UNCERTAINTY QUANTIFICATION ON SPENT NUCLEAR FUEL WITH LMC

Arnau Albà[1,2], **Andreas Adelmann**[1], **Dimitri Rochman**[1]

[1] *Paul Scherrer Institut, Forschungstrasse 111, 5232 Villigen, Switzerland*

[2] *ETH Zurich, Rämistrasse 101, 8092 Zurich Switzerland*
*arnau.albajacas@psi.ch

ABSTRACT

The recently developed method Lasso Monte Carlo (LMC) for uncertainty quantification is applied to the characterisation of spent nuclear fuel. The propagation of nuclear data uncertainties to the output of calculations is an often required procedure in nuclear computations. Commonly used methods such as Monte Carlo, linear error propagation, or surrogate modelling suffer from being computationally intensive, biased, or ill-suited for high-dimensional settings such as in the case of nuclear data. The LMC method combines multilevel Monte Carlo and machine learning to compute unbiased estimates of the uncertainty, at a lower computational cost than Monte Carlo, even in high-dimensional cases. Here LMC is applied to the calculations of decay heat, nuclide concentrations, and criticality of spent nuclear fuel placed in disposal canisters. The uncertainty quantification in this case is crucial to reduce the risks and costs of disposal of spent nuclear fuel. The results show that LMC is unbiased and has a higher accuracy than simple Monte Carlo.

KEYWORDS

Uncertainty quantification, Nuclear data, Decay heat, Criticality, Spent nuclear fuel

1. INTRODUCTION

Nuclear codes make predictions about spent nuclear fuel (SNF), that are then used by regulators to make informed decisions that reduce the risks and costs of SNF storage. It is therefore important that the uncertainties of these codes are quantified and taken into account when making predictions. This work studies methods to calculate the uncertainty in quantities used to characterise SNF, namely the decay heat, nuclide inventory, and the k-effective of the SNF canisters.

One of the main sources of uncertainty in neutron transport codes is nuclear data, which include tens of thousands of uncertain parameters such as neutron cross-sections, fission product yields, neutron multiplicity, or the fission spectra. The uncertainty propagation from nuclear data has traditionally been done with either linear perturbation methods, or Monte Carlo (MC) [1]. The former method assumes the linearity of the response function, and requires computing the sensitivity of the response with respect to each input dimension, thus it is potentially biased and its cost increases with the input dimension. On the other hand, MC methods are dimension-independent and make no (or weak) assumptions on the distribution of the output. However, MC is known to have a slow convergence and needing a large number of simulations, rendering the method computationally expensive.

In the case of Switzerland, over 12 thousand SNF assemblies are expected [2], corresponding to a 60-year operation of currently operating nuclear power plants. Computing the nuclear data uncertainty requires close to a thousand simulations per assembly, with each simulation running in an order of ten hours on a single CPU core. This amounts to a rough estimate of at least 100 million expected CPU hours for all the uncertainty calculations required in Switzerland. Therefore, it is of high interest to investigate methods that reduce the computational costs of uncertainty quantification (UQ) from nuclear data.

Recent advances in machine learning have shown large increases in speed in several domains, with the use of fast surrogate models that replace the computationally expensive codes (e.g. [3]). However, surrogate models, similarly to linear perturbation methods, introduce model bias by making assumptions on the response function, and suffer from the curse of dimensionality [4]: since nuclear data has tens of thousands of dimensions, training a surrogate model would require tens of thousands of expensive simulations. Other UQ approaches that focus on dimensionality reduction, such as principle component analysis [5] or active subspaces [6], have not been investigated in this work. However, these methods are in general also biased as they ignore the effect of certain inputs dimensions.

In this work, the Lasso Monte Carlo (LMC) method [7] is applied to UQ from nuclear data. This method uses a Lasso [8] surrogate model, which is a simple sparse linear model that requires few training samples, and then computes the moments of the response function with Multilevel Monte Carlo [9], which corrects for the bias of the surrogate model. Overall, the method is unbiased and more accurate than MC. This is computationally verified with UQ calculations on a variety of quantities for two fuel assembly (FA) examples, one uranium oxide case (UO2) and one mixed oxide case (MOX). Note that this paper only includes a brief explanation of the LMC method without any proofs, as the full details of the method can be found in [7].

The rest of the paper is organised as follows. In the *methodology section* several UQ methods are explained, with emphasis on the most recent method, LMC. This section also includes details regarding the computational tools, the uncertainties in nuclear data, and the two studied fuel assemblies. The results section compares LMC to the other described methods, all of which are applied to a range of quantities of interest. Finally, the conclusion summarises the work and discusses future research directions.

2. METHODOLOGY

2.1. Nuclear Codes

The burnup calculations were carried out with the nuclear lattice code CASMO5 [10]. A single fuel assembly was modelled as a 2D slice with reflecting boundaries. Starting from the fresh fuel characteristics, CASMO5 repeatedly solves the decay equation and the nuclear transport equation to simulate all the burnup and cooling cycles of the assembly, until its end of life (EOL). The output quantities, that were of interest in this work, were the decay heat w.r.t. cooling time, and the nuclide inventory at EOL.

