Gaussian Processes for Surrogate Modeling of Discharged Fuel Nuclide Compositions

Antonio Figueroa[1], Malte Göttsche[1]

Nuclear Verification and Disarmament Group, RWTH Aachen University
Schinkelstraße 2a, 52062 Aachen, Germany

Abstract

Several applications such as nuclear forensics, nuclear fuel cycle simulations and sensitivity analysis require methods to quickly compute spent fuel nuclide compositions for various irradiation histories. Traditionally, this has been done by interpolating between one-group cross-sections that have been pre-computed from nuclear reactor simulations for a grid of input parameters, using fits such as Cubic Spline. We propose the use of Gaussian Processes (GP) to create surrogate models, which not only provide nuclide compositions, but also the gradient and estimates of their prediction uncertainty. The former is useful for applications such as forward and inverse optimization problems, the latter for uncertainty quantification applications. For this purpose, we compare GP-based surrogate model performance with Cubic-Spline-based interpolators based on infinite lattice simulations of a CANDU 6 nuclear reactor using the `SERPENT 2` code, considering burnup and temperature as input parameters. Additionally, we compare the performance of various grid sampling schemes to quasirandom sampling based on the Sobol sequence. We find that GP-based models perform significantly better in predicting spent fuel compositions than Cubic-Spline-based models, though requiring longer computational runtime. Furthermore, we show that the predicted nuclide uncertainties are reasonably accurate. While in the studied two-dimensional case, grid- and quasirandom sampling provide similar results, quasirandom sampling will be a more effective strategy in higher-dimensional cases.

Email addresses: `figueroa@aices.rwth-aachen.de` (Antonio Figueroa), `goettsche@aices.rwth-aachen.de` (Malte Göttsche)

Keywords: Gaussian Process Regression, Surrogate Modeling, Quasirandom Sampling, Reactor Simulations, Spent Fuel Compositions,

1. Introduction

The computation of spent fuel nuclide compositions is a complex problem which involves the modeling of a nuclear reactor fuel assembly geometry and tracking the nuclide evolution as the fuel is irradiated with an operational history indicated by different parameter such as burnup, power level, and temperature. This involves the numerical solution of the neutron transport equation through probabilistic or deterministic methods [1], which are computationally expensive.

For several applications however, it is desirable to have a fast and computationally less intensive method to compute nuclide compositions, especially when many repeated calculations are needed for further analysis. An example of this are nuclear fuel cycle simulators [2], nuclear forensics applications [3], and sensitivity analyses [4].

This problem has been addressed in the past through pre-computed databases of reactor simulations, and the interpolation of either the one-group cross-sections [5] [6] or nuclide concentrations resulting from these simulations with different methods such as: Nearest Neighbors, Linear Fit, Lagrange Polynomials, Neural Networks [7], and Cubic Splines, the latter being currently the most commonly used.

Up until now, most of the available software packages have used infinite lattice reactor simulations sampled on a multi-dimensional grid of at most three dimensions [5][6]. For low dimensional problems ($d \leq 3$), grid sampling produces good results, however for a larger number of dimensions, this sampling method does not distribute the sampling points efficiently across space [8], resulting in poor exploration of the parameter space. Quasirandom Sampling methods would provide better space coverage properties at higher dimensions [8]. However, some of the above mentioned interpolation methods require the samples to be distributed on a grid in order to perform correctly. Furthermore, while the interpolation quality can be estimated, the aforementioned interpolation methods do not provide information on the expected variance of the interpolation at non-sampled points.

Here, we propose using Gaussian Processes (GP) for interpolation, as Quasirandom Sampling can be used. Furthermore, also the gradient and estimates of their prediction uncertainty at non-sampled points are directly obtained. The former is useful for applications such as forward and inverse optimization problems, the latter for uncertainty quantification applications.GP belong to a set of tools used in the Machine Learning communities for a variety of tasks, including classification and regression. With Gaussian Process Regression (GPR), the interpolation is not performed on a specific function but over an infinite distribution of functions that share common properties as defined by the user.

An equivalent concept to GPR known as kriging, is well known in the field of geology [9]. Recently, researchers have used GP-based surrogate models for nuclear engineering applications such as modeling of equipment degradation and preventive maintenance [10], study of fuel performance and thermo-hydraulics [11]. They also have been used for the prediction of fuel nuclide composition and compared to surrogate models based on Dynamic Mode Decomposition, by performing regression on a Principal Component Analysis (PCA) model of the fuel isotopics [12]. Additionally, we have started exploring their use for the direct prediction of spent fuel composition, without a PCA-reduced model and using multidimensional input variables [13].

In this paper, we compare the performance of GP-based surrogate models to models based on Cubic Splines for the direct interpolation of spent fuel nuclide compositions (GP could also interpolate cross-sections). We compare to Cubic Splines as they often produce smoother and higher quality interpolators in comparison to methods such as Langrange and Newton polynomials [14]. The performance of both techniques will be explored over different experimental sampling configurations using both Grid sampling and Sobol Quasirandom Sequences in order to assess the impact of the sampling strategy on the regressions. We study two-dimensional problems to examine whether GPR performs well already in problems where grid sampling is still effective.

