, \{\widetilde{Q}\}_N, \{\widetilde{P}\}_M\right]\right) &= \text{Var}\left(h_2\left[\{Q\}_N\right]\right) + \text{Var}\left(h_2\left[\{\widetilde{Q}\}_N\right]\right) - \frac{2}{N}\left(\mu_{2,2}\left(Q, \widetilde{Q}\right) - \mu_2(Q)\mu_2(\widetilde{Q})\right) + \mathcal{O}\left(\frac{1}{N^2}\right),
\end{aligned} \tag{9}$$

with $\mu_{2,2}$ the bivariate second order moment defined as

$$\mu_{2,2}\left(Q, \widetilde{Q}\right) := \mathbb{E}\left[\left(Q - \mathbb{E}[Q]\right)^2\left(\widetilde{Q} - \mathbb{E}[\widetilde{Q}]\right)^2\right] \tag{10}$$

By comparing the MSEs of simple MC (Eq. (8)) and MFMC (Eq. (9)), several statements can be made:

- the MSE has the same convergence rate $\mathcal{O}\left(\frac{1}{N}\right)$ for all estimators,
- the MSEs of MFMC depend on the moments of $\widetilde{Q} = \widetilde{f}(\boldsymbol{X})$, and hence can be controlled by the choice of low-fidelity model $\widetilde{f}$, whereas the MSEs of simple MC cannot be controlled (other than by N),
- for the mean estimation $h_{\text{MF},1}$, the error is reduced if $\text{Cov}\left(Q, \widetilde{Q}\right)$ is large,
- for the variance estimation $h_{\text{MF},2}$, the error is reduced if the bivariate moment $\mu_{2,2}\left(Q, \widetilde{Q}\right)$ is large,
- more generally, in the limit where $\widetilde{f}$ is a perfect approximation of f, the error of MFMC is reduced to zero i.e.

$$\left|\widetilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right| = 0 \quad \forall \boldsymbol{x} \quad \Rightarrow \quad \text{MSE}[h_{\text{MF},i}] = 0 \quad \forall i \in \mathbb{Z}^+,$$

- if the low-fidelity model is a constant function, uncorrelated to f, then MFMC and simple MC have equal errors, i.e.

$$\widetilde{f}(\boldsymbol{x}) = \text{const} \quad \forall \boldsymbol{x} \quad \Rightarrow \quad \text{MSE}[h_{\text{MF},i}] = \text{MSE}[h_i] \quad \forall i \in \mathbb{Z}^+.$$

Based on these statements, the choice of low-fidelity model can be guided by a simple heuristic: a low-fidelity model should be chosen such that $\left|\widetilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right|$ is small everywhere, which will lead to reduced costs of the MFMC algorithm.

As shown in Eq. (5), the speedup from using one method or another can be computed as the ratio between their MSEs. If this is done for the μ_1 estimator the speedup is

$$\frac{\text{MSE}\left[h_1\right]}{\text{MSE}\left[h_{\text{MF},1}\right]} = \frac{1}{1 + \frac{\mu_2[\widetilde{Q}]}{\mu_2[Q]} - 2\frac{\text{Cov}[Q,\widetilde{Q}]}{\mu_2[Q]}},$$

where one can see that the speedup is independent of N, and is controlled by $\text{Cov}[Q, \widetilde{Q}]$. Furthermore, with the approximation $\mu_2[Q] \simeq \mu_2[\widetilde{Q}]$ the speedup reduces to Eq. (6).

A.3. α-MFMC

The idea of inserting a *control coefficient* α in the estimators is not new, it is commonly done for control variate methods and in the multifidelity literature (Peherstorfer et al., 2016; Qian et al., 2018), where the suggested estimator is

$$h_i\left[\{q\}_N\right] - \alpha\, h_i\left[\{\widetilde{q}\}_N\right] + \alpha\, h_i\left[\{\widetilde{p}\}_M\right]. \tag{11}$$

Note that this estimator is slightly different from the estimator employed in this paper (see Eq. (4), reproduced here):

$$h_i\left[\{q\}_N\right] - h_i\left[\alpha \cdot \{\widetilde{q}\}_N\right] + h_i\left[\alpha \cdot \{\widetilde{p}\}_M\right]. \tag{12}$$

The difference is that in Eq. (11) the coefficient α is multiplied in front of h_i, whereas in Eq. (12) α is directly multiplied with the low-fidelity outputs $\alpha \cdot \{\widetilde{q}\}_N = \alpha\widetilde{q}_1, \ldots, \alpha\widetilde{q}_N$. (When estimating the mean, $i = 1$, both equations are the same, but for higher order moments they differ.) The latter approach gave better results in practice, thus it was chosen for this work.

If α is a constant, chosen independently from the other $N + M$ samples, then Eq. (12) is unbiased, and the MSE is controlled by optimally tuning α. In the MFMC literature it is recommended to choose $\alpha = \text{Var}^{-1}[\widetilde{Q}]\text{Cov}[Q, \widetilde{Q}]$, which minimises the error of $h_{\text{MF},1}$. However, since the exact moments of Q and $\widetilde{Q}$ are not known, α needs to be estimated from a finite set of samples. In Peherstorfer et al. (2016) the suggestion is to use a small set of samples N', independent of the other $N + M$ samples, to estimate the control coefficient $\alpha_{N'}$. This incurs additional costs since the high-fidelity model is executed an additional N' times. In this work it was decided to avoid the N' additional samples.

