AN OVERVIEW OF REDUCED ORDER MODELING TECHNIQUES FOR SAFETY APPLICATIONS

D. Mandelli[†], A. Alfonsi, P. Talbot, C. Wang, D. Maljovec, C. Smith, C. Rabiti, J. Cogliati

Idaho National Laboratory, 2525 North Fremont Street, Idaho Falls (ID)

[†] *Corresponding author: diego.mandelli@inl.gov*

The RISMC project is developing new advanced simulation-based tools to perform Computational Risk Analysis (CRA) for the existing fleet of U.S. nuclear power plants (NPPs). These tools numerically model not only the thermal-hydraulic behavior of the reactors primary and secondary systems, but also external event temporal evolution and component/system ageing. Thus, this is not only a multi-physics problem being addressed, but also a multi-scale problem (both spatial, µm-mm-m, and temporal, seconds-hours-years). As part of the RISMC CRA approach, a large amount of computationally-expensive simulation runs may be required. An important aspect is that even though computational power is growing, the overall computational cost of a RISMC analysis using brute-force methods may be not viable for certain cases. A solution that is being evaluated to assist the computational issue is the use of reduced order modeling techniques. During the FY2015, we investigated and applied reduced order modeling techniques to decrease the RISMC analysis computational cost by decreasing the number of simulation runs; for this analysis improvement we used surrogate models instead of the actual simulation codes. This article focuses on the use of reduced order modeling techniques that can be applied to RISMC analyses in order to generate, analyze, and visualize data. In particular, we focus on surrogate models that approximate the simulation results but in a much faster time (microseconds instead of hours/days).

I. INTRODUCTION

In the Risk Informed Safety Margin Characterization (RISMC) [1] approach, what we want to understand is not just the frequency of an event like core damage, but how close we are (or not) to key safety-related events and how might we increase the safety margin. A safety margin can be characterized in one of two ways:

- A deterministic margin, typically defined by the ratio (or, alternatively, the difference) of a capacity (i.e., strength) over the load
- A probabilistic margin, defined by the probability that the load exceeds the capacity. A probabilistic safety margin is a numerical value quantifying the probability that a safety metric (e.g., for an important process observable such as clad temperature) will be exceeded under accident scenario conditions.

The RISMC Pathway uses the probabilistic methods to determine safety margins and quantify their impacts to reliability and safety for existing Nuclear Power Plants (NPPs), i.e., pressurized and boiling water reactors (PWRs and BWRs). As part of the quantification, we use both probabilistic (via risk simulation) and mechanistic (via system simulators) approaches. Probabilistic analysis is represented by the risk analysis while mechanistic analysis is represented by the plant physics calculations. In the plant simulation, all the deterministic aspects that characterize system dynamics (e.g., thermo-hydraulic, thermo-mechanics, neutronics) are coupled to each other.

The risk simulation contains all deterministic elements that impact accident evolution (such as safety systems control logic and accident scenario initial and boundary conditions) in addition to stochastic ones (such as system/component failures and stochastic perturbation of internal elements of the physics simulation). The stochastic analysis [2] is performed in two steps: 1) sampling the stochastic parameters, and 2) evaluating the system response for the given set of sampled parameters.

In the RISMC applications, system simulator codes model not only plant thermo-hydraulic, thermo-mechanic, neutronic and ageing behavior but also model external event and human interactions with the plant itself. This is not only a multi-physics problem (i.e., different sets of equations are solved) but also a multi-scale one (i.e., both temporal and spatial scales). The drawback is that a single plant accident analysis (e.g., prediction of the seismic response of a BWR that underwent to a 60 years life extension license) might require long computational time that grows exponentially if multiple runs (through the sampling process) are needed.

In [3] we have focused our attention on the sampling strategies that we are employing. We initially employed classical sampling algorithms like Monte-Carlo, Grid and Latin Hypercube. In addition, we investigated more advanced sampling algorithms that aim to reduce the number of samples required to perform the desired stochastic analysis. We have shown how this reduction would allow the user to greatly reduce the computational costs of a typical RISMC analysis.

In this article, however we are focusing on how we can reduce the computational costs by more broadly employing Reduced Order Modeling techniques in typical RISMC-type analyses. We will show how reduced order modeling techniques can be applied to any RISMC analysis to generate, analyze and visualize data. In particular, we focus on surrogate models that approximate the simulation results but in a much faster time (microseconds instead of hours/days). We apply reduced order and surrogate modeling techniques to several RISMC types of analyses using RAVEN [4] and show the advantages that can be gained.

II. RISMC METHOD

The RISMC approach employs both deterministic and stochastic methods in a single analysis framework (see Figure 2). In the deterministic method set we include:

- Modeling of the thermal-hydraulic behavior of the plant [5]
- Modeling of external events such as flooding [6]
- Modeling of the operators responses to the accident scenario [7]

Note that deterministic modeling of the plant or external events can be performed by employing specific simulator codes but also surrogate models (see Section 5), known as reduced order models (ROM). ROMs would be employed in order to decrease the high computational costs of employed codes.

In addition, multi-fidelity codes can be employed to model the same system; the idea is to switch from low-fidelity to high-fidelity code when higher accuracy is needed (e.g., use low-fidelity codes for steady-state conditions and high-fidelity code for transient conditions). On the other hand, in the stochastic modeling we include all stochastic parameters that are of interest in the PRA analysis (such as uncertain parameters and stochastic failure of system/components).