2. Creating the datasets

2.1. Reactor simulations

For our research, we have implemented a 2D infinite lattice model of a Candu 6 reactor based on specifications from available literature [15]. The implementation has been made in the computer code `SERPENT 2` which couples a Monte Carlo neutron transport module with a fuel depletion solver based on the Chebyshev Rational Approximation Method [16]. The quality

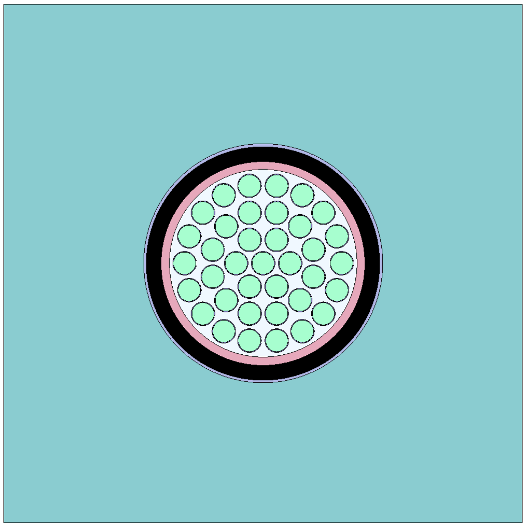

Figure 1: CANDU 6, 37-element fuel assembly, 2D infinite lattice implementation in `SERPENT 2` (light green indicates the fuel elements, white for the coolant, pink for the calandria and pressure tubes, black for the void between these tubes and turquoise for the moderator between fuel channels)

of the model has been examined by comparing end-of-cycle isotopic compositions with the `Bruce-1` dataset reported in the SFCOMPO-2.0 database [17]. Figure 1 shows the CANDU 6 37-elements fuel assembly implemented in `SERPENT 2`.

Parameters	**Range**
Moderator Temperature	333 - 363 K
Burnup	0.1 - 7 $\frac{\mathrm{MWd}}{\mathrm{kg_{HM}}}$

Table 1: Parameter ranges used in in the generation of the training and testing datasets

2.2. Sampling strategies

Strategies are required to sample sets of input parameters for each Serpent simulation involved in model development and testing. In this study, we sample moderator temperature and discharge burnup. Table 1 indicates the range of values considered for these parameters.

As mentioned, grid sampling performs poorly in higher-dimensional input spaces. While a random number sequence can overcome this issue in

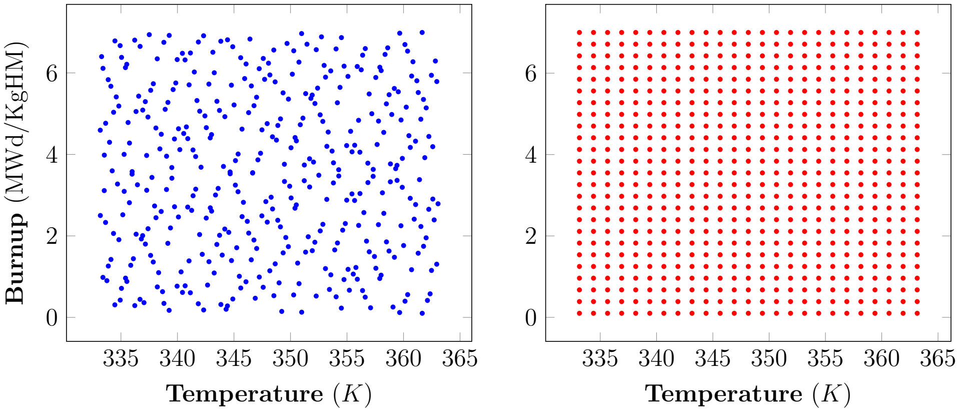

Figure 2: Comparison between the dataset based on Sobol quasirandom sampling (left) to the dataset based on Grid sampling (right). While both the Sobol sequence and grid sampling are effective in two dimensions as seen here, the Sobol sequence by far outperforms grid sampling in higher dimensions, at the same number of samples.

principle, due to the nature of pseudo-random number generators implementations, random number sequences tend to cluster, resulting in a non-uniform distribution of samples across the input space. In such cases, a quasirandom sequence can provide a set of samples with a better spatial distribution.[18]

Sobol quasirandom sampling is a method to generate quasi-random sequences which is designed to minimize the star discrepancy, namely, the difference between the distribution of values generated to a multidimensional uniform probability distribution. The generation of the samples involves using a special algorithm in which bitwise operations are performed. Details on the algorithm and its implementation can be found on [19].

