IMPROVED SAMPLING ALGORITHMS IN RISK-INFORMED SAFETY APPLICATIONS

D. Mandelli[†], A. Alfonsi, C. Smith, C. Rabiti, J. Cogliati

Idaho National Laboratory, 2525 North Fremont Street, Idaho Falls (ID)
[†] *Corresponding author:* *diego.mandelli@inl.gov*

The Risk-Informed Safety Margin Characterization (RISMC) approach is developing an advanced set of simulation-based methodologies in order to perform Probabilistic Risk Analyses. These methods randomly perturb (by employing sampling algorithms) timing/sequencing of events and uncertain parameters of the physics-based models in order to estimate stochastic outcomes such as off-normal and damage states of the facility. This modeling approach applied to complex systems such as nuclear power plants requires the analyst to perform a series of computationally-expensive simulation runs given a large set of uncertain parameters. One issue is related to the fact that the space of the possible solutions can be sampled only sparsely and this precludes the ability to fully analyze the impact of uncertainties on the system dynamics. This paper describes how we can use novel methods that optimize the information generated by the sampling process by sampling unexplored or risk-significant regions of the issue space; we call this approach adaptive (smart) sampling algorithms. These methods infer the system response using surrogate models constructed from existing samples and predict the best location of the next sample. Thus, it is possible to understand features of the issue space with a smaller number of carefully selected samples. In this paper, we present how it is possible to perform adaptive sampling using the RAVEN statistical tool and highlight the advantages compared to more classical sampling approaches such as Monte-Carlo.

I. INTRODUCTION

The Risk-Informed Safety Margin Characterization (RISMC) [1] Pathway (as part of the Light Water Sustainability (LWRS) Program [2]) aims to develop simulation-based tools and methods to assess risks for existing Nuclear Power Plants (NPPs).

This Pathway, by developing new simulation-based methods, is extending the Probabilistic Risk assessment (PRA) state-of-the-practice methods [3] which have been traditionally based on logic structures such as Event-Trees (ETs) and Fault-Trees (FTs) [4]. In more detail, the RISMC approach uses stochastic frameworks (i.e., RAVEN [5]) coupled with deterministic codes that model specific physical aspects of the plant (e.g., thermo-hydraulic and thermo-mechanic using RELAP5-3D [6] or RELAP-7 [7], and GRIZZLY [8] respectively).

One research direction is the use of surrogate models, also known as Reduced Order Models (ROMs), as possible substitutes for one or more of the needed physical aspects. The use of ROMs can greatly reduce the computational cost of a single multi-physics simulation run. This advantage is relevant when many simulation runs need to be performed according to the desired stochastic analysis (usually through a stochastic sampling process).

II. RISMC APPROACH

A single simulation 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:

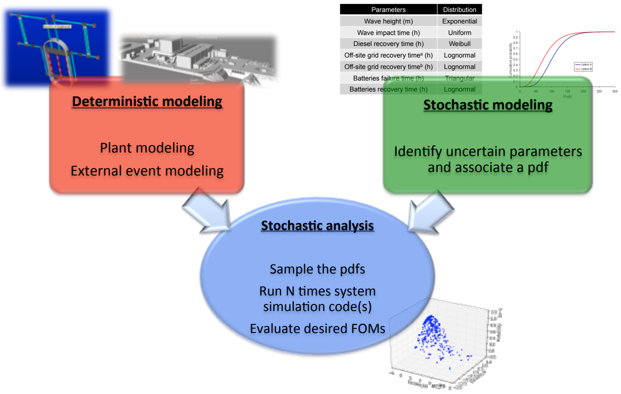
$$\frac{\partial \boldsymbol{\theta}(t)}{\partial t} = \boldsymbol{\mathcal{H}}(\boldsymbol{\theta}, \boldsymbol{s}, t) \qquad (1)$$

where:

- $\boldsymbol{\theta} = \boldsymbol{\theta}(t)$ represents the status of the system as function of time *t*; $\boldsymbol{\theta}(t)$ represents a 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 following these four steps (see Fig. 1):

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 core damage (CD) probability (P_{CD}).