The α-MFMC estimator (Eq. (4)) reuses the same N samples from $h_i\left[\{q\}_N\right]$ to estimate α_N. This approach has the disadvantage of introducing a small bias. However, it can be shown that it has the same convergence rate as simple MC and MFMC, i.e. $\text{MSE} = \mathcal{O}(1/N)$.

Theorem A.1 (*Mean Squared Error of α-MFMC*). *Let $\boldsymbol{X}$ be a random variable with a given probability density function, and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M$ be input vectors in $\mathbb{R}^d$ sampled from i.i.d. random variables distributed as $\boldsymbol{X}$, with $N \ll M$. Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ and $\widetilde{f} : \mathbb{R}^d \mapsto \mathbb{R}$ be two functions with computational costs satisfying Assumption 2.1, and with finite variance. Then the α-MFMC estimator defined in Eq. (4) has a bias that decays as $\mathcal{O}(N^{-1})$, and a mean squared error that decays as $MSE = \mathcal{O}(N^{-1})$.*

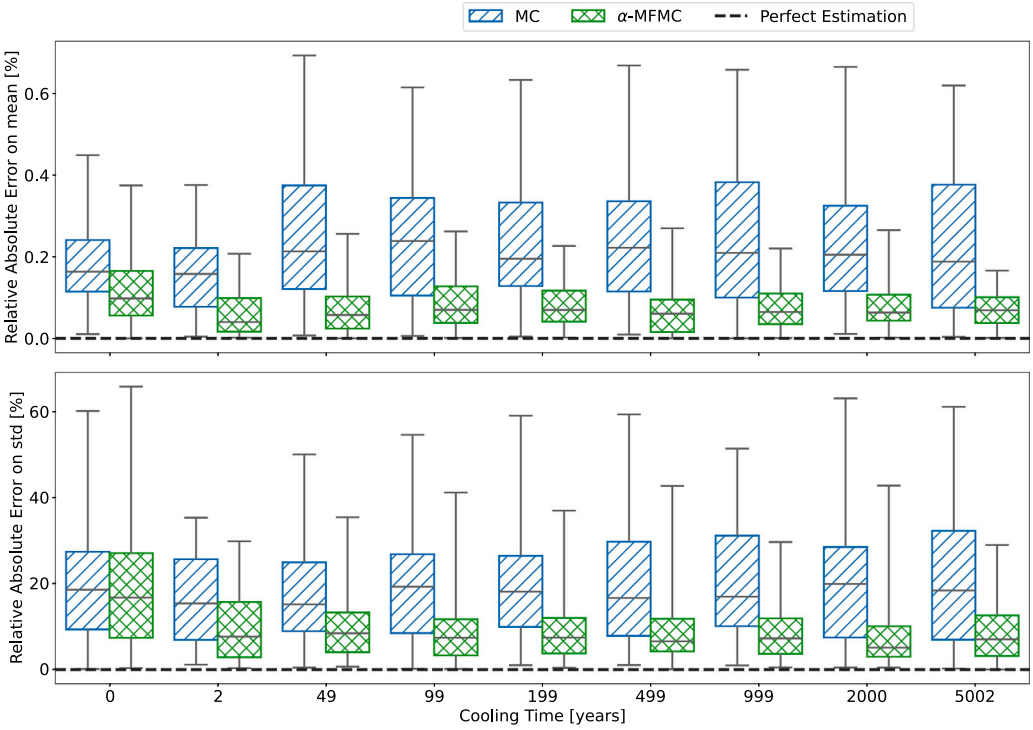

Fig. 15. Absolute error in the estimation of the mean and standard deviation of the k-inf at different cooling times, by using the simple MC and α-MFMC methods. Each method for each year was applied 50 times with random seeds, and $N = 10$. The boxes represent the interquartile range, and the whiskers represent the first and fourth quartiles.

Proof A.1. The absolute bias is given by $\left|\mathbb{E}\left(h_{\alpha-\mathrm{MF},i}\right) - \mu_i[Q]\right|$, and the expected value of the estimator is

$$\begin{aligned}\mathbb{E}\left(h_{\alpha-\mathrm{MF},i}\right) &= \mathbb{E}\left(h_i\left[\{Q\}_N\right]\right) - \mathbb{E}\left(h_i\left[\alpha_N \cdot \{\widetilde{Q}\}_N\right]\right) \\ &\quad + \mathbb{E}\left(\alpha_N^i\right)\mathbb{E}\left(h_i\left[\{P\}_M\right]\right) \\ &= \mu_i[Q] - \mathbb{E}\left(\alpha_N^i h_i\left[\{\widetilde{Q}\}_N\right]\right) + \mathbb{E}\left(\alpha_N^i\right)\mu_i[\widetilde{Q}] \\ &= \mu_i[Q] - \mathrm{Cov}\left(\alpha_N^i,\ h_i\left[\{\widetilde{Q}\}_N\right]\right)\end{aligned}$$

from which it follows that the bias is

$$\begin{aligned}\left|\mathbb{E}[h_{\alpha-\mathrm{MF},i}] - \mu_i[Q]\right| &= \left|\mathrm{Cov}\left(\alpha_N^i,\ h_i\left[\{\widetilde{Q}\}_N\right]\right)\right| \\ &\leq \sqrt{\mathrm{Var}\left(\alpha_N^i\right)\ \mathrm{Var}\left(h_i\left[\{Q\}_N\right]\right)} = \mathcal{O}\left(N^{-1}\right).\end{aligned}$$