In order to evaluate the impact of the sampling method on the prediction quality of the models, two datasets were simulated using the parameter ranges of Table 1, the first consisting of 625 samples created from a 2D 25x25 grid, and the second of 625 samples generated from a Sobol sequence sampler written in-house. Figure 2 shows a comparison between the samples generated via the Sobol sequence and the aforementioned grid. While grid sampling is still effective in two dimensions, the good space coverage of the Sobol sequence is evident.

To study the impact of the spatial arrangement of the training data on the model's quality assessed from the test data, both taken from the first dataset, we have designed three setups consisting of 25x25 grids with varying spacing

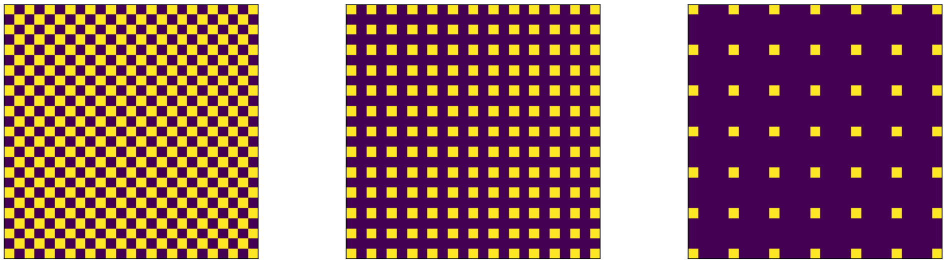

Figure 3: Spatial distribution patterns used to evaluate the performance of the surrogate models on a grid. From left to right the images correspond to the following patterns: Checkerboard, Grid - Spacing 2, Grid - Spacing 4. The yellow elements represent the simulations to be used for training the interpolators, while the dark regions represent the cells where the values will be interpolated.

and patterns (see Figure 3). In the case of interpolation using the second dataset, the training set will consist of the samples generated by the Sobol sequence, while the test set corresponds to the entire first dataset done on the 25x25 grid. This approach facilitates an internal comparison within the grid sampling schemes and an external comparison between the Sobol and Grid sampling strategies (see Table 2).

Configuration	**Training set size**	**Test set size**
Grid - Spacing 2*	169	456
Grid - Spacing 4*	49	576
Checkerboard	313	312
Sobol	625	625

* Spacing refers to the horizontal and vertical spacing between sampling points in the grid

Table 2: Grid pattern details. The different configurations allow for the study of model performance under varying spatial sample distributions and number of training points

3. Building the model

Based on the pre-calculated training data, we built the Cubic Spline and the GPR models. The former consists in fitting a piecewise cubic polynomial between the sampled points [14].

We have used the `SmoothBivariateSpline` module from the `Scipy` python package. The latter is discussed in the following.

3.1. Gaussian Process Regression

GP is a set of random variables that share a joint Gaussian probability distribution. GP's can be used for the construction of probabilistic surrogate models of black-box problems where an analytical form is unavailable or intractable and the computational cost is elevated. Without loss of generalization, the aim of GPR is to approximate a function $Y(X)$, $X = (x_1, ..., xi)$ through a GP as:

$$Y(X) \approx \mathcal{GP}(X) = \mathcal{N}(0, K(X)) \tag{1}$$

The kernel (K) of a GP is a function which describes the covariance between the inputs and outputs of the target function. It encodes a limited set of assumptions about the underlying function such as smoothness and differentiability. By choosing a kernel, an infinite set of functions which share the kernel properties are used to perform the regression, and by training the kernel parameters based on the input and output data a subset of those functions are chosen that match the data. We chose the Anisotropic Squared Exponential (ASE) kernel, which provides for very smooth interpolation and is infinitely differentiable, as we expect that the change of nuclide concentrations throughout the parameter space would meet these characteristics. An additional advantage is that the ASE kernel allows for the determination of relative input parameter relevance through the use of different correlation lengths parameters (ℓ_i) for each input:

$$K(x, z) = exp\left(-\sum_i^d \left(\frac{x_i - z_i}{\ell_i}\right)^2\right) \tag{2}$$

The smaller ℓ_i, the more sensitive the underlying model is to changes in input x_i. Once the parameters of the kernel that reproduce the training data are estimated, the posterior predictive distribution of the GP - our model - at an unseen point $(*)$ is given by a normal distribution with prediction mean (μ_*), prediction variance (σ_*^2) and prediction gradient $(\nabla\mu_*)$:

$$\begin{aligned}
\mu_* &= K_{train,*}(K_{train,train})^{-1}Y_{train} \\
\sigma_*^2 &= K_{*,*} - K_{*,train}^T(K_{train,train})^{-1}K_{*,train} \\
\nabla\mu_* &= \nabla K_{train,*}(K_{train,train})^{-1}Y_{train}
\end{aligned} \tag{3}$$

To implement the GPR models, we have used the `scikit-learn` Python package [20]. More details on GP's and its kernels can be found in [21].