As mentioned earlier, the RISMC approach heavily relies on multi-physics system simulator codes (e.g., RELAP5-3D [8]) coupled with stochastic analysis tools (e.g., RAVEN [4]). From a mathematical point of view, a single simulator run can be represented as a single trajectory in the phase space. The evolution of such a trajectory in the phase space can be described as follows:

$$\frac{\partial \boldsymbol{\theta}(t)}{\partial t} = \boldsymbol{\mathcal{H}}(\boldsymbol{\theta}, \boldsymbol{s}, t) \tag{1}$$

where:

- $\boldsymbol{\theta} = \boldsymbol{\theta}(t)$ represents the temporal evolution of a simulated scenario, i.e., $\boldsymbol{\theta}(t)$ represents a single simulation run
- $\boldsymbol{\mathcal{H}}$ is the actual simulator code that describes how $\boldsymbol{\theta}$ evolves in time
- $\boldsymbol{s} = \boldsymbol{s}(t)$ represents the status of components and systems of the simulator (e.g., status of emergency core cooling system, AC system)

By using the RISMC approach, the PRA analysis is performed by [9]:

1. Associating a probabilistic distribution function (pdf) to the set of parameters $\boldsymbol{s}$ (e.g., timing of events)
2. Performing stochastic sampling of the pdfs defined in Step 1
3. Performing a simulation run given $\boldsymbol{s}$ sampled in Step 2, i.e., solve Eq. (1)
4. Repeating Steps 2 and 3 M times and evaluating user defined stochastic parameters such as core damage (CD) probability (P_{CD}).

III. RAVEN

The RAVEN statistical framework is a code, funded by both NEAMS and RISMC programs, that allows the user to perform generic statistical analysis. By statistical analysis we include:

- Sampling of codes: either stochastic (e.g., Monte-Carlo [10] and Latin Hypercube Sampling [11]) or deterministic (e.g., grid and Dynamic Event Tree [12])
- Generation of Reduced Order Models [13] also known as Surrogate models
- Post-processing of the sampled data and generation of statistical parameters (e.g., mean, variance, covariance matrix)

Figure 1 shows a general overview of the elements that comprise the RAVEN statistical framework:

- Model: it represents the pipeline between input and output space. It comprises both codes and also Reduced Order Models
- Sampler: it is the driver for any specific sampling strategy (e.g., Monte-Carlo, LHS, DET)
- Database: the data storing entity
- Post-processing module: module that performs statistical analyses and visualizes results

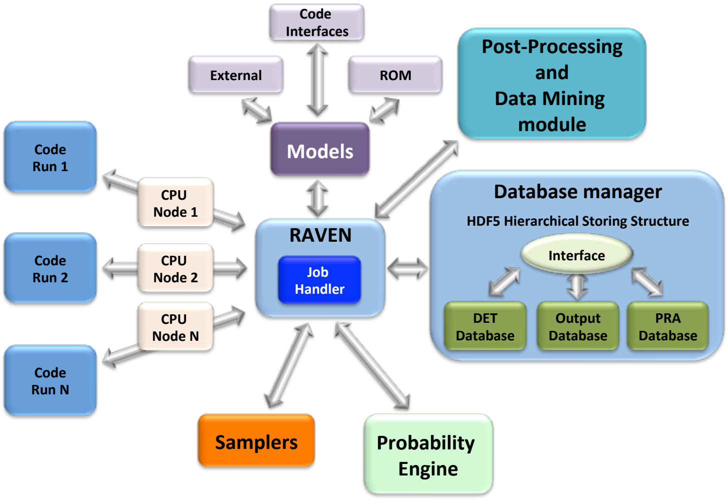

Fig. 1 – Scheme of RAVEN statistical framework components

IV. REDUCED ORDER MODELLING

Reduced order modeling is a fairly generic term. Its semantics changes from field to field (e.g., computer science, engineering and so on). For the scope of this article, we include in the Reduced Order Modeling term all methodologies and algorithms that aim to reduce the complexity of a problem, where "a problem" is considered a broad term and can be either an abstract entity (e.g., a simulator code or a dataset) or a concrete entity (e.g., an experimental facility, a power plant).

In the RISMC applications we are mainly dealing with numerical entities such as system codes like RELAP5-3D. It is relevant to highlight that the safety modeling of a nuclear system is not only a thermal-hydraulic problem but several other models are required: neutron transport, thermo-mechanics, chemistry, fracture propagation, etc. Note how the overall problem is not only multi-physics but also multi-scale both in the spatial scale but also in the temporal scale. While the coupling of these processes can be implicitly solved numerically, the RISMC project is focusing its attention toward the use of reduced order modeling techniques in order to decrease the computational cost (in terms of both computing power and memory requirement.)

In this document we divided the concept of Reduced Order Modeling into three main categories:

- Reduced physics: use of simulator codes that employ simplified physics problems. An example is the use of diffusion codes to solve neutronic problems instead of transport codes. In this category we also include the possibility to use in the same simulation run high and low fidelity models depending on the boundary conditions of the simulation.
- Reduced dimensionality: a simulation run can be seen as a trajectory in the phase space and a single point in the input space. The dimensionality of these spaces can be very high for the complex analyses. This category includes all methods than aim to reduce the dimensionality of these spaces and project the original problem into the reduced space.
- Surrogate model: surrogate models are mathematical objects that emulate the behavior of a code by learning its input/output relations and reconstructing such relations through a regression/interpolation based approach.

For the last two categories the set of methodologies employed are typically based on regression (e.g., Gaussian process models [14]), interpolation (e.g., spline kernel and linear kernel) and dimensionality reduction algorithms (e.g., Principal Component Analysis -PCA- [15] and ISOMAP [16]). In order to illustrate how Reduced Order Modeling techniques can be applied in the RISMC approach, we have indicated in Figure 2 the set of methods that can be applied to each of the four RISMC steps:

- Deterministic modeling: employment of reduced physics codes (i.e., multi-fidelity codes) or surrogate models instead of the actual codes
- Stochastic modeling: reduction of the number of stochastic parameters to be sampled (i.e., reduction of the dimensionality of the input space)
- Stochastic Analysis: reduction of the number of simulations to run by carefully choosing a minimum set of simulations that maximize the amount of information required by the analysis (adaptive – smart – sampling)
- Data Post-Processing: use of stochastic analysis tools (e.g., Kernel Density Estimation methods) to summarize large amounts of data and employment of advanced topology-based visualization tools to visualize high dimensional data.