For the UO2 simulations, large burnup steps with cycle-wise averaged quantities were used (as was done in [11]), and a quarter-symmetry was considered. These simulations needed approximately 10 minutes each. For the MOX case, a detailed burnup history with in-cycle variations was used, since it was available, and the full 2D slice of the FA was simulated, without any symmetry assumptions. The MOX burnup simulations needed approximately 3 hours each. All simulations and further calculations were carried out on single CPU cores, specifically on Intel Xeon Gold 6152 processors with 2.10 GHz and 4 GB of RAM.

The PSI in-house code Shark-X [12, 13] was used to perturb the nuclear data in the 20 energy groups used by CASMO5, based on the covariances of ENDF/B-VII.1 [14]. The perturbed quantities were the neutron elastic and inelastic cross-sections, neutron capture cross-sections, fission cross-sections, fission product yields, fission spectra, and neutron multiplicity. The total number of perturbed nuclear data values was 15 557.

The criticality calculations were preformed with the open source Monte Carlo code OpenMC [15]. These calculations were only carried out for the UO2 assemblies. The modelled system was an SNF canister containing 4 identical depleted FAs, surrounded by a 14 cm layer of carbon steel, and filled with water. Fig 1 shows the model of a spent fuel canister, based on the dimensions and arrangement used in [16] (although here the FAs are placed in zircaloy boxes rather than carbon steel boxes as in the original work). The calculations were done in 2D, and with an 8-fold symmetry and reflecting boundaries. The nuclide inventory of the depleted FAs was obtained from the CASMO5 output files, hence one criticality calculation was carried out for each burnup simulation. The nuclear data employed by OpenMC was ENDF/B-VII.1 [14] (formatted ACE files distributed by OpenMC developers), and was used without perturbations. Each OpenMC simulation required approximately 6 hours, and was carried out with $2 \cdot 10^5$ neutron histories for 400 batches (of which the first 100 were inactive), to obtain a statistical uncertainty $\sigma_{stat} = 10$ pcm. For the criticality calculations a RAM of 32 GB was used.

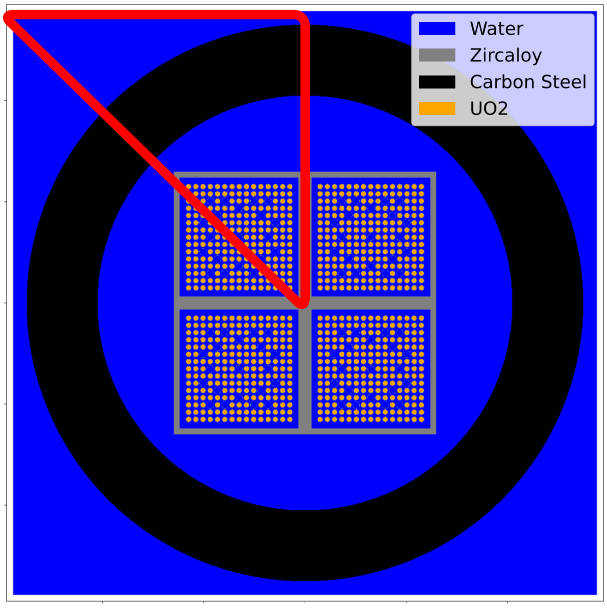


Figure 1: Canister design employed in the OpenMC criticality calculations, with shape based on the design from [16]. The inner and outer radii of the canister are 41 cm and 55 cm respectively, and the four SNF assemblies are identical. To speedup calculations, an 8-fold symmetry simulation was assumed, hence only the part in the red triangle was modelled, with reflecting boundary conditions.

2.2. Monte Carlo Uncertainty Quantification

Consider a quantity of interest q that is given as an output of a calculation $q = f(x)$, where $x \in \mathbb{R}^d$ is a vector containing the d input parameters. Consider now that the input is a random variable X with a known mean and covariance matrix, from which input vectors can be sampled $x_1, x_2, \ldots$. Then the output is also a random variable $Q = f(X)$, and one wants to calculate the mean $\mathbb{E}[Q]$ and standard deviation $\sqrt{\mathrm{Var}[Q]}$ (i.e. the uncertainty) of the quantity of interest. In the case of this paper q could be the decay heat, nuclide concentrations, or k-effective, the calculation f is a burnup computation with CASMO5, or CASMO5 followed by a criticality calculation with OpenMC, and the vector x contains $d = 15\,557$ nuclear data parameters.