Since the bias term appears squared in the expression for the MSE (see Eq. (7)), the MSE is dominated by the variance term:

$$\begin{aligned}\mathrm{MSE}\left(h_{\alpha\text{-MF},i}\right) &= \left(\mathbb{E}\left(h_{\alpha-\mathrm{MF},i}\right) - \mu_i\right)^2 + \mathrm{Var}\left(h_{\alpha\text{-MF},i}\left[\{Q\}_N\right]\right) \\ &= \mathcal{O}\left(\frac{1}{N^2}\right) + \mathrm{Var}\left(h_{\alpha\text{-MF},i}\left[\{Q\}_N\right]\right).\end{aligned}$$

The variance of the α-MFMC estimator is

$$\begin{aligned}\mathrm{Var}\left(h_{\alpha\text{-MF},i}\right) &= \mathrm{Var}\left(h_i\left[\{Q\}_N\right] - h_i\left[\alpha_N \cdot \{\widetilde{Q}\}_N\right]\right) \\ &\quad + \mathrm{Var}\left(h_i\left[\alpha_N \cdot \{\widetilde{P}\}_M\right]\right) \\ &\quad + 2\mathrm{Cov}\left(h_i\left[\{Q\}_N\right] - h_i\left[\alpha_N \cdot \{\widetilde{Q}\}_N\right],\ h_i\left[\alpha_N \cdot \{\widetilde{P}\}_M\right]\right) \\ &= \mathrm{Var}\left(h_i\left[\{Q\}_N\right] - \alpha_N^i h_i\left[\{\widetilde{Q}\}_N\right]\right) + \mu_i[\widetilde{Q}]\mathrm{Var}\left(\alpha_N^i\right) \\ &\quad + 2\mu_i[\widetilde{Q}]\mathrm{Cov}\left(h_i\left[\{Q\}_N\right] - \alpha_N^i h_i\left[\{\widetilde{Q}\}_N\right],\ \alpha_N^i\right),\end{aligned} \tag{13}$$

where it was assumed that M is large enough that $h_i\left[\{\widetilde{P}\}_M\right] = \mu_i[\widetilde{Q}]$ (following Assumption 2.1). All the terms in Eq. (13) have a convergence of $\mathcal{O}(N^{-1})$. Therefore

$$\mathrm{MSE}\left(h_{\alpha\text{-MF},i}\right) = \mathcal{O}\left(\frac{1}{N}\right). \quad \blacksquare$$

The theorem ensures that α-MFMC has the same convergence rate as simple MC and MFMC, but it does not provide any theoretical guarantees that it will have a larger or smaller MSE. Nevertheless, in the results Section 3 it was seen, empirically, that α-MFMC consistently has a smaller error than the other methods. An intuitive understanding of why this is the case can be obtained by calculating the speedup with Eq. (5). By making the approximation that the control coefficient is constant $\alpha_N \simeq \alpha$, the obtained speedup reads

$$\frac{\mathrm{MSE}\left[h_1\right]}{\mathrm{MSE}\left[h_{\alpha\text{-MF},1}\right]} = \frac{1}{1 + \alpha^2\frac{\mu_2[\widetilde{Q}]}{\mu_2[Q]} - 2\alpha\frac{\mathrm{Cov}[Q,\widetilde{Q}]}{\mu_2[Q]}}.$$

From this equation it is clear that the speedup is controlled by the coefficient α, and with the approximations $\mu_2[Q] \simeq \mu_2[\widetilde{Q}]$ and $\alpha \simeq \mathrm{Corr}[Q,\widetilde{Q}]$, one obtains the speedup from Eq. (6).

A.4. Error estimation

The stopping condition for the algorithms (algs. 1, 2) is that, for a desired error ε, one should have $\sqrt{\mathrm{MSE}} \leq \varepsilon$. However, the exact MSE cannot be calculated since the moments are unknown, therefore it needs to be estimated with the available finite set of samples. Two methods of estimating the MSE are suggested in the following paragraphs.

Firstly, an unbiased estimate $\widehat{\mathrm{MSE}}$ can be computed with formulas consisting of a linear combination of the h-estimators. For the first moment this is simply a matter of replacing μ_2 by h_2 in the MSE expressions (Eqs. (8) and (9)), which reads

$$\begin{aligned}\widehat{\mathrm{MSE}}\left(h_1\left[\{q\}_N\right]\right) &= \frac{1}{N}h_2\left[\{q\}_N\right], \\ \widehat{\mathrm{MSE}}\left(h_{\mathrm{MF},1}\left[\{q\}_N, \{\widetilde{q}\}_N, \{\widetilde{p}\}_M\right]\right) &= \frac{1}{N}h_2\left[\{q - \widetilde{q}\}_N\right] + \frac{1}{M}h_2\left[\{\widetilde{p}\}_M\right].\end{aligned}$$

For higher order estimators, the expressions become more complex and lengthy, and they are available in Krumscheid et al. (2020) and in the accompanying Python library (Albà, 2024). For the α-MFMC errors, the same formulas as MFMC are used by replacing $\widetilde{f}$ with $\alpha_N\widetilde{f}$.