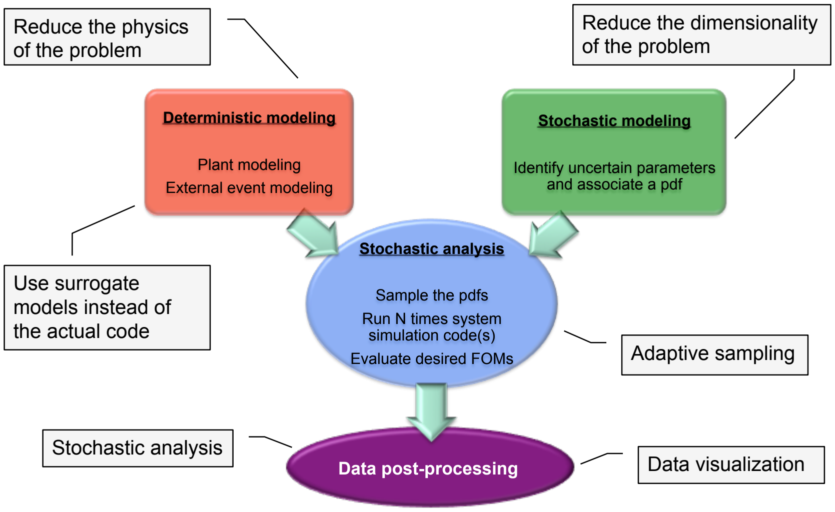

Fig. 2– Overview of how Reduced Order Modeling can be applied to the 4 main steps of a typical RISMC analysis

IV.A. Surrogate Models

A surrogate model is a mathematical model that aims to build a correlation given a set of data points. The starting point is typically a set of N data points:

$$(\boldsymbol{s_i}, \boldsymbol{\mathcal{H}}(\boldsymbol{s_i})) \qquad i = 1, \dots, N \tag{2}$$

that sample the response of the original model. Given the set of these N data points, the ROM is trained and the resulting outcome is a model $\boldsymbol{\Theta}(\boldsymbol{s})$ that approximates the original model $\boldsymbol{\mathcal{H}}(\boldsymbol{s})$(see Figure 3):

$$\boldsymbol{\Theta}(\boldsymbol{s}): \boldsymbol{s_i} \rightarrow \boldsymbol{\Theta}(\boldsymbol{s_i}) \cong \boldsymbol{\mathcal{H}}(\boldsymbol{s_i}) \tag{3}$$

The advantage of the ROM is the much faster computation of $\boldsymbol{\Theta}(\boldsymbol{s})$ (e.g., RELAP5-3D) compared to the original model $\boldsymbol{\mathcal{H}}(\boldsymbol{s})$. However, the evaluation of a ROM is affected by an intrinsic error, which can not always be bound and/or quantified.

We have identified two classes of ROM: model based and data based. In model based ROMs the prediction is performed using a blend of interpolation and regression algorithms[1]. Examples are:

- Gaussian Process Models (GPMs) [14]
- Multi-dimensional spline interpolators [17]

This class of algorithms has the advantage that they possess great prediction capabilities if the original $\boldsymbol{\mathcal{H}}(\boldsymbol{s})$ is relatively smooth (i.e., not discontinuous).

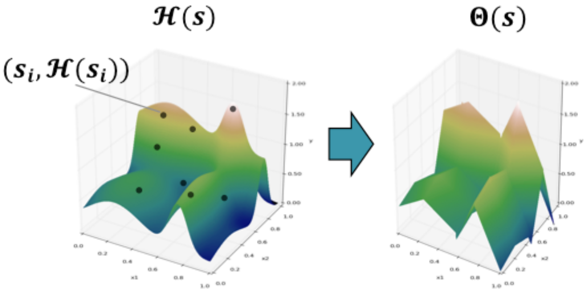

Fig. 3 – Example of reduced order modeling approximation of a sampled 3-D response surface

IV.B. Dimensionality Reduction

Dimensionality reduction is the process of finding a bijective mapping function $\mathfrak{I}$:

[1] Interpolation: Given a set of N data points (x_i, y_i) $i = 1, \dots, N$, interpolation aim to find a function F that is of some user-defined form (e.g., linear) that has the values in that points exactly as specified, i.e., it satisfies $F(x_i) = y_i$. Given the same set of N data points (x_i, y_i) $i = 1, \dots, N$, regression, regression look for a function that minimizes some cost, usually sum of squares of errors $\sum_{i=1}^{n}(F(x_i) - y_i)^2$. The requirement $F(x_i) = y_i$ is usually not imposed.

$$\mathfrak{I}: \mathbb{R}^D \to \mathbb{R}^d \quad \text{where } d < D$$

which maps the data points from the D-dimensional space into a reduced d-dimensional space (i.e. embedding on a manifold) in such a way that the distances between each point and its neighbors are preserved.

Linear algorithms, such as PCA [15] or multidimensional scaling (MDS) [18], have the advantage that they are easier to implement; however, they can only identify linear correlation among variables. On the other hand, methodologies such as Local Linear Embedding [19] and ISOMAP [16] are more computationally intensive but they are able to identify non-linear correlations.

The main idea behind PCA [15] is to perform a linear mapping of the data set onto a lower dimensional space such that the variance of the data in the low-dimensional representation is maximized. This is accomplished by determining the eigenvectors and their corresponding eigenvalues of the data covariance matrix[2] Σ. The eigenvectors that correspond to the largest eigenvalues (i.e., the principal components) can be used as a set of basis functions. Thus, the original space is reduced to the space spanned by a few eigenvectors.