With simple Monte Carlo (MC), a set of input vectors $x_1, x_2, \ldots, x_N$ is sampled and used as input for N calculations. Then the sample mean and variance can be estimated from the outputs $q_1, q_2, \ldots, q_N$ with the usual sample estimators (eq. 1). With this method, the expected errors on μ_N and σ_N converge to zero with the number of samples as $\mathcal{O}(1/\sqrt{N})$, i.e. the method is unbiased and dimensionality independent. The disadvantage of MC is that the convergence is slow, and thus f needs to be executed a large number of times to obtain a small error in the estimations. As an example, in this work it was found that approximately $N = 1000$ was required to obtain a 1% error in the estimator σ_N for the decay heat.

$$\mu_N = \frac{1}{N}\sum_{i=1}^{N} q_i, \quad \sigma_N^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(q_i - \sum_{j=1}^{N}\frac{q_j}{N}\right)^2 . \tag{1}$$

2.3. Lasso Monte Carlo (LMC)

The LMC method, recently introduced in [7], uses a combination of machine learning (ML) and multilevel Monte Carlo (MLMC), to estimate the mean and variance of Q. The following paragraphs briefly explain how ML and MLMC can each be used for UQ, before finally introducing the LMC method and how it combines these two techniques. All of the following methods use the same data as simple MC, i.e. the N input-output samples $x_1, x_2, ..., x_N$ and $q_1, q_2, ..., q_N$, and an additional set of M inputs $z_1, z_2, ..., z_M$, where it is assumed that $M \gg N$.

In recent years there has been a trend in the use of ML to reduce computational costs in methods that use repeated calculations. With this approach, the set of N input-output pairs is used to train a fast surrogate model $\widetilde{f}$. Then, $\widetilde{f}$ is used for the other M evaluations to obtain the outputs $\widetilde{q}_1, \widetilde{q}_2, ..., \widetilde{q}_M$, which are used for the usual estimators (eq. 1). The bottleneck of this approach is evaluating f N times, since training the surrogate model and evaluating it M times has a negligible cost. Therefore, this method is clearly only advantageous if $N < M$. However, since ML techniques suffer from the curse of dimensionality, the required size of the training set increases exponentially with the input dimension d, and any model trained with $N < d$ will be heavily biased, and produce biased estimates of the mean and variance. In the case of nuclear data, the input dimension is $d > 10^4$, hence using $N = 1000$ as in the simple MC case will lead to poor results. In conclusion, using ML for UQ, when the input dimension is large, has a higher computational cost than simple MC.

In the remaining part of the paper it will be assumed that $\widetilde{f}$ is a Lasso surrogate model [8]. While it is possible, in principle, to use any other ML technique for the methods explained here, Lasso is a natural choice in cases where the training set is small $N \ll d$, which is the case in all the examples in this paper. The Lasso is a simple sparse linear model, i.e. it fits a model of type $\widetilde{f}(x) = \beta \cdot x + b$, where the weight vector β is sparse. The method of fitting the model (coordinate descent) ensures that, the weights of β corresponding to input variables with small correlations to the output, are set to zero.

The MLMC [9, 17] approach is similar to using ML, but it produces unbiased results. Here, the N input-output samples are split into a training set of size N_{tr} and an evaluation set of size N_{eval}, with $N_{tr} + N_{eval} = N$. The training set is again used to train a surrogate $\widetilde{f}$, which is used to evaluate the M and N_{eval} samples. Then, all of the evaluations of f and $\widetilde{f}$ are used together in the *multilevel estimators* (eq. 2). These estimators are unbiased and converge at a rate $\mathcal{O}\left(1/\sqrt{N_{eval}}\right)$. Despite MLMC being unbiased, there is no guarantee that for a given sample size N it will be more or less accurate than simple MC, and its accuracy largely depends on the choice of N_{tr} and N_{eval}.

$$\begin{aligned}
\mu_{N_{eval},M} &= \frac{1}{M}\sum_{i=1}^{M}\widetilde{f}(z_i) + \frac{1}{N_{eval}}\sum_{i=1}^{N_{eval}}\left[f(x_i) - \widetilde{f}(x_i)\right] , \\
\sigma^2_{N_{eval},M} &= \frac{1}{M-1}\sum_{i=1}^{M}\left(\widetilde{f}(z_i) - \sum_{j=1}^{M}\frac{\widetilde{f}(z_j)}{M}\right)^2 \\
&+ \frac{1}{N_{eval}-1}\sum_{i=1}^{N_{eval}}\left[\left(f(x_i) - \sum_{j=1}^{N_{eval}}\frac{f(x_j)}{N_{eval}}\right)^2 - \left(\widetilde{f}(x_i) - \sum_{j=1}^{N_{eval}}\frac{\widetilde{f}(x_j)}{N_{eval}}\right)^2\right] .
\end{aligned} \tag{2}$$

The LMC method can be seen as a variation of MLMC, that removes the need to optimally balance the training and evaluation sets, and guarantees that its estimates will be equally or more accurate than simple MC. Here the N input-output samples are split into S subsets of equal size N/S. Then for each subset a surrogate is trained and the multilevel estimates are calculated $\mu_{N/S,M}$ and $\sigma_{N/S,M}$ with eq. 2. The LMC estimators (eq. 3) are simply the mean of these S multilevel estimates. In the LMC paper [7] it is shown that, under mild conditions on f, if the surrogate is a Lasso model and $M \gg N$, then the LMC estimators $\mathscr{M}_{N,M}$ and $\Sigma_{N,M}$ are equally or more accurate than simple MC for a given N. Therefore, to obtain the same error in the estimates as simple MC, LMC will require less expensive evaluations of f. The full LMC method is shown in Algorithm 1, and table 1 compares the four explained methods.