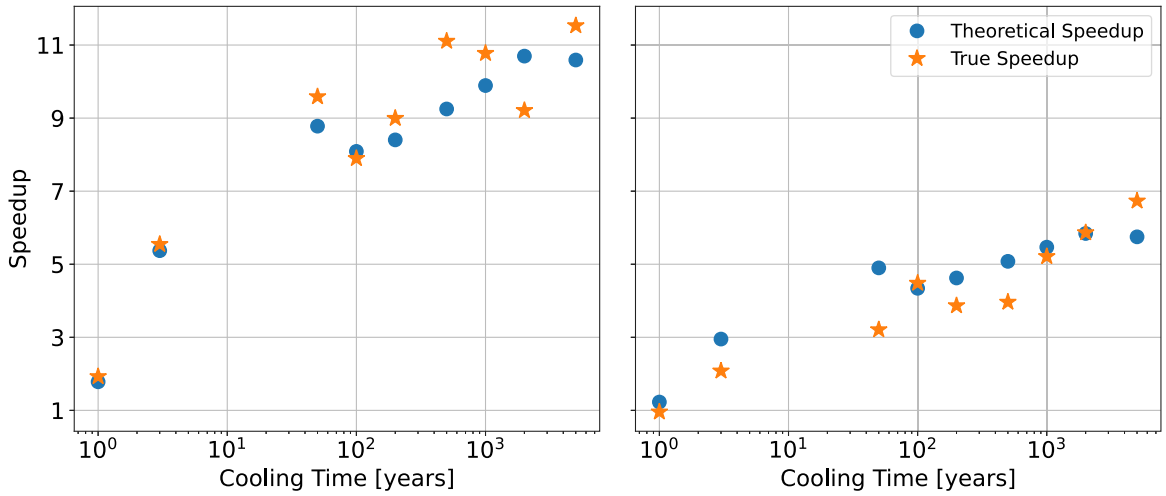

Fig. 16. Speedup from calculating the mean (left plot) and the variance (right plot) of the k-inf, with α-MFMC instead of simple MC. The true speedup was calculated with the data in Fig. 15, and the theoretical MSE was calculated with the formulas in Appendix A.

An example of estimating the MSE with these formulas can be found in Fig. 17(a), where the *running average* is surrounded by halo which represents $\pm\sqrt{\widehat{\text{MSE}}}$. The expected behaviour can be observed, where the simple MC error is much larger than the MFMC and α-MFMC errors.

A second method to estimate the errors is bootstrapping. This involves resampling N_{boot} times the N available samples (sampling with replacement), and thus computing N_{boot} estimators for the moments. The MSE is then estimated as the variance of the N_{boot} estimations. This approach is slower since it requires resampling and re-estimating several times. An example is shown in Fig. 17(b), with results practically indiscernible from Fig. 17(a). As shown in Fig. 18, the two strategies yield the same error estimates, thus both can be used for the stopping condition of the algorithms.

A.5. Case with normal distributions

In the literature of UQ for nuclear computations, it is common to assume that the output distribution Q is normally distributed, in order to simplify formulas (Eq. (8)), and to gain a more intuitive understanding of the MSE convergence. Additionally, as was seen in the results Section 3, it is often the case that when the input perturbations are small, the outputs do indeed resemble a normal distribution. Following this same approach, in this section the simple MC and MFMC errors are simplified for the case where outputs are normal.

Assume that the output is a normal distribution $Q \sim \mathcal{N}(\mu_1, \sigma)$ with mean μ_1 and standard deviation $\sigma = \sqrt{\mu_2}$. Furthermore assume that N is large enough such that $\mathcal{O}(1/N)$ terms become negligible w.r.t. $\mathcal{O}\left(1/\sqrt{N}\right)$. Then the simple MC errors for estimating the mean and standard deviation become

$$\sqrt{\text{MSE}\left(h_1\right)} = \frac{\sigma}{\sqrt{N}},$$

$$\sqrt{\text{MSE}\left(\sqrt{h_2}\right)} = \frac{\sigma}{\sqrt{2N}},$$

which are the MC error formulas commonly reported in the literature. The MFMC errors (Eq. (9)) can be simplified in a similar fashion, by assuming that $\widetilde{Q}$ has the same normal probability density function as Q. Then the errors become

$$\sqrt{\text{MSE}\left(h_{\text{MF},1}\right)} = \frac{\sigma}{\sqrt{N}}\sqrt{2}\sqrt{1-\rho},$$

$$\sqrt{\text{MSE}\left(\sqrt{h_{\text{MF},2}}\right)} = \frac{\sigma}{\sqrt{2N}}\sqrt{3}\sqrt{1-\eta},$$

with $\rho = \frac{1}{\sigma^2}\text{Cov}\left[Q, \widetilde{Q}\right]$ and $\eta = \frac{1}{3\sigma^4}\mu_{2,2}\left[Q, \widetilde{Q}\right]$, where $\mu_{2,2}$ is the bivariate second order moment defined in Eq. (10). Furthermore, ρ and η satisfy

$$-1 \le \rho \le 1, \quad \text{and} \quad 0 \le \eta \le 1.$$

With these simplified error expressions, the importance of the correlation factors ρ and η becomes clear.