3.2. Cross-validation

Since the kernel training process is strongly dependent on the training data, weak prediction performance can occur if an inappropriate selection of the training set is made. This can be avoided by cross-validation.

It is implemented by splitting the training set into k "folds" of approximately equal size and performing the training on each fold, thus obtaining the model parameters, then using the other folds as test data to evaluate the model predictive quality.

This should be performed several times by randomizing the selection of the folds, thus generating a set of plausible model parameters from which the best performing combinations can be selected. Notwithstanding this, a major benefit of cross-validation is that it allows the training of models with different combinations of samples spanning the entire input space, providing parameters sets that tend to enhance the model generalization properties, typically at a smaller computational cost since the training set size is reduced.

Each `SERPENT 2` simulation provides an output vector containing about 1300 nuclides. We have created GPR models for each of these nuclides for both datasets. Each GPR model has been trained using a 5-fold cross validation scheme that has been repeated 10 times, thus generating a set of 50 kernel parameter combinations from which the best-performing is chosen. The ^{239}Pu GPR model based on the Sobol sequence is shown in Fig. 4.

4. Results and Discussion

We have compared GPR based models of spent fuel nuclide concentrations to Cubic Splines models of these quantities. The Splines models have been implemented using the `SmoothBivariateSpline` module from the `Scipy` [22] python package. The following comparison is focused on two main elements: Runtime Performance and Model Quality.

4.1. Runtime performance

We have studied the mean time required to perform an interpolation for the interpolation methods discussed in this article. The tests have been performed on a 2.2 GHz `Intel Core i7` computer with 4 cores and 16GB of ram. Table 3 shows the regression time required averaged over all the models, with the time for a GP model separated into the the time required for the prediction **GPR**$-\mu$ and the time required for the estimation of the

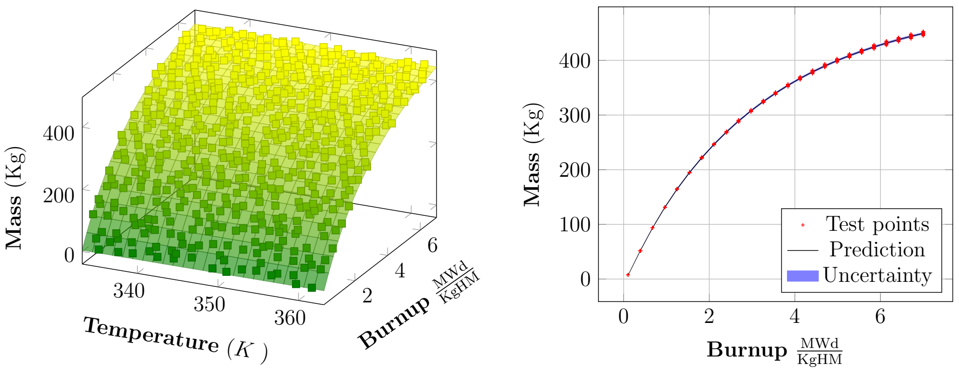

Figure 4: Reconstruction plots for ^{239}Pu. The left plot shows the GPR model based on samples generated by the Sobol sequence, showing the mass as a function of burnup at a temperature. The reported masses have been scaled by the number of fuel assemblies in the reactor and their dimensions. The squares indicate the samples obtained from the Sobol sequence. The plot on the right shows the same model at a temperature of 348.15 K. The predicted uncertainty is shown at the 6-σ level. The test points stem from grid sampling and show the accuracy of the model.

prediction variance **GPR**$-\sigma$, as these calculations can be run independently of each other.

While in our implementation they are calculated sequentially, the values shown in this table indicate that the time required for prediction and variance estimation is similar, thus hinting at a potential total runtime reduction by a factor of 2. Nevertheless, interpolation using GP models can require significantly more time than interpolation based on Cubic Splines. It might be possible that through the use of more efficient implementations, the runtime difference between the two can be reduced in the future.

Configuration	**Cubic Spline**	**GPR**$-\mu$	**GPR**$-\sigma$	**GPR Total**
	[s]	[s]	[s]	[s]
Checkerboard	1.61e-04	4.47e-04	6.13e-04	1.06e-03
Grid - Spacing 2	3.40e-05	4.61e-04	6.00e-04	1.06e-03
Grid - Spacing 4	3.03e-06	3.64e-04	3.17e-04	6.82e-04
Sobol - Grid	9.74e-06	1.18e-03	9.95e-04	2.18e-03

Table 3: Run-time comparison for different configurations. Reported values are means over regressions performed on all isotopes, therefore the total time for a GPR prediction does not match with the sum of GPR components run-time.

4.2. Model Quality

We have analyzed the quality of the models based on GP's and Cubic Splines using the experimental designs presented in Table 2 and Figures 2 and 3. For this, we have considered an array of metrics to quantify model performance, namely the root-mean-square-errors ($RMSE$) and the coefficients of determination R^2. We also examine an additional metric related to the posterior predictive variance produced by GPR in order to study the quality of this estimator. This quantity is $Pred - 1\sigma$, which quantifies the fraction of model predictions located within 1 predictive standard deviations from the true values of the test sets, i.e. it shows whether the predicted variance represents the variance that is actually observed.