Parameters	Distribution
Wave height (m)	Exponential
Wave impact time (h)	Uniform
Diesel recovery time (h)	Weibull
Off-site grid recovery time^a (h)	Lognormal
Off-site grid recovery time^b (h)	Lognormal
Batteries failure time (h)	Triangular
Batteries recovery time (h)	Lognormal

Fig. 1. Overview of the RISMC modeling approach.

III. RAVEN FRAMEWORK

In order to perform PRA analyses of NPPs, the RISMC pathway is employing the RAVEN statistical framework, which is a recent add-on of the RAVEN package [9] that allows the user to perform generic statistical analysis. By statistical analysis we include: sampling of codes (e.g., Monte-Carlo [10] and Latin Hypercube Sampling [11], grid sampling, and Dynamic Event Tree (DET) [12]), generation of ROMs [13] (also known as surrogate models or emulators) and post-processing of the sampled data and generation of statistical parameters (e.g., mean, variance, covariance matrix).

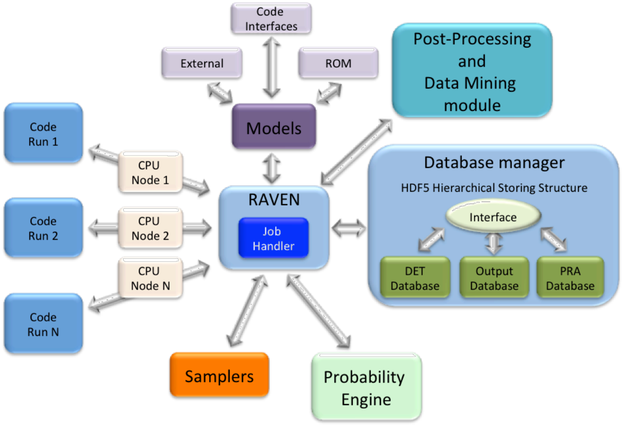


Fig. 2. Overview of the RAVEN statistical framework.

Figure 2 shows an overview of the elements that comprise the RAVEN statistical framework:

- Model: represents the pipeline between the input and output spaces. It is comprised of both interfaces for mechanistic codes (e.g., RELAP-7) and ROMs
- Sampler: the driver for any specific sampling strategy (e.g., Monte-Carlo, Latin Hypercube Sampling, DET)
- Database: the data storing entity
- Post-processing: module that performs statistical analyses and visualizes results

RAVEN is interfaced with several codes and, additionally, the users can build their own interfaces for the code they are interested in.

If multiple simulations need to be run, RAVEN has the capability to run simulations in parallel on multiple nodes and/or multiple CPUs. RAVEN applicability ranges from Linux based desktop/laptop to high performance computing machines.

RAVEN has also the capability to “train” ROMs from any data set generated by any code. These ROMs are usually a blend of interpolation and regression algorithms and such “training process” basically consists of setting the optimal parameters of the interpolation and regression algorithms that best fit the input data set. Once the ROMs are generated, they can be used instead of the actual codes to perform any type of analysis since the generation of data from ROMs is much faster than the original code.

IV. SURROGATE MODELS

A ROM 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:

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

that samples the response $\boldsymbol{\mathcal{H}}(\boldsymbol{s})$ of the original model. Given the set of these N data points, the ROM is trained and the resulting outcome is a model $\boldsymbol{\Theta}(\boldsymbol{s})$ [13] that approximates the original model $\boldsymbol{\mathcal{H}}(\boldsymbol{s})$(see Figure 2):

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

The advantage is the much faster computation of $\boldsymbol{\Theta}(\boldsymbol{s})$ (e.g., RELAP) compared to the original model $\boldsymbol{\mathcal{H}}(\boldsymbol{s})$.

We have identified two classes of ROM: model based and data based. These two classes are described in the next two sections.

IV.A. Model Based ROMs

In model based ROMs the prediction is performed using a blend of interpolation and regression algorithms. Examples are:

- Gaussian Process Models (GPMs)
- Multi-dimensional spline interpolators

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).