Appendix B. Training the low-fidelity model

In all of the examples shown in the results Section 3, a low-fidelity machine learning model $\widetilde{f}$ was trained by fitting to a dataset of $N = 500$ input–output samples, with an input dimension $d > 15\,000$. Since the problem is grossly underdetermined, i.e. the number of training samples is much smaller than the input dimension $N \ll d$, the employed method was a Lasso machine learning model (Tibshirani, 1996). Lasso is a sparse linear model, meaning that is has the form $\widetilde{f}(\boldsymbol{x}) = \boldsymbol{\beta} \cdot \boldsymbol{x}$, with the peculiarity that the coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^d$ is sparse, i.e. it contains many zero elements. A linear model approach offers a crude first-order approximation to the high-fidelity model f, but in UQ problems where the input perturbations are small, one expects a $\widetilde{f}$ and f to be well correlated. The Lasso model is trained by finding $\boldsymbol{\beta}$ that minimises the loss function

$$\mathcal{L}\left(\boldsymbol{\beta}, \{\boldsymbol{x}\}_N, \{q\}_N\right) = \sum_{i=1}^{N}\left(q_i - \boldsymbol{\beta}\cdot\boldsymbol{x}_i\right)^2 + \lambda\|\boldsymbol{\beta}\|_1, \quad \text{with} \quad \|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{d}|\beta_j|,$$

where the first loss term is the usual ordinary least squares error, and the second term is a regularisation term that prevents overfitting, ensures that the problem is not ill-posed, and causes $\boldsymbol{\beta}$ to be sparse. The hyperparameter λ controls the magnitude of the regularisation term, and by proxy the sparseness of $\boldsymbol{\beta}$. Generally λ is chosen such that prediction accuracy of $\widetilde{f}$ is maximised on a validation set, or with k-fold cross-validation. However these strategies require splitting the training set and iteratively training many models, which renders the training process more computationally expensive. For this reason in this work the implemented strategy was to choose λ such that $supp(\boldsymbol{\beta}) = 95\% \cdot N$, where $supp(\boldsymbol{\beta})$ is the number of non-zero elements of $\boldsymbol{\beta}$.

The Lasso model was fitted with the implementation from the Scikit-Learn Python library (Pedregosa et al., 2011), which uses the coordinate descent algorithm (Wu and Lange, 2008). The training data was scaled before training, with a standard scaler for the inputs (divide by standard deviation and subtract mean), and a Yeo-Johnson transformation (Yeo and Johnson, 2000) (reduce skewness, approximate normality) for the output.