Figure 4 shows an example of dimensionality reduction using PCA for a data set distributed in a 2-dimensional space. After performing the eigenvalue-eigenvector decomposition of the covariance matrix, the algorithm chooses the eigenvector having the largest eigenvalue (i.e., λ_1) as subspace to project the original data. The algorithm is very easy to implement but, on the other hand, PCA is not able to identify non-linear correlations of more complex data sets.

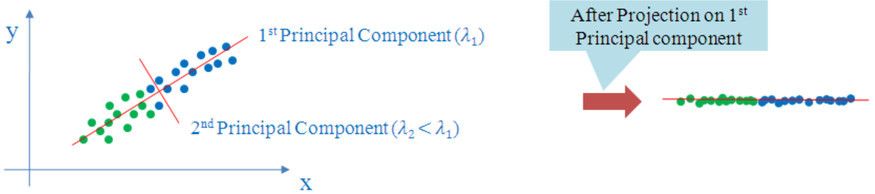
y
x
1st Principal Component (λ_1)
2nd Principal Component ($\lambda_2 < \lambda_1$)
After Projection on 1st Principal component

Fig. 4 – Example of dimensionality (from $D = 2$ to $d = 1$) reduction using PCA

IV.C Stochastic data analysis

The RISMC uncertainties and safety methods usually generate a large number of simulation runs (database storage may be on the order of gigabytes or higher). This section shows how clustering algorithms that can be used to analyze and extract information from large data sets containing time dependent data. In this context, “extracting information” means constructing input-output correlations, finding commonalities, and identifying outliers.

From a mathematical viewpoint, clustering [20] aims to find a partition $\boldsymbol{C} = \{C_1, \dots, C_l, \dots, C_L\}$ of Ξ where each C_l $(l = 1, \dots, L)$ is called a cluster. The partition $\boldsymbol{C}$ is such that:

$$\begin{cases} C_l \neq \emptyset \ \ \forall l = 1, \dots, L \\ \bigcup_{l=1}^{L} C_l = \Xi \end{cases} \tag{4}$$

Even though the number of clustering algorithms available in the literature is large, usually the most used ones when applied to time series are the following: Hierarchical [21], K-Means [22] and Mean-shift [23]. Hierarchical algorithms build a hierarchical tree from the individual points (leaves) by progressively merging them into clusters until all points are inside a single cluster (root). Clustering algorithms such as K-Means and Mean-Shift, on the other hand, seek a single partition of the data sets instead of a nested sequence of partitions obtained by hierarchical methodologies.

An example is shown in Fig. 5 applied to a data set containing the time evolution of 1000 time series has been generated by randomly changing (through a Monte-Carlo sampling) three variables (i.e., x, y, z). We introduced a “discontinuity” in the temporal evolution of the time series depending if $x > 4$ or $x < 4$. By using K-Means clustering algorithm we were able partition the 1000 generated scenario into 2 clusters (see Figure 6).

[2] Given a data set in form of a vector Z, rows correspond to data dimensions (D) and columns correspond to data observations (Λ), the covariance matrix Σ is determined as $\Sigma = \frac{1}{\Lambda - 1} Z\, Z'$

.

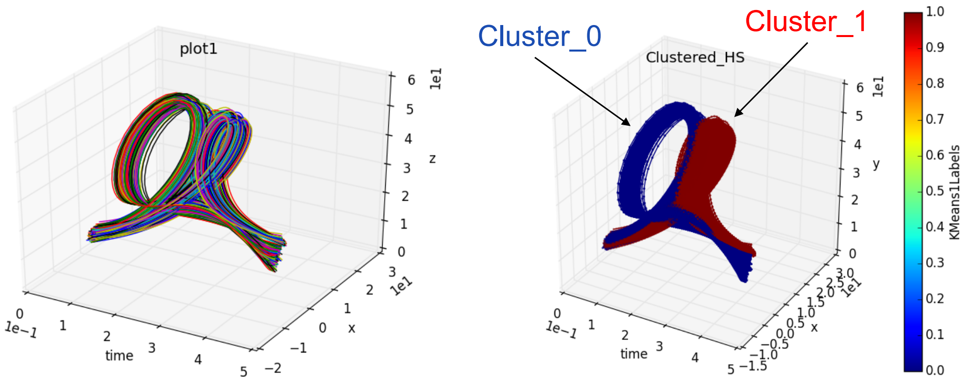

Fig. 5 - Plot of a 1000 time series data set in a 2-dimensional space (plus time) Plot of the clusters obtained

Note how the scenarios in each cluster have a very similar temporal behavior. Then, by looking at the histograms of the sampled variables x, y, z for the scenarios contained in each cluster we were able to verify that x was creating the splitting of the data set. Figure 7 shows the histograms of x for both clusters: for Cluster_0 $x < 4$ while $x > 4$ for Cluster_1. Note that we would not have been able to capture this "discontinuity" by considering only the end or max values of the time series [24,25].

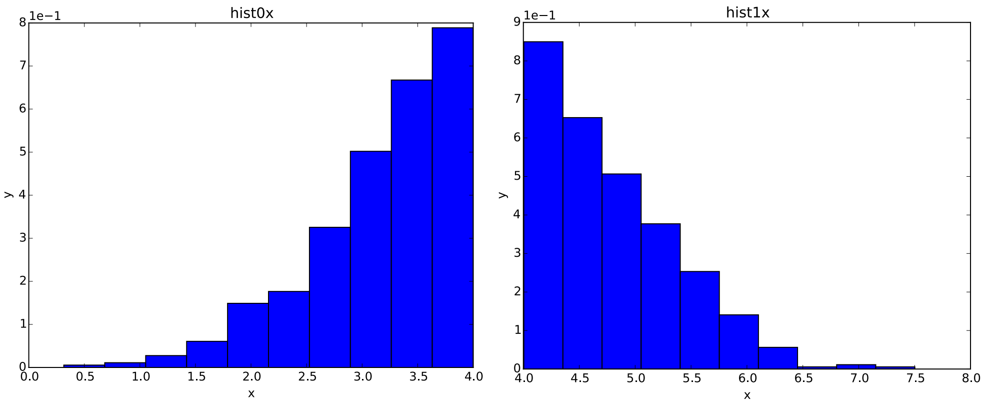

Fig. 6 - Histograms of the sampled values for Cluster_0 and Cluster_1 (shown in Fig. 6) that created them and were captured by the clustering algorithm.