While models were created for each of the nuclides tracked by `SERPENT 2`, Tables 4, 5, A.6, and A.7 contain the results of our analysis for a small subset of them representative of both major actinides and fission products. The mean values shown have been scaled by the assembly length and number of fuel assemblies in the reactor core.

$Nuclide$	μ	Cubic Spline	GPR	RMSE $\frac{Spline}{GPR}$	R^2_{Sp}	R^2_{GPR}	$Pred - 1\sigma$
	Kg	RMSE	RMSE				%
^{85}Rb	2.06	$3.8 \cdot 10^{-3}$	$3.4 \cdot 10^{-3}$	1.12	1	1	74.4
^{90}Sr	11.55	$2.2 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$	1.18	1	1	75.3
^{106}Ru	4.34	$1.9 \cdot 10^{-2}$	$7.7 \cdot 10^{-3}$	2.45	1	1	76
^{110}Cd	0.12	$4.9 \cdot 10^{-4}$	$3.4 \cdot 10^{-4}$	1.44	1	1	85.9
^{135}Cs	2.58	$5.2 \cdot 10^{-3}$	$4.3 \cdot 10^{-3}$	1.19	1	1	75.3
^{137}Cs	24.48	$4.2 \cdot 10^{-2}$	$4.1 \cdot 10^{-2}$	1.02	1	1	77.2
^{137}Ba	0.17	$3.4 \cdot 10^{-4}$	$3.3 \cdot 10^{-4}$	1.02	1	1	99.4
^{143}Nd	16.52	$2.1 \cdot 10^{-1}$	$2.8 \cdot 10^{-2}$	7.69	1	1	74.4
^{147}Nd	0.85	$6.6 \cdot 10^{-2}$	$1.2 \cdot 10^{-3}$	51.31	0.76	1	64.7
^{148}Nd	7.56	$1.3 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	1.03	1	1	71.5
^{147}Sm	0.5	$1.1 \cdot 10^{-3}$	$9.9 \cdot 10^{-4}$	1.12	1	1	77.9
^{151}Sm	0.29	$1.3 \cdot 10^{-2}$	$4.6 \cdot 10^{-4}$	28.18	0.98	1	100
^{154}Eu	0.13	$3.0 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$	1.07	1	1	92.9
^{239}Pu	301.76	$9.9 \cdot 10^{-1}$	$4.9 \cdot 10^{-1}$	2.02	1	1	69.9
^{240}Pu	75.15	$5.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$	4.14	1	1	74.7
^{241}Pu	13.6	$5.4 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$	1.7	1	1	78.5
^{241}Am	0.11	$1.8 \cdot 10^{-3}$	$3.1 \cdot 10^{-4}$	5.93	1	1	79.5

Table 4: Comparison between GPR and Cubic spline models for a selection of nuclides. The models have been trained on a squared grid with a Checkerboard arrangement. Highlighted cells indicate smallest error in the comparison. RMSE values are in Kg

$Nuclide$	μ	Cubic Spline	GPR	RMSE $\frac{Spline}{GPR}$	R^2_{Sp}	R^2_{GPR}	$Pred - 1\sigma$
	Kg	RMSE	RMSE				%
^{85}Rb	2.06	$3.8 \cdot 10^{-3}$	$3.3 \cdot 10^{-3}$	1.14	1	1	73.6
^{90}Sr	11.55	$2.2 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$	1.21	1	1	73.4
^{106}Ru	4.34	$2.0 \cdot 10^{-2}$	$7.7 \cdot 10^{-3}$	2.59	1	1	75.5
^{110}Cd	0.12	$4.5 \cdot 10^{-4}$	$3.2 \cdot 10^{-4}$	1.4	1	1	90.9
^{135}Cs	2.58	$5.2 \cdot 10^{-3}$	$4.3 \cdot 10^{-3}$	1.21	1	1	75.4
^{137}Cs	24.48	$4.1 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$	1.03	1	1	96.3
^{137}Ba	0.17	$3.7 \cdot 10^{-4}$	$3.2 \cdot 10^{-4}$	1.13	1	1	99
^{143}Nd	16.52	$2.3 \cdot 10^{-1}$	$2.8 \cdot 10^{-2}$	8.13	1	1	70.9
^{147}Nd	0.85	$7.1 \cdot 10^{-2}$	$1.2 \cdot 10^{-3}$	57.35	0.72	1	64.8
^{148}Nd	7.56	$1.2 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	1.02	1	1	66.6
^{147}Sm	0.5	$1.1 \cdot 10^{-3}$	$9.8 \cdot 10^{-4}$	1.2	1	1	73.1
^{151}Sm	0.29	$1.4 \cdot 10^{-2}$	$4.6 \cdot 10^{-4}$	30.77	0.97	1	70.9
^{154}Eu	0.13	$2.8 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	1.06	1	1	99.7
^{239}Pu	301.76	$1.0 \cdot 10^{1}$	$5.9 \cdot 10^{-1}$	1.71	1	1	68.2
^{240}Pu	75.15	$6.1 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$	4.47	1	1	75.4
^{241}Pu	13.6	$5.8 \cdot 10^{-2}$	$3.3 \cdot 10^{-2}$	1.78	1	1	74.4
^{241}Am	0.11	$1.8 \cdot 10^{-3}$	$3.0 \cdot 10^{-4}$	6.01	1	1	85.6