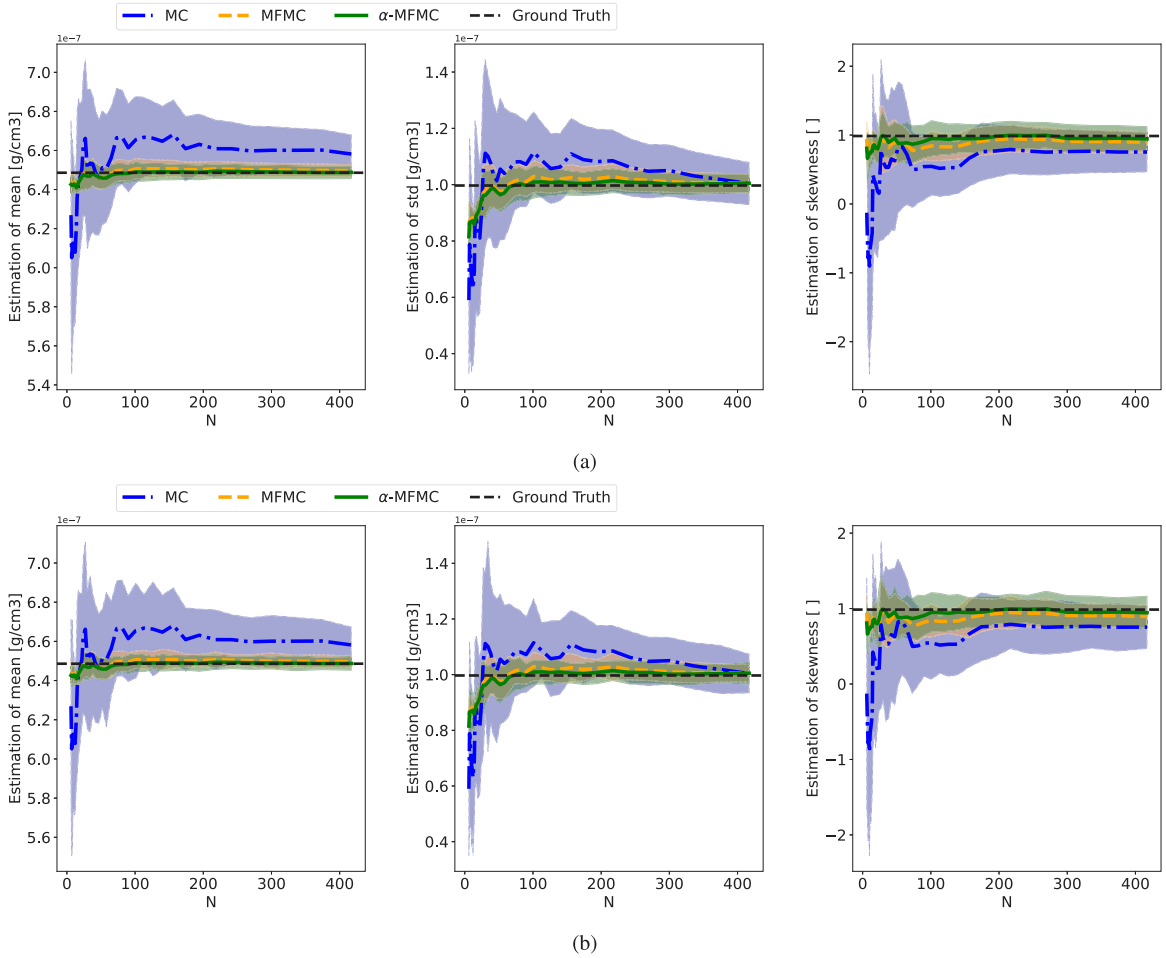

Fig. 17. Running average plots for the estimation of ^{155}Gd concentration in the fuel assembly C12. The coloured-in regions correspond to the estimated error $\pm\sqrt{\widehat{\mathrm{MSE}}}$, calculated with a formula (a), or with bootstrapping (b). The bootstrap was performed with $N_{boot} = 50$ resamplings.

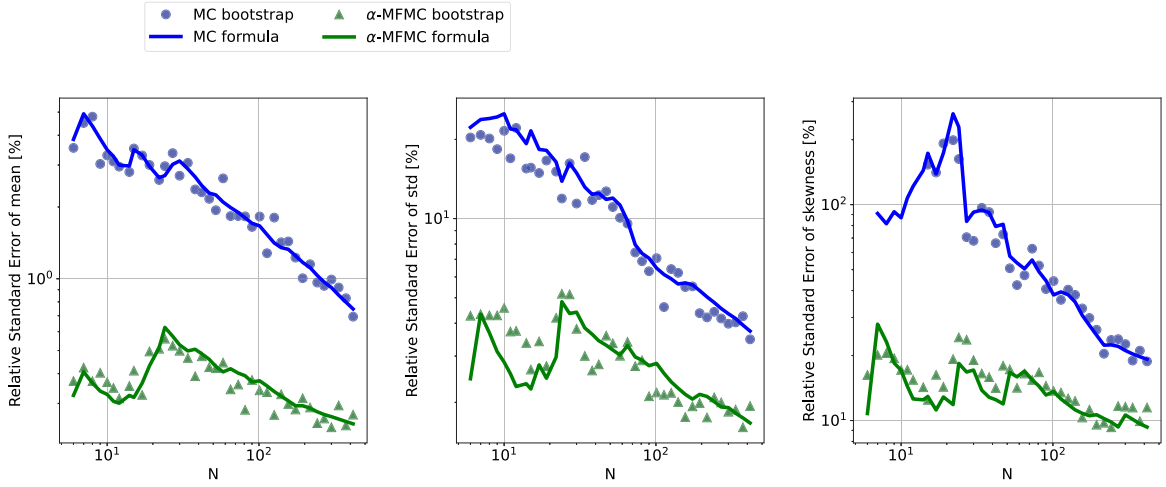

Fig. 18. Estimated standard error $\sqrt{\widehat{\mathrm{MSE}}}$ for the estimation of ^{155}Gd in C12, as seen in Fig. 5. The bootstrap estimates were calculated with $N_{boot} = 50$ samples.

References

Adams, B.M., Eldred, M.S., Geraci, G., Portone, T., Ridgway, E.M., Stephens, J.A., Wildey, T.M., 2022. Deployment of Multifidelity Uncertainty Quantification for Thermal Battery Assessment Part I: Algorithms and Single Cell Results. Technical Report SAND2022-11856, Sandia National Lab. (SNL-NM), Albuquerque, NM (United States.

Albà, A., 2024. 2-level MFMC for spent nuclear fuel. https://github.com/arnaualba/2-level-MFMC.

Albà, A., Adelmann, A., Münster, L., Rochman, D., Boiger, R., 2024. Fast uncertainty quantification of spent nuclear fuel with neural networks. Ann. Nucl. Energy 196, 110204.

Albà, A., Adelmann, A., Rochman, D., 2023a. Uncertainty quantification on spent nuclear fuel with LMC. In: International Conference on Nuclear Criticality Safety (ICNC) 2023. Sendai, Japan, arXiv. arXiv:2309.00364 [physics].

Albà, A., Boiger, R., Rochman, D., Adelmann, A., 2023b. Lasso Monte Carlo, a variation on multi fidelity methods for high dimensional uncertainty quantification. arXiv: 2210.03634 [stat].