Algorithm 1 The Lasso Monte Carlo Algorithm

```
Require: The training sets $\{x_1,...x_N\}$ and $\{z_1,...z_M\}$, and model $f : \mathbb{R}^d \mapsto \mathbb{R}$.
Compute $f(x_1),...,f(x_N)$.
Split the $N$ input-output samples into $S$ subsets of size $\frac{N}{S}$ each. The subsets are $s_1,s_2,...,s_S$.
for $p = 1 \dots S$ do
    The p-th subset is the evaluation set, containing the samples $s_p = \{x'_1,x'_2,...,x'_{N/S}\}$.
    Make a training set of size $N\frac{S-1}{S}$ by joining the other subsets $s_1 \cup s_2 \cup ... \cup s_{p-1} \cup s_{p+1} \cup ... \cup s_S$.
    Fit a Lasso model $\widetilde{f}_p$ to the training set.
    Compute $\widetilde{f}_p(z_1),\widetilde{f}_p(z_2),...,\widetilde{f}_p(z_M)$, and $\widetilde{f}_p(x'_1),\widetilde{f}_p(x'_2),...,\widetilde{f}_p\left(x'_{N/S}\right)$ .
    Combine the $N/S$ evaluations of $f$ and $\widetilde{f}_p$ with the $M$ evaluations of $\widetilde{f}_p$ to compute the two-level estimators $(\mu_{N/S,M})_p$ and $(\sigma^2_{N/S,M})_p$ (eq. 2).
end for
Compute the LMC mean and variance, by averaging out the estimations of each split (eq. 3).
```

$$\mathscr{M}_{N,M} = \frac{1}{S}\sum_{p=1}^{S}\left(\mu_{N/S,M}\right)_p , \quad \text{and} \quad \Sigma^2_{N,M} = \frac{1}{S}\sum_{p=1}^{S}\left(\sigma^2_{N/S,M}\right)_p . \tag{3}$$

	Monte Carlo	Machine Learning	Multilevel Monte Carlo	Lasso Monte Carlo
Split	No	No	Training and evaluation sets $N_{tr} + N_{eval} = N$	S subsets of size N/S
Strategy	Compute μ_N and σ_N	Train $\widetilde{f}$, then evaluate M samples and compute μ_M and σ_M	Train $\widetilde{f}$, then compute multilevel estimators $\mu_{N_{eval},M}$ and $\sigma_{N_{eval},M}$	For each subset, train $\widetilde{f}$, then compute $\mu_{N/S,M}$ and $\sigma_{N/S,M}$. Average out the estimations to compute $\mathscr{M}_{N,M}$ and $\Sigma_{N,M}$
Error	Unbiased error$_{MC} \sim \frac{1}{\sqrt{N}}$	Biased and *curse of dimensionality*	Unbiased	Unbiased error$_{LMC}$ $\leq$error$_{MC}$

Table 1: Summary of the discussed UQ methods. Each method starts with a set of N input-output samples, and M input samples, with the assumption $M \gg N$.

2.4. Implementation of LMC

The LMC implementation is available in a GitLab repository [18]. It is written in Python and makes use of standard libraries. The user can choose any surrogate model for the LMC method, but for this paper only the `Lasso` Sklearn [19] implementation was used. The sparsity of β in the Lasso model is controlled by a regularisation parameter λ (see [8], referred to as `alpha` in the Sklearn implementations), which needs to be selected by the user. It can be chosen via model selection strategies such as cross-validation or BIC, with all strategies leading to similar results. In this paper, λ was chosen such that the number of nonzero elements of β would be the closest integer to $0.95 \cdot N_{tr}$, for a given number of training samples N_{tr}.

2.5. Simulated Assemblies

Two representative assemblies of UO2 and MOX fuel were used in this work, and the studied quantities of interest where the decay heat at cooling times of 2 and 50 years, and the concentrations of ^{235}U, ^{239}Pu, and ^{137}Cs at EOL of the assembly. Furthermore, for the UO2 assembly, the k-effective of a canister containing the depleted fuel at EOL (fig. 1) was also computed.