Table 5: Comparison between GPR and Cubic spline models for a selection of nuclides. The models have been trained on samples generated from a 2-dimensional Sobol quasirandom sequence. Highlighted cells indicate smallest error in the comparison. RMSE values are in Kg

In general, we observe that based on the R^2 metric, both interpolation methods have good performance for most of the nuclides. Furthermore, we note that based on the $RMSE$ metric, GP based models perform better in almost all the experimental configurations for all isotopes. The improvement of the mean $RMSE$ is in most cases between a factor of 1.02 and 10. Still, Cubic Splines still provide reasonable results for most nuclides.

Two outliers are ^{147}Nd and ^{151}Sm, however, where the improvement is up to a factor of 57. Based on the R^2 metric, the respective Cubic Spline models did not model the data very well. Both of these nuclides have very large absorption cross-sections, especially in the thermal and epithermal regions of the energy spectrum, so small perturbations in the flux can potentially have large effects on the nuclide concentrations. We believe that in this case Cubic Splines has problems interpolating such perturbations in the model outputs. Interestingly we have observed that GPR also has problems interpolating ^{147}Nd when implemented over the grid with spacing of 4 units. We believe that in this case, the reason might be the low spatial density of the training design, resulting in a very smooth interpolation, missing the fluctuations in the nuclide concentration. In practice this can be solved by increasing the global sampling density or locally if prior information about the distribution of concentrations is known.

Also remarkable is the fact that the variance predictor of the GP model predicts the true values at the $1-\sigma$ level reasonably well, which is not generalizable to any GPR application. In addition, we have not observed a major difference between the use of Sobol sequences and grid sampling for the studied models, beyond the expected $RMSE$ error reduction with increased number and density of samples. However, we find that a Grid with a spacing of 2 units provides a good compromise between interpolation error and number of simulations in the training set. Nevertheless, we would expect the impact of Sobol sequences to be made manifest in problems of higher dimensionality.

5. Conclusion

We have compared two interpolation methods for direct fast prediction of spent fuel nuclide compositions. As expected, GP models typically require significantly more time to interpolate values. However, in return an estimation of the prediction uncertainty can be obtained, as well as an explicit method for estimating the gradient of the underlying function, provided the kernels used are differentiable. This can be useful for the solution of both forward and inverse optimization problems.

Additionally, we have noted that GP models result in smaller interpolation errors under the $RMSE$ metric, with varying reduction factors of up to 57 when compared to Cubic Spline interpolation. This varies depending on the experimental design used, however, a significant effect of using Sobol to other Grid sampling patterns was not observed. We expect however, that the use of Sobol sampling for higher dimensional problems would outperform the results obtained from Grid-based methods.

Several possibilities exist for the extension of this work. As of now, we have created GP models depending only on two input parameters using the ASE kernel. Future research could entail the inclusion of a larger number of parameters such as power, temperature or reactor downtime, as well as the use of different kernels and even combinations of kernels for more flexible models.

Higher fidelity GP models can be created by the use of 3-D full core reactor simulations, to account for spatial variation of nuclide concentrations and burnup, temperature and power levels. Building such GP-based models would, however, be significantly more computationally expensive.

As we have only studied GPR to directly predict isotopic compositions, one should study the predictive performance when implementing GPR on a cross-section level. This could enable implementing changes in the model parameters during the irradiation cycle. This approach could make feasible the forward uncertainty estimation via Monte Carlo methods, calculating the nuclide concentrations through the matrix exponential method, all thanks to the predictive variance of the interpolated one-group cross-sections.

In conclusion, we hope that the full potential of GP-based modelling of nuclear processes will be further studied and exploited in the future, having discussed here the many advantages it has for nuclear engineering applications.

Data Availability

The code used for the implementation of GP models and their comparison with Cubic Spline models can be obtained at: `https://github.com/FigueroaAC/GPs-for-SpentFuel`

Acknowledgments

This research has been funded by the Mathematics Division of the Center for Computational Engineering Science at RWTH Aachen University, and the Volkswagen Foundation. The funding sources had no role in the study design, collection, analysis and interpretation of data as well in the writing of the report. The calculations were run on RWTH Aachen's Compute Cluster under the rwth0503 project. Lastly, we would like to thank Manuel Torrilhon for his support.