IV.B. Data Based ROMs

In data based ROMs the prediction is performed by solely considering the input data by using data searching algorithms. Examples are:

- K nearest neighbor classifier (KNN) [14]

- Graph based models [15]

While the predictions of this class of ROMs is limited compared to model based ROMs, they have the advantage that they are able to handle very discontinuous $\boldsymbol{\mathcal{H}}(\boldsymbol{s})$.

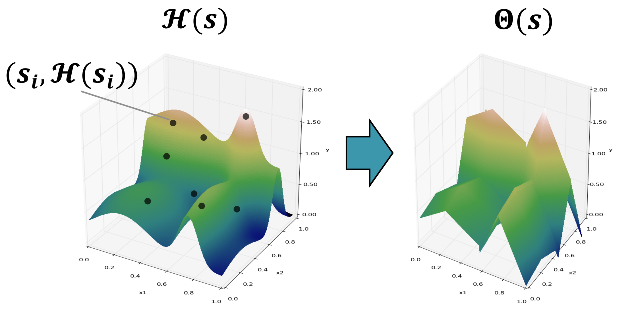


Fig. 3. Example of reduced order modeling approximation of a sampled 3-D response surface.

V. ADAPTIVE SAMPLING

The general adaptive sampling pipeline [16,17] begins firstly 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 the selected sampling technique, 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.

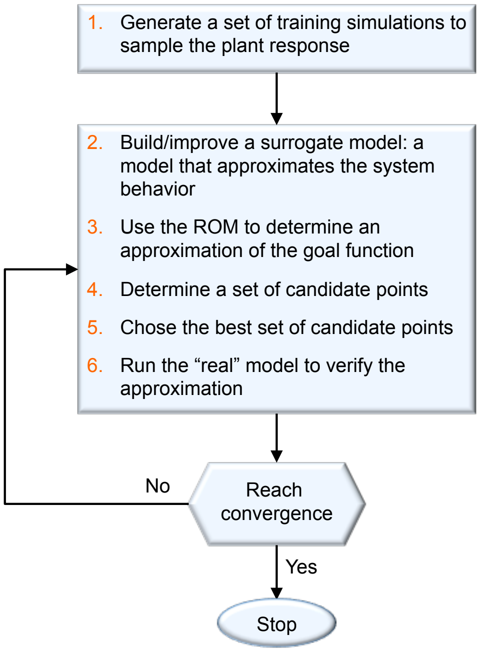
As mentioned earlier this kind of sampling strategy requires not only simulator codes but also one, or possibly more, ROMs [13]. In our case, it is possible to view the code as a black-box $\boldsymbol{\mathcal{H}}$ that produces a set of output variables $\boldsymbol{y}$ given a set of input parameters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

$$\boldsymbol{\mathcal{H}}: \boldsymbol{s} \rightarrow \boldsymbol{y}(t) = \boldsymbol{\mathcal{H}}(\boldsymbol{\theta}, \boldsymbol{s}, t) \qquad (4)$$

In addition, an "objective function" is needed. This function gives indications on what is the desired "exploration" criteria. We give more description of the objective functions in Section V.A.

The main adaptive sampling steps are explained as follows (see Fig. 4)

1. Perform a set of runs using 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 maximize the objective function
3. Employ the ROM to approximate the structure of the goal function
4. Identify a set of points that satisfies the conditions specified in the goal function
5. Chose a subset of points from the ones obtained in Step 4 that maximize the goal function
6. Perform a simulation run for each of the points obtained in Step 5 using the simulator code
7. Repeat Steps 2-6 until convergence is reached



1. Generate a set of training simulations to sample the plant response
2. Build/improve a surrogate model: a model that approximates the system behavior
3. Use the ROM to determine an approximation of the goal function
4. Determine a set of candidate points
5. Chose the best set of candidate points
6. Run the "real" model to verify the approximation
Reach convergence
No
Yes
Stop

Fig. 4. Workflow of adaptive sampling algorithms.