V. PRA APPLICATIONS

In this section we cover a set of applications that employs Reduced Order Modeling techniques. We will show a set of applications that are of interest in the RISMC pathway.

V.A. Adaptive Sampling

The general adaptive sampling pipeline [26] begins by selecting some initial training data, running the simulation and obtaining a collection of true responses at these data points. Second, it fits a response surface surrogate model from the initial set of training data. Third, a set of candidate points is chosen in the parameter space based on certain sampling techniques, and the surrogate model is evaluated at these points, obtaining a set of approximated values. Fourth, each candidate point is assigned a score based on some adaptive sampling scoring function (usually derived from qualitative or quantitative relations between the training points, their true and estimated response values). Finally, the candidate(s) with the highest score(s) are selected and added to the set of training data to begin a new cycle.

As mentioned earlier this kind of sampling strategy requires not only simulator codes but also one, or possibly more, ROMs [13]. In our case, it is possible to view the code as a black-box $\boldsymbol{\mathcal{H}}$ that produces a set of output variables $\boldsymbol{y}$ given a set of input parameters $\boldsymbol{s}$:

$$\boldsymbol{\mathcal{H}}: \boldsymbol{s} \rightarrow \boldsymbol{y}(t) = \boldsymbol{\mathcal{H}}(\boldsymbol{\theta}, \boldsymbol{s}, t) \qquad (5)$$

In addition, it is needed to provide what we call an "objective function". The objective function gives indications on what is the desired "exploration" criteria. The main adaptive sampling steps are explained as follows (see Figure 7):

1. Perform a set of runs of the simulator code: the number of required runs may depend on the dimensionality of the input space.
2. Given the set of simulation runs obtained in Step 1, create a ROM. The objective of this ROM is to:
   - Infer the response of the simulator code, i.e., create an approximate output given the same set of input parameters
   - Predict the regions in the input space that maximizes the objective function
3. Employ the ROM to approximate the structure of the goal function
4. Identify a set of points that satisfy the conditions specified in the goal function and choose a subset of points from the ones obtained in Step 4 that maximize the goal function
5. Perform a simulation run for each of the points obtained in Step 5 using the simulator code
6. Repeat Steps 2 through 6 until convergence is reached

As part of safety margin quantification, the RISMC approach aims to evaluate a set of limit surfaces [27]. A limit surface (see Figure 7) represents the boundaries in the input space (i.e., d-dimensional space; each dimension is one the d sampled variables) that separate the failure region (i.e., characterized by the undesired simulation outcome; e.g., core damage) from the success region (i.e., characterized by the desired simulation outcome; e.g., max clad temperature below 2200 F).

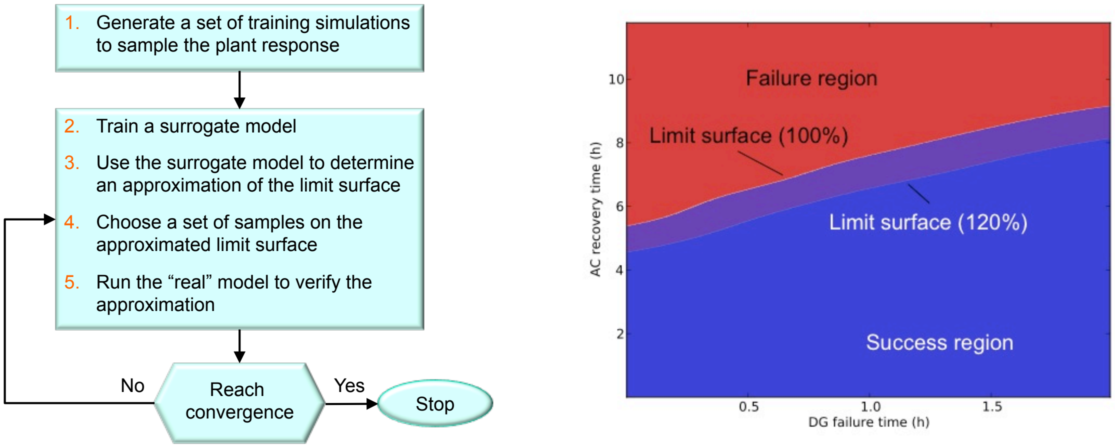
1. Generate a set of training simulations to sample the plant response
2. Train a surrogate model
3. Use the surrogate model to determine an approximation of the limit surface
4. Choose a set of samples on the approximated limit surface
5. Run the "real" model to verify the approximation
Reach convergence
No
Yes
Stop
Failure region
Limit surface (100%)
Limit surface (120%)
Success region
AC recovery time (h)
DG failure time (h)
10
8
6
4
2
0.5
1.0
1.5

Fig. 7 – Workflow of adaptive sampling algorithms (left) and example of limit surface calculation for two different values of core power levels [28]

The limit surface has a pure deterministic value; the stochastic information is generated when the probability of occurrence of the undesired event (e.g., core damage) P_{CD} is determined as:

$$P_{CD} = \int_{\substack{failure\\ region}} pdf(\varpi)\, d\varpi \tag{6}$$

Equation 6 shows that P_{CD} is equal to the area of the failure region weighted by the probability of being in the failure region itself (through the probability distribution function $pdf(\varpi)$).