References

[1] D. G. Cacuci, Handbook of Nuclear Engineering, Vol. 1, Springer Science & Business Media, Berlin Heidelberg, 2010, Ch. 5, pp. 532–540.

[2] K. D. Huff, M. J. Gidden, R. W. Carlsen, et al., Fundamental concepts in the cyclus nuclear fuel cycle simulation framework, Advances in Engineering Software 94 (2016) 46–59. doi:10.1016/j.advengsoft.2016.01.014.

[3] K. Dayman, S. Biegalski, Feasibility of fuel cycle characterization using multiple nuclide signatures, Journal of Radioanalytical and Nuclear Chemistry 296 (1) (2012) 195–201. doi:10.1007/s10967-012-1987-4.

[4] E. D. Kitcher, J. M. Osborn, S. S. Chirayath, Sensitivity studies on a novel nuclear forensics methodology for source reactor-type discrimination of separated weapons grade plutonium, Nuclear Engineering and Technology 51 (5) (2019) 1355–1364. doi:10.1016/j.net.2019.02.019.

[5] S. M. Bowman, L. C. Leal, ORIGEN-ARP: Automatic Rapid Process for Spent Fuel Depletion, Decay, and Source Term Analysis - NUREG/CR-0200 Revision 6, Vol. 1-D1, Oak Ridge National Laboratory, 2006.

[6] A. M. Scopatz, E. A. Schneider, A new method for rapid computation of transient fuel cycle material balances, Nuclear Engineering and Design 239 (10) (2009) 2169–2184. doi:10.1016/j.nucengdes.2009.02.022.

[7] B. Leniau, B. Mouginot, N. Thiolliere, et al., A neural network approach for burn-up calculation and its application to the dynamic fuel cycle code CLASS, Annals of Nuclear Energy 81 (2015) 125–133. doi:10.1016/j.anucene.2015.03.035.

[8] L. Diego, M. Pedergnana, G. Sebastián, Smart sampling and incremental function learning for very large high dimensional data, Neural Networks 78 (2016) 75–87. doi:10.1016/j.neunet.2015.09.001.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd Edition, Cambridge University Press, USA, 2007.

[10] P. Baraldi, F. Mangili, E. Zio, A prognostics approach to nuclear component degradation modeling based on gaussian process regression, Progress in Nuclear Energy 78 (2015) 141–154.

[11] X. Wu, T. Kozlowski, H. Meidani, K. Shirvan, Inverse uncertainty quantification using the modular bayesian approach based on gaussian process, part 2: Application to trace, Nuclear Engineering and Design 335 (2018) 417–431.

[12] R. Elzohery, M. Abdo, J. Roberts, Comparison between gaussian processes and dmd surrogates for isotopic composition prediction, in: 2018 ANS Annual Meeting: "Driving the Future of Nuclear Technology", 2018.

[13] A. Figueroa, M. Goettsche, Nuclear archaeology: Reconstructing reactor histories from reprocessing waste, ESARDA Bulletin 59 (2019) 39–46.

[14] R. L. Burden, J. D. Faires, Numerical Analysis, 9th Edition, Brooks/Cole, Cengage Learning, The address, 2010.

[15] W. J. Garland (Ed.), The Essential CANDU, A Textbook on the CANDU Nuclear Power Plant Technology, University Network of Excellence in Nuclear Engineering (UNENE), 2017.
URL https://www.unene.ca/education/candu-textbook

[16] J. Leppnen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho, The serpent monte carlo code: Status, development and applications in 2013, Annals of Nuclear Energy 82 (2015) 142–150.

[17] F. Michel-Sendis, et al., Sfcompo-2.0: An oecd nea database of spent nuclear fuel isotopic assays, reactor design specifications, and operating data, Annals of Nuclear Energy 110 (2017) 779–788.

[18] S. K. Sen, T. Samanta, A. Reese, Quasi- versus pseudo-random generators: Discrepancy, complexity and integration-error based comparison, International Journal of Innovative Computing, Information and Control 2 (3) (2006) 621–651.
URL http://www.ijicic.org/05-036-1.pdf

[19] J. Stephen, K. Frances, Constructing sobol sequences with better two-dimensional projections, SIAM J. Sci. Comput. 30 (2008) 2635–2654.

[20] F. Pedregosa, et al., Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[21] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, Massachusetts Institue of Technology, 2006.

[22] P. Virtanen, et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:https://doi.org/10.1038/s41592-019-0686-2.