V.A. Objective Functions: Limit Surface

As part of safety margin quantification, the RISMC approach aims to evaluate a set of limit surfaces [16,17]. A limit surface represents the boundaries in the input space (i.e., *d*-dimensional space; each dimension is one the *d* sampled variables) that separate failure region (i.e., characterized by the undesired simulation outcome; e.g., core damage) from success region (i.e., characterized by the desired simulation outcome; e.g., max clad temperature below 2200 F).

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

$$P_{CD} = \int_{\substack{failure \\ region}} pdf(\varpi)\, d\varpi \qquad (5)$$

Equation 5 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 pdf $pdf(\varpi)$).

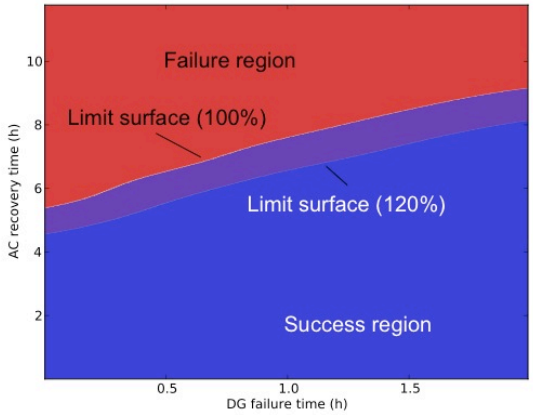


Fig. 5. Example of limit surface calculation for two different values of core power levels.

Figure 5 shows the limit surface in a 2-dimensional space generated in [18] using RAVEN coupled with RELAP-7 for a boiling 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 level.

Note that ΔP_{CD} can be written as the same integral indicated in Eq. 9 but evaluated only in the *expanded failure region* ($\Delta\Omega_{Failure}$)

$$\Delta P_{CD} = \int_{\Delta\Omega_{Failure}} pdf(\varpi)\, d\varpi \tag{6}$$

V.B. Convergence Criteria

During the adaptive sampling step, RAVEN continues to generate new sample coordinates until convergence is reached. RAVEN allows three types of convergence criteria:

1. Maximum number of samples
2. CDF-weight: the convergence is checked in terms of probability (cumulative distribution function: CDF)
3. Value-weight: the convergence is checked on the hyper-volume in terms of variable values

In addition it is possible to specify a value of persistence which allows the user to create an additional convergence check. It represents the number of times the computed convergence error needs to be below the tolerance value given by the user before stopping the adaptive-sampling calculation.

VI. TEST CASES

In this section we show how the methods and algorithms presented in the past sections are applied for particular sampling cases. These sampling cases cover both analytical problems (see Sections VI.A. VI.B. and VI.C.) but also safety related applications using safety analysis codes such as RELAP-7 (see Section VI.D.). For these cases we employed SVMs with Gauss kernel (also know as radial basis function) [19] as ROM to guide the choice of the adaptive sampling samples.

VI.A. Single Region

In this simple case, the limit surface is a single region located at the lower left corner of a 2-dimensional space. This is the most basic case that can be encountered. The model considered for this test case, given x_1 and x_2 produces an output variable y as:

$$y = x_1^2 + x_2 - 0.5 \tag{7}$$

Figure 6 shows the surface representing y; this test case failure occurs when $y > 0$ (red plane in Fig. 6) while the analytical limit surface is the red line shown in Fig. 6.

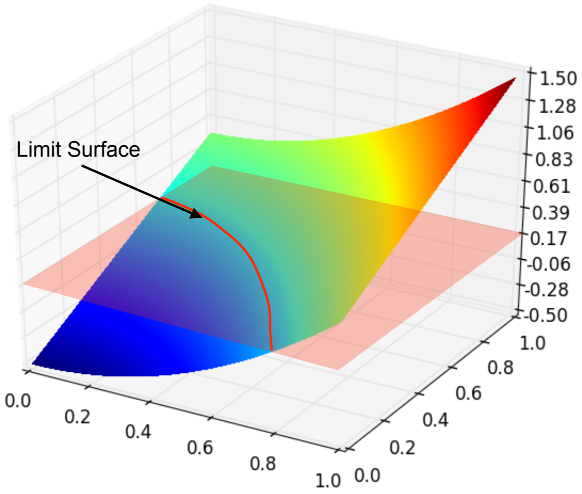


Fig. 6. Analytical shape of the single region limit surface estimated limit surface for different adaptive sampling iterations.