Amela, R., Ayoul-Guilmard, Q., Badia, R.M., Ganesh, S., Nobile, F., Rossi, R., Tosi, R., 2019. XMC (1.0.0).

Aures, A., Bostelmann, F., Hursin, M., Leray, O., 2017. Benchmarking and application of the state-of-the-art uncertainty analysis methods XSUSA and SHARK-X. Ann. Nucl. Energy 101, 262–269.

Cacuci, D.G., 2010. Sensitivity and uncertainty analysis of models and data. In: Nuclear Computational Science: A Century in Review. Springer Netherlands, Dordrecht, pp. 291–353.

Caruso, S., Pantelias Garcés, M., 2015. Spent nuclear fuel management in Switzerland: Perspective for final disposal. In: International Conference on Management of Spent Nuclear Fuel from Nuclear Power Reactors: an Integrated Approach To the Back End of the Fuel Cycle, IAEA. Vienna.

Chadwick, M.B., Herman, M., Obložinský, P., Dunn, M.E., Danon, Y., Kahler, A.C., Smith, D.L., Pritychenko, B., Arbanas, G., Arcilla, R., Brewer, R., Brown, D.A., Capote, R., Carlson, A.D., Cho, Y.S., Derrien, H., Guber, K., Hale, G.M., Hoblit, S., Holloway, S., Johnson, T.D., Kawano, T., Kiedrowski, B.C., Kim, H., Kunieda, S., Larson, N.M., Leal, L., Lestone, J.P., Little, R.C., McCutchan, E.A., MacFarlane, R.E., MacInnes, M., Mattoon, C.M., McKnight, R.D., Mughabghab, S.F., Nobre, G.P.A., Palmiotti, G., Palumbo, A., Pigni, M.T., Pronyaev, V.G., Sayer, R.O., Sonzogni, A.A., Summers, N.C., Talou, P., Thompson, I.J., Trkov, A., Vogt, R.L., van der Marck, S.C., Wallner, A., White, M.C., Wiarda, D., Young, P.G., 2011. ENDF/B-VII.1 nuclear data for science and technology: Cross sections, covariances, fission product yields and decay data. Nucl. Data Sheets 112 (12), 2887–2996.

Ebiwonjumi, B., Cherezov, A., Dzianisau, S., Lee, D., 2021. Machine learning of LWR spent nuclear fuel assembly decay heat measurements. Nucl. Eng. Technol. 53 (11), 3563–3579.

Ebiwonjumi, B., Kong, C., Zhang, P., Cherezov, A., Lee, D., 2020. Uncertainty quantification of PWR spent fuel due to nuclear data and modeling parameters. Nucl. Eng. Technol. S1738573320303521.

ENSI, 2020. Richtlinie für die Schweizerischen Kernanlagen. Technical Report ENSI-G03/d, Eidgenössisches Nuklearsicherheitsinspektorat ENSI.

Frankl, M., Hursin, M., Rochman, D., Vasiliev, A., Ferroukhi, H., 2021. Nuclear data uncertainty quantification in criticality safety evaluations for spent nuclear fuel geological disposal. Appl. Sci. 11 (14), 6499, Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

Giles, M.B., 2008. Multilevel Monte Carlo path simulation. Oper. Res. 56 (3), 607–617.

Gruber, A., Gunzburger, M., Ju, L., Lan, R., Wang, Z., 2023. Multifidelity Monte Carlo estimation for efficient uncertainty quantification in climate-related modeling. Geosci. Model Dev. 16 (4), 1213–1229, Publisher: Copernicus GmbH.

Halmos, P.R., 1946. The theory of unbiased estimation. Ann. Math. Stat. 17 (1), 34–43.

Herrero, J., Rochman, D., Leray, O., Vasiliev, A., Pecchia, M., Ferroukhi, H., Caruso, S., 2017a. Impact of nuclear data uncertainty on safety calculations for spent nuclear fuel geological disposal. In: Plompen, A., Hambsch, F.-J., Schillebeeckx, P., Mondelaers, W., Heyse, J., Kopecky, S., Siegler, P., Oberstedt, S. (Eds.), EPJ Web Conf. 146, 09028.

Herrero, J., Vasiliev, A., Pecchia, M., Rochman, D., Ferroukhi, H., Johnson, L., Caruso, S., 2017b. Criticality Safety Assessment for Geological Disposal of Spent Fuel Using PSI BUCSS-R Methodology. Technical Report Arbeitsbericht NAB 17-23, Nagra.

Koning, A.J., Rochman, D., 2008. Towards sustainable nuclear energy: Putting nuclear physics to work. Ann. Nucl. Energy 35 (11), 2024–2030.

Koppen, M., 2000. The curse of dimensionality. In: 5th online world conference on soft computing in industrial applications (WSC5). Vol. 1, pp. 4–8.

Krumscheid, S., Nobile, F., Pisaroni, M., 2020. Quantifying uncertain system outputs via the multilevel Monte Carlo method — Part I: Central moment estimation. J. Comput. Phys. 414, 109466.

Law, F., Cerfon, A., Peherstorfer, B., 2022. Accelerating the estimation of collisionless energetic particle confinement statistics in stellarators using multifidelity Monte Carlo. Nucl. Fusion 62 (7), 076019, Publisher: IOP Publishing.