The UO2 assembly is a 15 × 15 FA from the Swedish Ringhals-2 pressure water reactor (PWR), with an enrichment of 3.095% ^{235}U, and a final reported burnup of 35.7 MWd/kgU [20]. This FA was part of a decay heat measurement campaign, and was used as a benchmark for a CASMO validation study [11].

The MOX example is a 14 × 14 FA from a Swiss PWR, with an average concentration of 3.6% fissile plutonium, and a final reported burnup of 58.9 MWd/kgU [21]. This FA has been analysed and compared to simulations in [22].

3. RESULTS

This section shows the results of UQ calculations with the described methods (table 1). In the following plots, and for the remainder of the paper, the UQ methods use $M = 6000$ to ensure $M \gg N$. Additionally, the MLMC method uses $N_{tr} = 0.8 \cdot N$, and LMC uses $S = 5$ (see table 1 and alg. 1).

Figure 2 shows 20 independent UQ calculations of the decay heat at 2 years of cooling, with simple MC and LMC, with respect to the number of CASMO5 simulations N. The uncertainty is due to nuclear data perturbation of the aforementioned 15 557 quantities. It is clear that LMC converges faster, and that for $N > 200$, the estimations become stable, whereas the simple MC method still shows large oscillations in its estimations up to $N = 1000$. Only LMC and simple MC are compared in this case to allow for easier readability. A comparison between all four UQ methods can be found in figure 3, where it can be seen that LMC is more accurate than all other methods for any N.

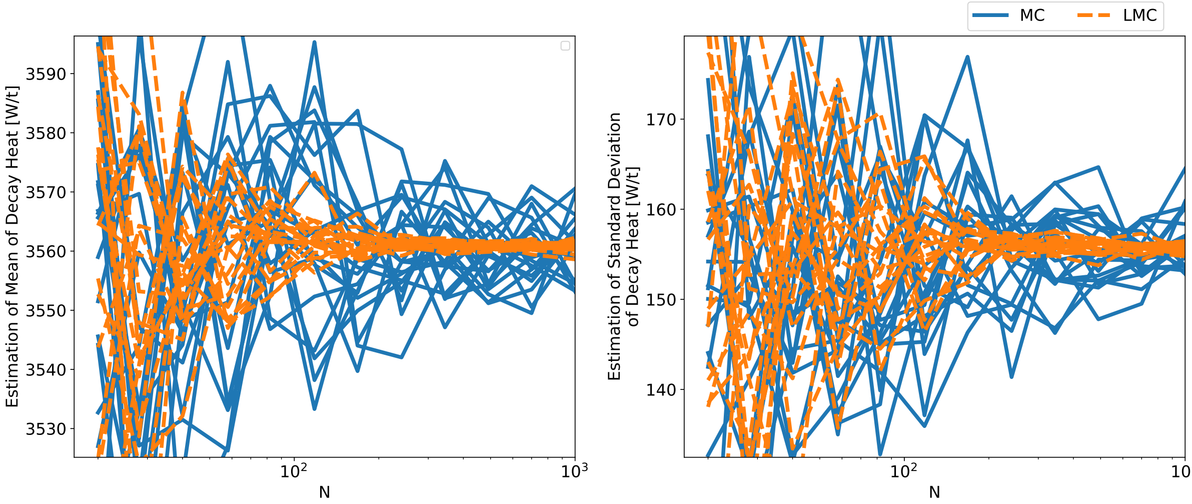


Figure 2: Estimation of mean and standard deviation of the decay heat of the UO2 assembly after 2 years of cooling. Each line is an independent UQ calculation with increasing N, and there are 20 lines for each method.