Figure 7 show a summary of the adaptive sampling process generated using RAVEN. Each row in the plots of Figure 7 corresponds to an iteration of the adaptive sampling process. For each iteration two plots are shown: sample locations (left) and the estimated limit surface (right). For this case convergence (convergence in value is equal to 5 10^5) was reached after 170 samples.

Table 1 compares the number of samples required to evaluate this limit surface by using classical Monte-Carlo and adaptive sampling. Note the much higher number of Monte-Carlo samples that are needed to achieve such low value of convergence.

TABLE 1. Number of samples required to evaluate the single region limit surface by using classical Monte-Carlo and adaptive sampling (convergence in value is equal to $5\ 10^{-5}$)

	Number of Samples
Monte-Carlo	~10^7
Adaptive	170

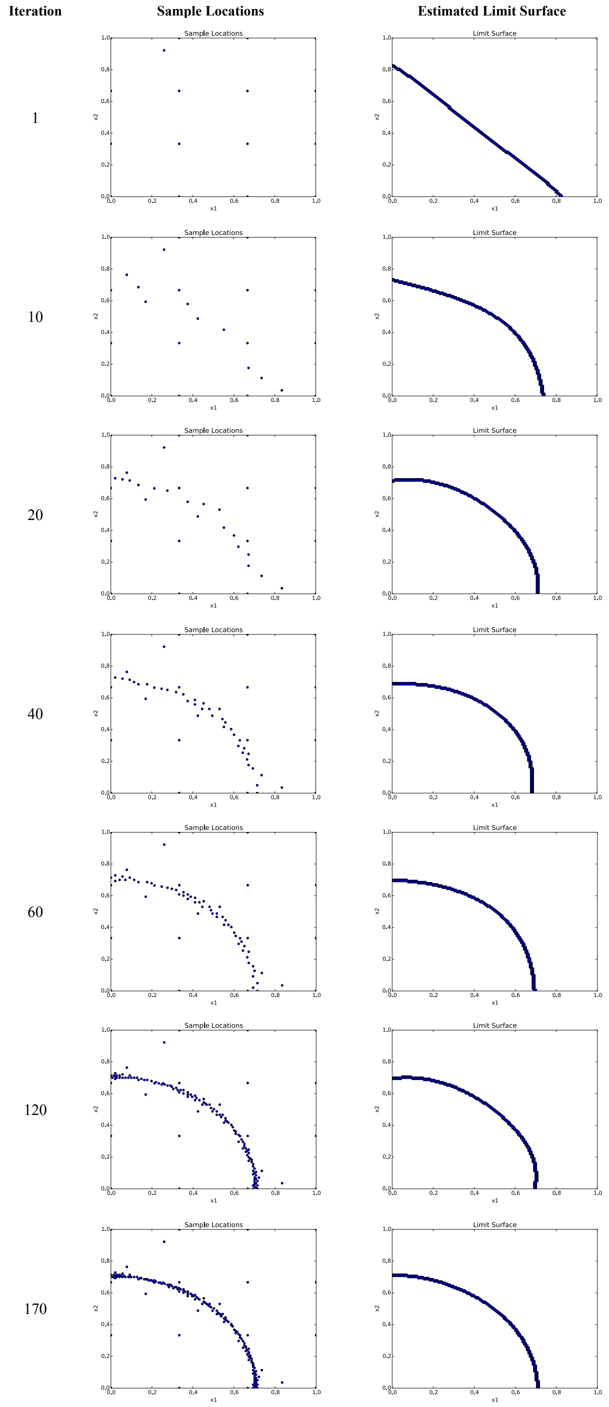


Fig. 7. Single region limit surface: sample locations and the estimated limit surface for different adaptive sampling iterations.