Figure 7 shows the limit surface in a 2-dimensional space generated in [28] using RAVEN for a pressurized water reactor station blackout initiating event. As part of the analysis, we were interested in the evaluation of the safety impacts of power uprate (reactor core power increased from 100 to 120%). Such evaluation has been performed by evaluating both the increased core damage probability ΔP_{CD} and the limit surface for both 100 and 120% reactor core power levels Note that ΔP_{CD} can be written as the same integral indicated in Eq. 6 but evaluated only in the *expanded failure region* ($\Delta\Omega_{Failure}$)

$$\Delta P_{CD} = \int_{\Delta\Omega_{Failure}} pdf(\varpi)\, d\varpi \tag{7}$$

V.B. Uncertainty Quantification and Sensitivity Analysis

Another set of application of Surrogate Models relevant to the RISMC project is Uncertainty Quantification (UQ) and Sensitivity Analysis (SA). For these kinds of applications we are following a response-surface approach where a Surrogate Models is trained in the region of the input space of interest. This training process aim to reconstruct the system response in this limited region of the input space. Then, the forward propagation of the uncertainties of the input parameters is performed by using the surrogate models instead of the actual code. In order to evaluate the performances of surrogate models we have decided to compare the following:

- First three moments of the figure of merits: mean, sigma and skewness
- Pearson coefficients of the input parameters
- Sensitivity coefficients of the input parameters

Figure 8 shows a plot of the response function for the PWR natural circulation loop using peak fuel temperature as the figure of merit.

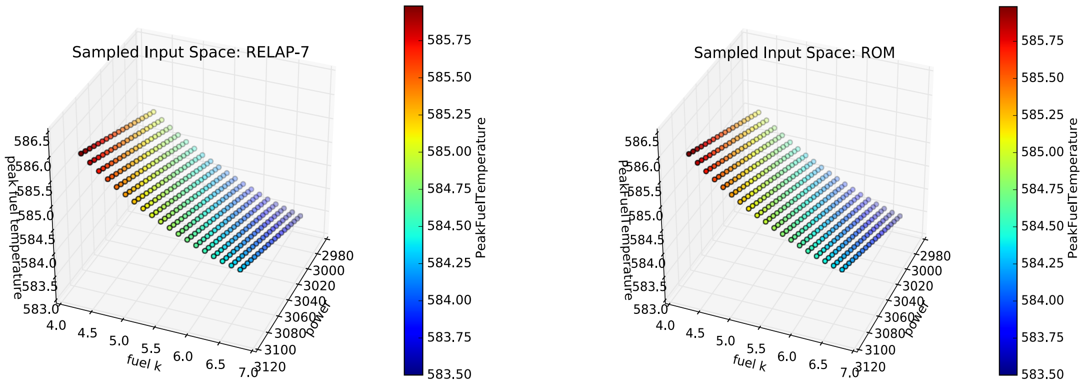

Fig. 8 – Comparsion of the response function of peak fuel temperature: original data from code (left) and the one obtained by using the ROM (right)

Table 1 – Comparison of the first three statistical moments, sensitivity and Pearson coefficients of peak fuel temperature

	Code	ROM	Relative Error
mu	5.845 E+2	5.845 E+2	3.90 E-7
sigma	3.469 E-1	3.472 E-1	-9.64 E-4
skewness	3.446 E-1	3.434 E-1	3.72 E-3
K sensitivity	-6.663 E-1	-6.671 E-1	-1.15 E-3
P sensitivity	7.231 E-3	7.234 E-3	-3.65 E-4
K pearson	-9.056 E-1	-9.057 E-1	-1.90 E-4
P pearson	-3.972 E-4	-3.972E-4	1.20 E-8

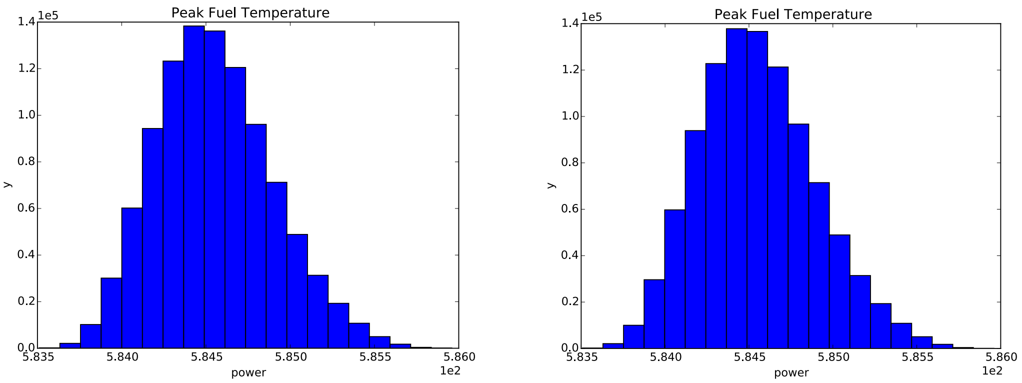

Fig. 9 – Distribution of peak fuel temperature: code (left) and ROM (right)

V.C. Temporal Predictors

In the previous section we introduced the concept of response surface methods and surrogate models as tools to predict an approximated $\boldsymbol{\Theta}(\boldsymbol{s})$ (which represents, for example, a simulated system response under an accident scenario) for a set of conditions specified in $\boldsymbol{s}$. The vector $\boldsymbol{s}$ contains elements s_d such as timing and sequencing of events (e.g., recovery time of

AC power, failure time of core cooling injection). Note that the value $\boldsymbol{\Theta}(\boldsymbol{s})$ is a scalar and, thus, does not contain any temporal evolution type of information.