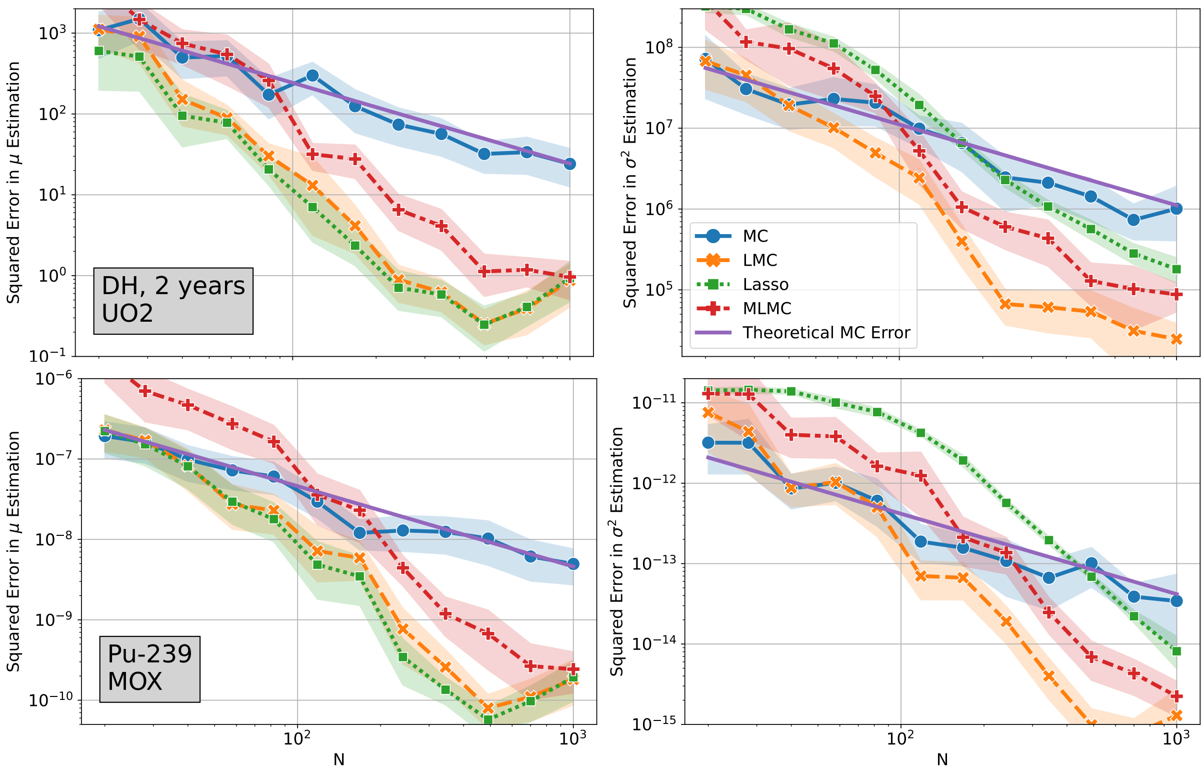
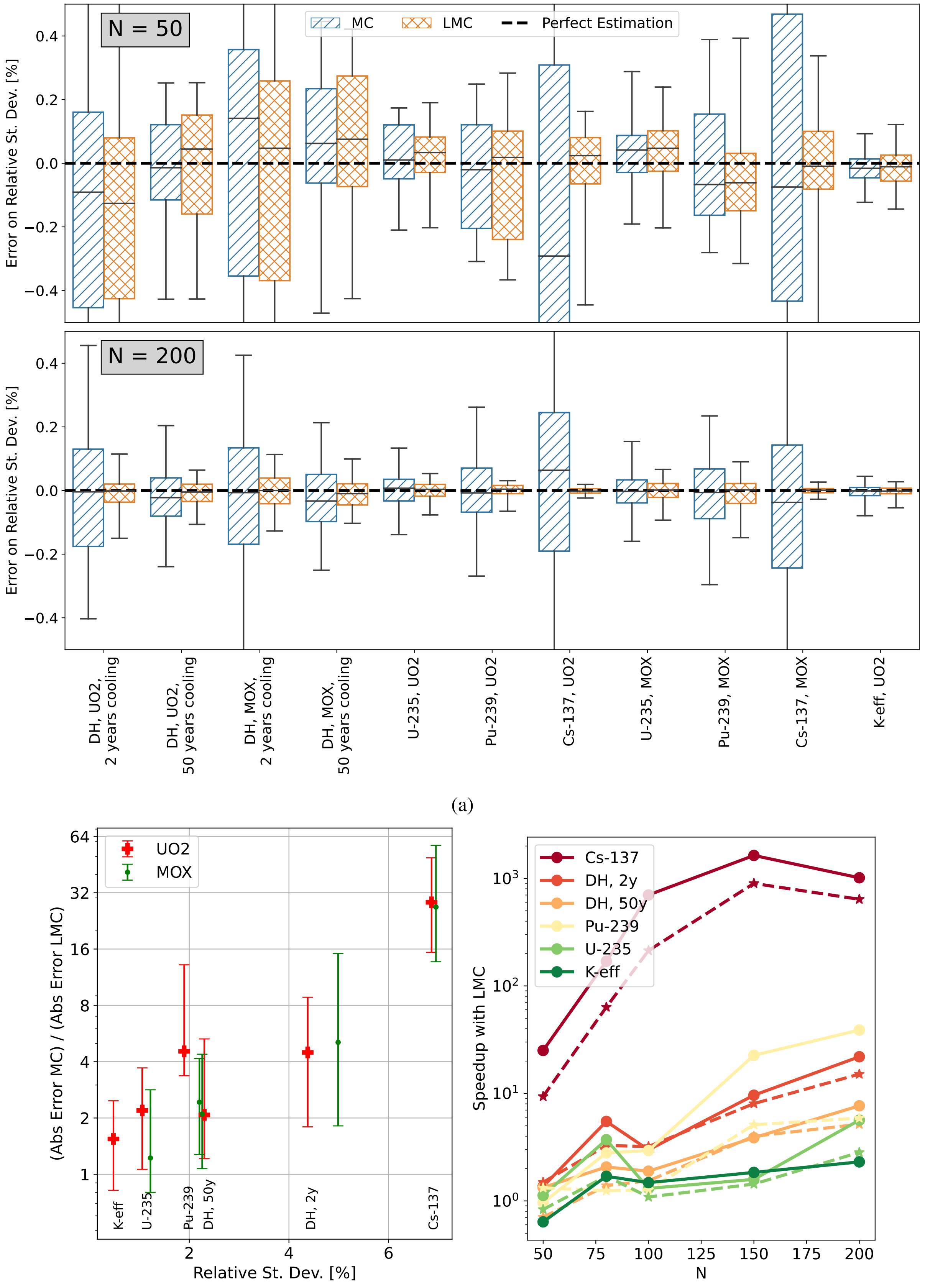