Appendix A. Result Tables

$Nuclide$	μ	Cubic Spline	GPR	RMSE $\frac{Spline}{GPR}$	R^2_{Sp}	R^2_{GPR}	$Pred - 1\sigma$
	Kg	RMSE	RMSE				%
^{85}Rb	2.06	$3.9 \cdot 10^{-3}$	$3.4 \cdot 10^{-3}$	1.14	1	1	74.1
^{90}Sr	11.55	$2.3 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	1.2	1	1	73.7
^{106}Ru	4.34	$1.8 \cdot 10^{-2}$	$7.8 \cdot 10^{-3}$	2.37	1	1	75.2
^{110}Cd	0.12	$4.8 \cdot 10^{-4}$	$3.2 \cdot 10^{-4}$	1.49	1	1	82.5
^{135}Cs	2.58	$5.0 \cdot 10^{-3}$	$4.4 \cdot 10^{-3}$	1.15	1	1	75
^{137}Cs	24.48	$4.2 \cdot 10^{-2}$	$4.1 \cdot 10^{-2}$	1.02	1	1	98
^{137}Ba	0.17	$3.4 \cdot 10^{-4}$	$3.3 \cdot 10^{-4}$	1.02	1	1	97.6
^{143}Nd	16.52	$2.2 \cdot 10^{-1}$	$3.0 \cdot 10^{-2}$	7.32	1	1	66
^{147}Nd	0.85	$6.8 \cdot 10^{-2}$	$8.8 \cdot 10^{-3}$	7.77	0.67	0.99	65.4
^{148}Nd	7.56	$1.3 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	1.02	1	1	76.5
^{147}Sm	0.5	$1.1 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	1.11	1	1	77.2
^{151}Sm	0.29	$1.3 \cdot 10^{-2}$	$6.8 \cdot 10^{-4}$	19.89	0.97	1	98.9
^{154}Eu	0.13	$2.8 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	1.08	1	1	82.9
^{239}Pu	301.76	$1.0 \cdot 10^{1}$	$5.2 \cdot 10^{-1}$	1.95	1	1	63.2
^{240}Pu	75.15	$5.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$	4.08	1	1	74.3
^{241}Pu	13.6	$5.4 \cdot 10^{-2}$	$3.3 \cdot 10^{-2}$	1.64	1	1	74.1
^{241}Am	0.11	$1.8 \cdot 10^{-3}$	$3.1 \cdot 10^{-4}$	6.01	1	1	75.2

Table A.6: Comparison between GPR and Cubic spline models for a selection of nuclides. The models have been trained on a squared grid with a spacing of 2 units. Highlighted cells indicate smallest error in the comparison. RMSE values are in Kg

$Nuclide$	μ	Cubic Spline	GPR	RMSE $\frac{Spline}{GPR}$	R^2_{Sp}	R^2_{GPR}	$Pred - 1\sigma$
	Kg	RMSE	RMSE				%
^{85}Rb	2.06	$4.6 \cdot 10^{-3}$	$3.6 \cdot 10^{-3}$	1.26	1	1	69.8
^{90}Sr	11.55	$2.7 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	1.32	1	1	68.6
^{106}Ru	4.34	$2.1 \cdot 10^{-2}$	$8.7 \cdot 10^{-3}$	2.4	1	1	71.4
^{110}Cd	0.12	$5.7 \cdot 10^{-4}$	$3.4 \cdot 10^{-4}$	1.67	1	1	78.3
^{135}Cs	2.58	$5.5 \cdot 10^{-3}$	$4.6 \cdot 10^{-3}$	1.19	1	1	71.7
^{137}Cs	24.48	$4.7 \cdot 10^{-2}$	$4.4 \cdot 10^{-2}$	1.08	1	1	74.5
^{137}Ba	0.17	$4.0 \cdot 10^{-4}$	$3.6 \cdot 10^{-4}$	1.1	1	1	100
^{143}Nd	16.52	$2.7 \cdot 10^{-1}$	$8.2 \cdot 10^{-2}$	3.33	1	1	90.6
^{147}Nd	0.85	$8.7 \cdot 10^{-2}$	$7.8 \cdot 10^{-2}$	1.11	0.51	0.6	88.9
^{148}Nd	7.56	$1.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	1.08	1	1	73.3
^{147}Sm	0.5	$1.3 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	1.21	1	1	73.8
^{151}Sm	0.29	$1.7 \cdot 10^{-2}$	$8.4 \cdot 10^{-3}$	2.03	0.96	0.99	98.1
^{154}Eu	0.13	$3.4 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$	1.2	1	1	65.5
^{239}Pu	301.76	$1.7 \cdot 10^{1}$	$2.5 \cdot 10^{1}$	0.7	1	1	46.9
^{240}Pu	75.15	$6.7 \cdot 10^{-1}$	$1.8 \cdot 10^{-1}$	3.66	1	1	58.9
^{241}Pu	13.6	$6.0 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$	1.11	1	1	34
^{241}Am	0.11	$2.1 \cdot 10^{-3}$	$3.0 \cdot 10^{-4}$	7.02	1	1	76.9

Table A.7: Comparison between GPR and Cubic spline models for a selection of nuclides. The models have been trained on a squared grid with a spacing of 4 units. Highlighted cells indicate smallest error in the comparison. RMSE values are in Kg