Leray, O., Fiorito, L., Rochman, D., Ferroukhi, H., Stankovskiy, A., Van den Eynde, G., 2017. Uncertainty propagation of fission product yields to nuclide composition and decay heat for a PWR UO2 fuel assembly. Prog. Nucl. Energy 101, 486–495.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12 (85), 2825–2830.

Peherstorfer, B., Willcox, K., Gunzburger, M., 2016. Optimal model management for multifidelity Monte Carlo estimation. SIAM J. Sci. Comput. 38 (5), A3163–A3194.

Peherstorfer, B., Willcox, K., Gunzburger, M., 2018. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. SIAM Rev. 60 (3), 550–591.

Primm, R., 2002. ARIANE International Programme Final Report. Technical Report ORNL/SUB–97-XSV750-1, ORNL Oak Ridge National Laboratory (United States). Funding organisation: US Department of Energy (United States), United States, ORNL/SUB–97-XSV750-1 INIS Reference Number: 34070003.

Qian, E., Peherstorfer, B., O'Malley, D., Vesselinov, V.V., Willcox, K., 2018. Multifidelity Monte Carlo estimation of variance and sensitivity indices. SIAM/ASA J. Uncertain. Quantif. 6 (2), 683–706.

Radaideh, M.I., Price, D., Kozlowski, T., 2021. Modeling nuclear data uncertainties using deep neural networks. In: Margulis, M., Blaise, P. (Eds.), EPJ Web Conf. 247, 15016.

Rhodes, J., Smith, K., Lee, D., 2006. CASMO-5 Development and Applications. Technical Report, American Nuclear Society - ANS, La Grange Park (United States.

Rochman, D., van der Marck, S.C., Koning, A.J., Sjöstrand, H., Zwermann, W., 2014. Uncertainty propagation with fast Monte Carlo techniques. Nucl. Data Sheets 118, 367–369.

Rochman, D., Vasiliev, A., Ferroukhi, H., Hursin, M., 2021. Analysis for the ARIANE BM1 and BM3 samples: nuclide inventory and decay heat. EPJ Nucl. Sci. Technol. 7, 18, Publisher: EDP Sciences.

Rochman, D., Vasiliev, A., Ferroukhi, H., Zhu, T., van der Marck, S.C., Koning, A.J., 2016. Nuclear data uncertainty for criticality-safety: Monte Carlo vs. linear perturbation. Ann. Nucl. Energy 92, 150–160.

Romano, P.K., Horelik, N.E., Herman, B.R., Nelson, A.G., Forget, B., Smith, K., 2015. OpenMC: A state-of-the-art Monte Carlo code for research and development. Ann. Nucl. Energy 82, 90–97.

Shama, A., Rochman, D., Caruso, S., Pautz, A., 2022. Validation of spent nuclear fuel decay heat calculations using Polaris, ORIGEN and CASMO5. Ann. Nucl. Energy 165, 108758.

Shama, A., Rochman, D., Pudollek, S., Caruso, S., Pautz, A., 2021. Uncertainty analyses of spent nuclear fuel decay heat calculations using SCALE modules. Nucl. Eng. Technol..

Solans, V., Rochman, D., Brazell, C., Vasiliev, A., Ferroukhi, H., Pautz, A., 2021. Optimisation of used nuclear fuel canister loading using a neural network and genetic algorithm. Neural Comput. Appl. 33 (23), 16627–16639.

Steering Committee for Nuclear Energy, 2016. Evaluation Guide for the Evaluated Spent Nuclear Fuel Assay Database (SFCOMPO). Technical Report NEA/NSC/R(2015)8, OECD Nuclear Energy Agency.

Sturek, F., Agrenius, L., Osifo, O., 2006. Measurements of Decay Heat in Spent Nuclear Fuel at the Swedish Interim Storage Facility, Clab. Technical Report R-05-62, Svensk Kärnbränslehantering AB, p. 253.

Tibshirani, R., 1996. Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1), 267–288.

Vasiliev, A., Herrero, J., Pecchia, M., Rochman, D., Ferroukhi, H., Caruso, S., 2019. Preliminary assessment of criticality safety constraints for Swiss spent nuclear fuel loading in disposal canisters. Materials 12 (3), 494.

Vasiliev, A., Herrero, J., Rochman, D., Pecchia, M., Ferroukhi, H., Caruso, S., 2018. Criticality safety evaluations for the concept of Swiss PWR spent fuel geological repository. In: ANS Best Estimate Plus Uncertainty International Conference (BEPU 2018). American Nuclear Society, Lucca, Italy, p. 12.

Wieselquist, W., Zhu, T., Vasiliev, A., Ferroukhi, H., 2013. PSI methodologies for nuclear data uncertainty propagation with CASMO-5M and MCNPX: Results for OECD/NEA UAM benchmark Phase I. Sci. Technol. Nucl. Install. 2013, 1–15.

Wu, T.T., Lange, K., 2008. Coordinate descent algorithms for lasso penalized regression. Ann. Appl. Stat. 2 (1).

Yeo, I.-K., Johnson, R.A., 2000. A new family of power transformations to improve normality or symmetry. Biometrika 87 (4), 954–959.