We extend the concept of ROM in order to be able to handle time dependent $\boldsymbol{\Theta}(\boldsymbol{s})$: given $\boldsymbol{s}$, $\boldsymbol{\Theta}(\boldsymbol{s},t)$ is a time dependent variable. In this case, the training consists of N points:

$$(\boldsymbol{s}_i, \boldsymbol{\mathcal{H}}(\boldsymbol{s},t)_i) \;\; i = 1, \dots, N \tag{8}$$

Our approach is to start by dividing the temporal scale into intervals (assumed here to be of equal length but it is not required):

$$t = [t_1, \dots, t_T] \tag{9}$$

For each time point t_k $(k = 1, \dots, T)$ we consider the subset of points:

$$(\boldsymbol{s}_i, \boldsymbol{\mathcal{H}}(\boldsymbol{s},t_k)_i) \;\; i = 1, \dots, N \tag{10}$$

and we build the corresponding $\boldsymbol{\Theta}(\boldsymbol{s})_k$. Thus, now we have a set of ROMs for each time point t_k $(k = 1, \dots, T)$. The temporal predictor $\Psi(\boldsymbol{x},t)$ is simply the vector of:

$$\boldsymbol{\Psi}(\boldsymbol{x},t) = [\boldsymbol{\Theta}(\boldsymbol{s})_1, \dots, \boldsymbol{\Theta}(\boldsymbol{s})_k, \dots, \boldsymbol{\Theta}(\boldsymbol{s})_T] \tag{11}$$

In our applications, when each of the data points has been generated by safety analysis codes (e.g., RELAP5-3D):

- $\boldsymbol{s}$ is the configuration of the simulation (e.g., timing of events, values associated with uncertain parameters)
- $\boldsymbol{\Theta}(\boldsymbol{s},t)$ is the simulation associated with $\boldsymbol{s}$.

We performed a few tests with different types of datasets in order to identify performances and limitations of this algorithm. Figure 10 (left) shows a set of $n = 20$ simulations, i.e. $\boldsymbol{\mathcal{H}}(\boldsymbol{s},t)_i$ $(i = 1, \dots, 20)$, generated by sampling two stochastic parameters, i.e. $\boldsymbol{s}_i = [s_1, s_2]$. We initially divided the time scale uniformly [0,2500] into $T = 100$ intervals and for each time point t_k $(k = 1, \dots, 100)$ we considered the data points $(\boldsymbol{x}_i, \boldsymbol{\mathcal{H}}(\boldsymbol{s},t_k)_i)$ $(i = 1, \dots, 20)$ and built the reduced order models $\boldsymbol{\Theta}(\boldsymbol{s})_k$. We then tested the temporal predictor:

$$\boldsymbol{\Psi}(\boldsymbol{s},t) = [\boldsymbol{\Theta}(\boldsymbol{s})_1, \dots, \boldsymbol{\Theta}(\boldsymbol{s})_{100}] \tag{12}$$

for several $\boldsymbol{s}_j$ $(j \neq i)$ and compared them with the simulated $\boldsymbol{\Theta}(\boldsymbol{s},t)$.

Figure 10 (right) shows the predicted scenario $\Psi(\boldsymbol{s},t)$ (green line) and the actual simulated scenario $\boldsymbol{\mathcal{H}}(\boldsymbol{s},t)$. For this particular case we built $\Psi(\boldsymbol{s},t)$ using Gaussian Process Models [14] as basic ROM. A useful feature is that these algorithms are also capable of providing the uncertainty associated with the predicted results.

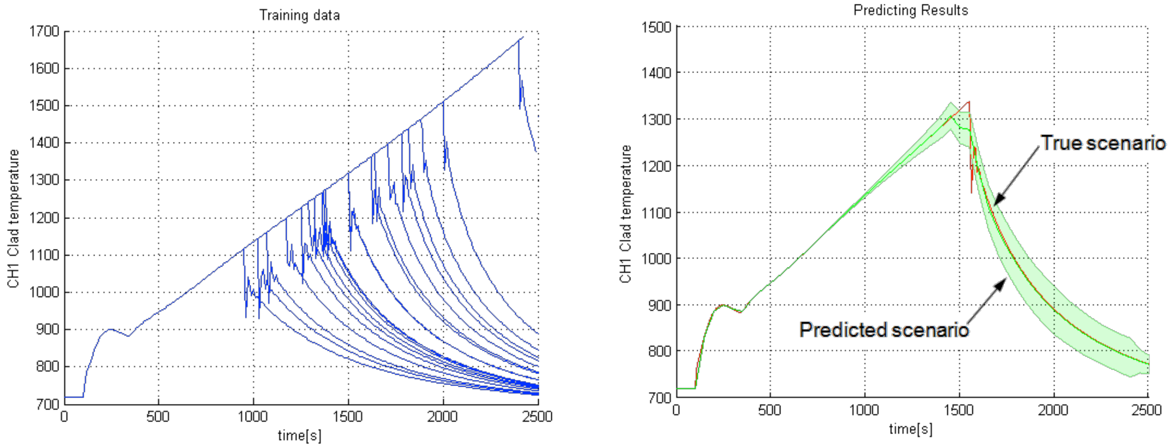

Fig. 10 – Initial protoype results of temporal surrogate model

VII. CONCLUSIONS

In this article we have given a fairly broad overview of the Reduced Order Modeling capabilities available in the RISMC toolkit and, in particular, in the RAVEN statistical framework. We have shown how Reduced Order Modeling can be applied in every step of a typical RISMC analysis: from the generation to the analysis and to visualization of data. We have shown how surrogate models can be used as substitute for actual code to speed up the statistical analysis required by the RISMC approach. We have indicated how it is possible to reduce the computational cost of such statistical analyses by smartly sampling the input space. The most important application of reduced order modeling techniques focuses on propagation of uncertainties and sensitivity analysis types of applications. In this respect we employed few thermo-hydraulic models of RELAP-7 and we showed how this process can be performed by using the RAVEN code. We have shown how reduced order modeling techniques can be also employed for data mining types of applications to visualize high dimensional data and extract useful information from large amounts of data. We have in particular focused on surrogate models, both classifiers and regressors. We have tested their performances on a set of analyses that are of interest in the RISMC approach. We used not only analytical tests but also tests that involved RELAP-7 systems. The comparison allowed us to identify the pros and cons of each algorithm and identify the best surrogate model depending on the case under consideration. This overview was not however exhaustive since many more surrogate models are available in RAVEN. The set of algorithms we chose is a representative set of methods that most likely will be used within the RISMC project.