VI.B. Multiple Regions

In this second analytical test case we provided a more challenging test case where the limit surface lies not in one but in multiple regions. More specifically, the limit surface in the top right and bottom left corner of a 2-dimensional space. The analytical limit surface is shown in Fig. 8. The scope of this test is to show how the sampling process is able to identify limit surfaces that are topologically more complex than the one presented in Section VI.A.

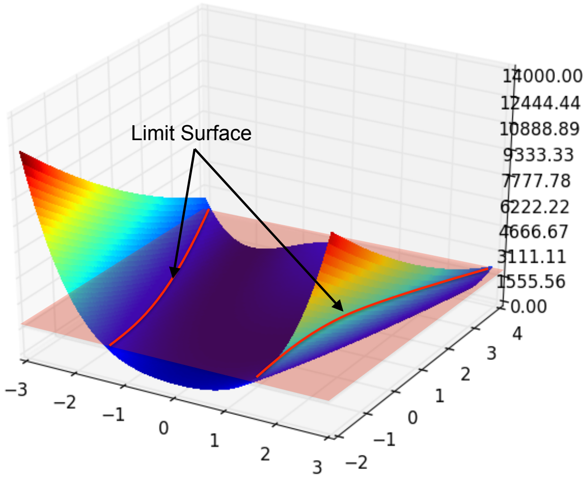


Fig. 8. Analytical shape of the multiple regions limit surface.

The sample locations and the estimated limit surface are shown for different steps of the sampling process, i.e. at iteration 1, 30, 60, 150, 250 and 371 (see Fig. 9) past the training sampling (performed using a 6×6 cartesian grid). For each iteration note how the sample locations are quickly approaching the exact location of the limit surface and the estimated limit surface is converging. Note that for such complex limit surface the required number of samples has increased compared to the case shown in Section VI.A.: 371 vs. 170.

Table 2 compares the number of samples required to evaluate this limit surface by using classical Monte-Carlo and adaptive sampling.

TABLE 2. Number of samples required to evaluate the multiple region limit surface by using classical Monte-Carlo and adaptive sampling (convergence in value is equal to $5\ 10^{-5}$)

	Number of Samples
Monte-Carlo	~10^7
Adaptive	371

VI.C. Convex Region

The third analytical case aims to test the adaptive sampling algorithms to identify convex limit surfaces (i.e., islands). The analytical limit surface is the result of the intersection of the 3-dimensional surface (see Fig. 10):

$$y = x_1^2 + x_2^2 - 0.5 \tag{8}$$

along with the plane $y = 0$ (red plane in Fig. 10). The resulting limit surface is shown as the red line in Fig. 10.