Figure 3: Convergence of the squared error, in the estimation of the mean and variance of the outputs. The studied quantities are the decay heat at 2 years of cooling for UO2, and the concentration of ^{239}Pu in MOX fuel at its EOL. Each line is the mean squared error of 20 independent UQ calculations, and the halos represent the 95% confidence intervals.

Figure 4a shows the errors with simple MC and LMC for several quantities and $N = 50, 200$, when estimating the relative standard deviation $100 \cdot \frac{\sigma}{\mu}$. The performance of both methods varies extensively among the predicted quantities, but LMC is consistently equal or more accurate than simple MC. The quantities where MC has the largest errors are ^{137}Cs and the decay heat at short cooling times, and both of these quantities have the largest improvement in error when using LMC. This suggests that LMC is the most advantageous when MC has large errors, and this can be further observed in figure 4b. It shows the improvement in accuracy $\frac{\text{Absolute Error with MC}}{\text{Absolute Error with LMC}}$ with respect to the relative standard deviation. There appears to a be a positive correlation between these two quantities, which indicates that quantities which have the largest uncertainties due to nuclear data are the ones that benefit the most from LMC.

The computational speedup of LMC with respect to MC is plotted in figure 4c for each quantity. The speedup is measured as speedup$= \frac{N_{MC}}{N}$, where N_{MC} is the average number of samples required with simple MC, to achieve the same mean squared error that LMC achieves with N. Similar speedups are observed between the UO2 and MOX quantities, except for the ^{239}Pu, where the speedup is roughly 4 times smaller in the MOX case. The largest speedup is around 1000 for the concentration of ^{137}Cs, followed by speedups in the order of 10 for the decay heats and nuclide concentrations. The lowest speedup is in the k-effective, although this could be explained by the small effect of nuclear data on this quantity – since the nuclear data was only perturbed in the burnup calculations, the k-effective was only affected by the perturbations in the nuclide concentrations. If one were to perturb the nuclear data of the criticality calculations it is likely that the k-effective would have larger variations and hence a stronger speedup from LMC.



(a)

(b)

(c)

Figure 4: (a) Errors of 20 estimations with MC and with LMC. The boxes represent the interquartile range (50% of samples fall in this range), while the whiskers represent the minimum and maximum. The line inside the boxes is the median of the errors. (b) Improvement in accuracy when using LMC versus MC, for $N = 200$. The markers are the medians of 40 UQ calculations, and the errorbars are the interquantile ranges. (c) The mean speedup with LMC versus MC. Continuous lines are for the UO2 calculations, and dashed lines for the MOX ones.

4. DISCUSSION AND CONCLUSION

The UQ problem of propagating the uncertainty of nuclear data is challenging due to its high-dimensionality. The best method currently in use is simple MC, which requires a large number N of costly simulations. In this paper the novel method LMC [7] was applied to this problem, as well as two other methods, multilevel Monte Carlo and Lasso machine learning.

The LMC method is the most accurate of the four methods, and it is the only one that is equal or better than simple MC for any N. Although LMC is consistently more accurate than MC, its exact improvement depends largely on the studied quantity. The quantities with the largest improvement from LMC are the concentration of ^{137}Cs and the decay heat at 2 years of cooling. Experiments suggest that quantities that show the strongest improvement from using LMC are those which are the most affected by nuclear data perturbations, i.e. those with the largest uncertainty. For the prediction of k-effective, LMC and MC performed similarly, with negligible differences. The computational speedups due to LMC are up to 1000 for predicting the concentration of ^{137}Cs, and in the order of 10 for most other quantities. Similar results and speedups were found between the UO2 and MOX cases.

Future work will further explore the applications of LMC to other quantities that characterise SNF. Efforts will go towards a better prediction of the k-effective uncertainty at different cooling times, and the perturbation of nuclear data will be included to criticality calculations, not only the burnup simulations.

ACKNOWLEDGEMENTS

This work was part of the COLOSS and COLOSS-2 projects (COmbined Loading Optimization with Simulations and Surrogate models), funded by Swissnuclear.

REFERENCES

[1] D. Rochman, A. Vasiliev, H. Ferroukhi, T. Zhu, S. C. van der Marck, and A. J. Koning. Nuclear data uncertainty for criticality-safety: Monte Carlo vs. linear perturbation. *Annals of Nuclear Energy*, 92:150–160, June 2016.