REFERENCES

[1] C. Smith, C. Rabiti, and R. Martineau, "Risk Informed Safety Margins Characterization (RISMC) Pathway Technical Program Plan", Idaho National Laboratory technical report: INL/EXT-11-22977 (2011).

[2] C. W. Gardiner, *Handbook of stochastic methods: for physics, chemistry and the natural sciences*, Springer series in synergetics, 13, Springer (2002).

[3] D. Mandelli, C. Smith, T. Riley, J. Nielsen, J. Schroeder, C. Rabiti, A. Alfonsi, J. Nielsen, R. Kinoshita, D. Maljovec, B. Wang, and V. Pascucci, "Overview of new tools to perform safety analysis: BWR station black out test case," in *Proceedings for PSAM 12 Conference*, Honolulu (2014).

[4] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, R. Kinoshita, and A. Naviglio, "RAVEN and Dynamic Probabilistic Risk Assessment: Software Overview," in *Proceedings of European Safety and Reliability Conference ESREL* (2014).

[5] D. Mandelli, S. Prescott, C. Smith, A. Alfonsi, C. Rabiti, J. Cogliati, R. Kinoshita, "Modeling of a Flooding Induced Station Blackout for a Pressurized Water Reactor Using the RISMC Toolkit," *in ANS PSA 2015 International Topical Meeting on Probabilistic Safety Assessment and Analysis Columbia, SC*, on CD-ROM, American Nuclear Society, LaGrange Park, IL, 2015.

[6] S. Prescott, C. Smith, R. Sampath, "Incorporating Dynamic 3D Simulation into PRA", in *ANS PSA 2015 International Topical Meeting on Probabilistic Safety Assessment and Analysis Columbia, SC,* on CD-ROM, American Nuclear Society, LaGrange Park, IL, 2015.

[7] R. L. Boring, R. Benish Shirley, J. C. Joe, D, Mandelli, and C. Smith, "Simulation and Non-Simulation Based Human Reliability Analysis Approaches", Idaho National Laboratory technical report: INL/EXT-14-33903 (2014).

[8] RELAP5 Code Development Team, "RELAP5-3D Code Manual", Idaho National Laboratory Technical Report INEEL-EXT-98-00834 (2012).

[9] Mandelli, D., Smith, C., Riley, T., J. Nielsen, Alfonsi, A., Cogliati, J., Rabiti, C., Schroeder, J., "BWR Station Blackout: A RISMC Analysis Using RAVEN and RELAP-3D", *Nuclear Technology*, **193**, no. 1, pp. 161-174 (2016).

[10] E. Zio, M. Marseguerra, J. Devooght, and P. Labeau, "A concept paper on dynamic reliability via Monte Carlo simulation," in *Mathematics and Computers in Simulation*, pp. 47-371 (1998).

[11] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," Reliability Engineering & System Safety, 81-1 (2003).

[12] A. Amendola and G. Reina, "Dylam-1, a software package for event sequence and consequence spectrum methodology," in EUR-924, CEC-JRC. ISPRA: Commission of the European Communities (1984).

[13] H. S. Abdel-Khalik, Y. Bang, J. M. Hite, C. B. Kennedy, C. Wang, "Reduced Order Modeling For Nonlinear Multi-Component Models," *International Journal on Uncertainty Quantification*, **2** - 4, pp. 341-361 (2012).

[14] C. E. Rasmussen, "Gaussian Processes in Machine Learning", Advanced Lectures on Machine Learning, Lecture Notes in Computer Science, **3176**, pp. 63–71 (2004).

[15] I. T. Jolliffe, *Principal Component Analysis*, Springer, second ed., October 2002.

[16] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, **290**, pp. 2319-2323 (2000).

[17] C. Habermann and F. Kindermann, "Multidimensional Spline Interpolation: Theory and Applications," *Computational Economics*, **30** – 2, pp. 153-169 (2007).

[18] I. Borg and P. Groenen, Modern Multidimensional Scaling: Theory and Applications. Springer-Verlag New York (2005).

[19] S. T. Roweis and L. K. Saul, ``Nonlinear dimensionality reduction by locally linear embedding," Science, vol. 290, pp. 2323-2328 (2000).

[20] A. K. Jain, K. Dubes, and C. Richard, *Algorithms for clustering data*, Upper Saddle River, NJ (USA): Prentice-Hall, Inc. (1988).

[21] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," ACM Computing Surveys, **31**, no. 3, pp. 264-323 (1999).

[22] J. B. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, pp. 281-297, University of California Press (1967).

[23] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, no. 8, pp. 790-799 (1995).

[24] D. Mandelli, A. Yilmaz, T. Aldemir, K. Metzroth, and R. Denning, "Scenario clustering and dynamic probabilistic risk assessment," *Reliability Engineering & System Safety*, **115**, pp. 146-160 (2013).

[25] D. Mandelli, C. Smith, C. Rabiti, A. Alfonsi, R. Youngblood, V. Pascucci, B. Wang, D. Maljovec, P.-T. Bremer, T. Aldemir, A. Yilmaz, and D. Zamalieva, "Dynamic PRA: an overview of new algorithms to generate, analyze and visualize data," *in Proceeding of American Nuclear Society (ANS)*, Washington DC (2013).

[26] D. Mandelli and C. Smith, "Adaptive sampling using support vector machines," in *Proceeding of American Nuclear Society (ANS)*, San Diego (CA), **107**, pp. 736-738 (2012).

[27] D. Mandelli, S. Prescott, C. Smith, A. Alfonsi, C. Rabiti, J. Cogliati, and R. Kinoshita, A fooding induced station blackout analysis for a pressurized water reactor using the RISMC toolkit," Science and Technology of Nuclear Installations, **2015** (2015).