[2] V. Solans, D. Rochman, H. Ferroukhi, A. Vasiliev, and A. Pautz. Loading optimization for Swiss used nuclear fuel assemblies into final disposal canisters. *Nuclear Engineering and Design*, 370:110897, December 2020.

[3] B. Ebiwonjumi, A. Cherezov, S. Dzianisau, and D. Lee. Machine learning of LWR spent nuclear fuel assembly decay heat measurements. *Nuclear Engineering and Technology*, 53(11):3563–3579, November 2021.

[4] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When Is "Nearest Neighbor" Meaningful? In Catriel Beeri and Peter Buneman, editors, *Database Theory — ICDT'99*, Lecture Notes in Computer Science, pages 217–235, Berlin, Heidelberg, 1999.

[5] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, April 2016. Publisher: Royal Society.

[6] P. G. Constantine, E. Dow, and Q. Wang. Active subspace methods in theory and practice: Applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.

[7] A. Albà, R. Boiger, D. Rochman, and A. Adelmann. Lasso Monte Carlo, a Novel Method for High Dimensional Uncertainty Quantification. 2022. arXiv:2210.03634.

[8] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[9] M. B. Giles. Multilevel Monte Carlo Path Simulation. *Operations Research*, 56(3):607–617, June 2008.

[10] J. Rhodes, K. Smith, and D. Lee. CASMO-5 development and applications. Technical report, American Nuclear Society - ANS; La Grange Park (United States), July 2006.

[11] A. Shama, D. Rochman, S. Caruso, and A. Pautz. Validation of spent nuclear fuel decay heat calculations using Polaris, ORIGEN and CASMO5. *Annals of Nuclear Energy*, 165:108758, January 2022.

[12] W. Wieselquist, T. Zhu, A. Vasiliev, and H. Ferroukhi. PSI Methodologies for Nuclear Data Uncertainty Propagation with CASMO-5M and MCNPX: Results for OECD/NEA UAM Benchmark Phase I. *Science and Technology of Nuclear Installations*, 2013:1–15, 2013.

[13] A. Aures, F. Bostelmann, M. Hursin, and O. Leray. Benchmarking and application of the state-of-the-art uncertainty analysis methods XSUSA and SHARK-X. *Annals of Nuclear Energy*, 101:262–269, March 2017.

[14] M. B. Chadwick, M. Herman, P. Obložinský, M. E. Dunn, Y. Danon, A. C. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D. A. Brown, R. Capote, A. D. Carlson, Y. S. Cho, H. Derrien, K. Guber, G. M. Hale, S. Hoblit, S. Holloway, T. D. Johnson, T. Kawano, B. C. Kiedrowski, H. Kim, S. Kunieda, N. M. Larson, L. Leal, J. P. Lestone, R. C. Little, E. A. McCutchan, R. E. MacFarlane, M. MacInnes, C. M. Mattoon, R. D. McKnight, S. F. Mughabghab, G. P. A. Nobre, G. Palmiotti, A. Palumbo, M. T. Pigni, V. G. Pronyaev, R. O. Sayer, A. A. Sonzogni, N. C. Summers, P. Talou, I. J. Thompson, A. Trkov, R. L. Vogt, S. C. van der Marck, A. Wallner, M. C. White, D. Wiarda, and P. G. Young. ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data. *Nuclear Data Sheets*, 112(12):2887–2996, December 2011.

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

[16] A. Vasiliev, J. Herrero, M. Pecchia, D. Rochman, H. Ferroukhi, and S. Caruso. Preliminary Assessment of Criticality Safety Constraints for Swiss Spent Nuclear Fuel Loading in Disposal Canisters. *Materials*, 12(3):494, January 2019.

[17] S. Krumscheid, F. Nobile, and M. Pisaroni. Quantifying uncertain system outputs via the multilevel Monte Carlo method — Part I: Central moment estimation. *Journal of Computational Physics*, 414:109466, August 2020.

[18] GitLab-PSI: LassoMonteCarlo. `https://gitlab.psi.ch/albajacas_a/lassomontecarlo`, May 2023.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[20] F. Sturek, L. Agrenius, and O. Osifo. Measurements of decay heat in spent nuclear fuel at the Swedish interim storage facility, Clab. Technical Report R-05-62, Svensk Kärnbränslehantering AB, December 2006.

[21] RT Primm. ARIANE International Programme Final Report. Technical report, United States, May 2002. ORNL/SUB–97-XSV750-1 INIS Reference Number: 34070003.

[22] D. Rochman, A. Vasiliev, H. Ferroukhi, and M. Hursin. Analysis for the ARIANE BM1 and BM3 samples: nuclide inventory and decay heat. *EPJ Nuclear Sciences & Technologies*, 7:18, 2021.