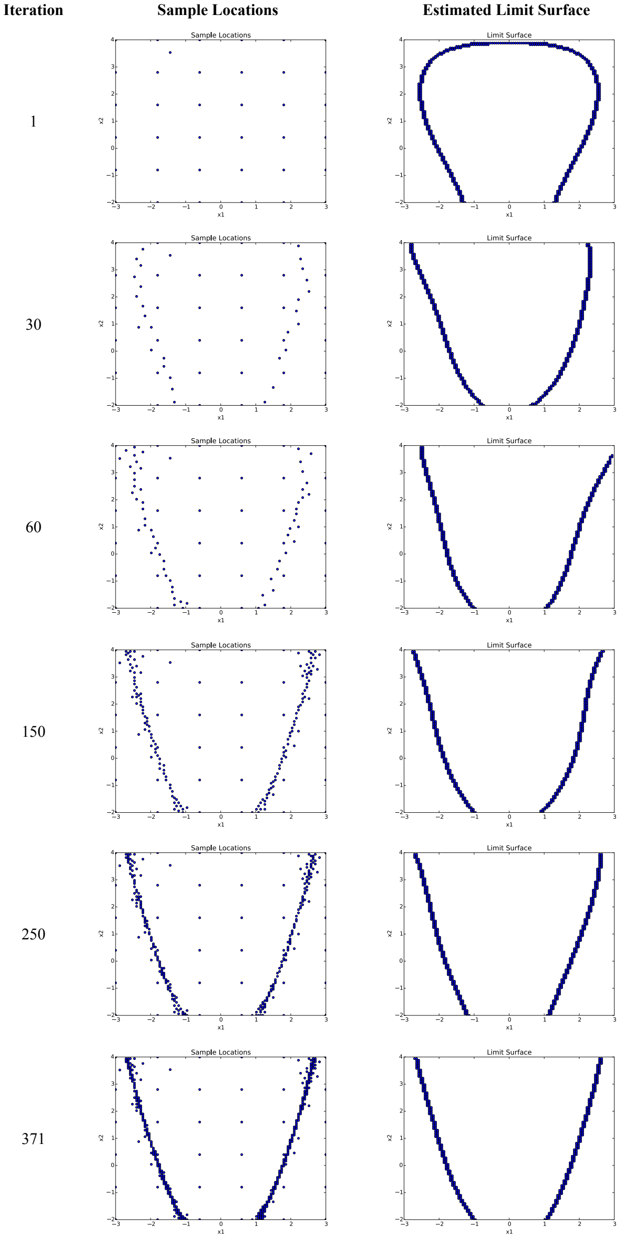


Fig. 9. Multiple regions limit surface: sample locations and the estimated limit surface for different adaptive sampling iterations.

We performed the adaptive sampling analysis for this test case following an initial 6×6 Cartesian grid sampling for training. The sample locations and the estimated limit surface are shown for different steps of the sampling process, i.e. at iteration 1, 30, 60, 150, 250, 350 and 456 (see Fig. 11) past the training sampling. For each iteration, note how the sample locations are quickly approaching the exact location of the limit surface and the estimated limit surface is converging.

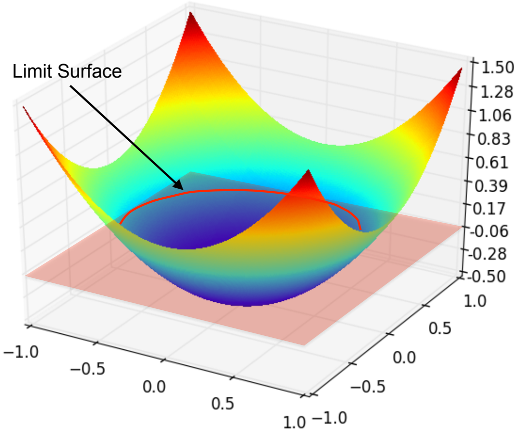


Fig. 10. Analytical shape of the convex limit surface.

Note that for such complex limit surface the required number of samples has increased compared to the case shown in Section VI.A.: 456 vs. 170. Again note such number of samples could drastically decrease for less stringent constraints on convergence criteria.

Table 3 compares the number of samples required to evaluate this limit surface by using classical Monte-Carlo and adaptive sampling.

TABLE 3. Number of samples required to evaluate the convex limit surface by using classical Monte-Carlo and adaptive sampling (convergence in value is equal to 5 10^{-5})

	Number of Samples
Monte-Carlo	~10^7
Adaptive	456

VI.D. RELAP-7 Test Case

The fourth case presented to test the adaptive sampling scheme uses a more realistic application of adaptive sampling using the RISMC toolkit. Here we employed the RELAP-7 PWR system (see Fig. 12) as model coupled to RAVEN to perform adaptive sampling testing.

The scenario considered is a grid-related loss of off-site power (LOOP). In more detail, the scenario is the following (see Fig. 13):

1. An external event (i.e., earthquake) causes the disruption in the power grid and causes a LOOP initiating event; the reactor successfully scrams and, thus, the power generated in the core follows the characteristic exponential decay curve
2. The DGs start and cooling to the core is provided by the Emergency Core Cooling System (ECCS)

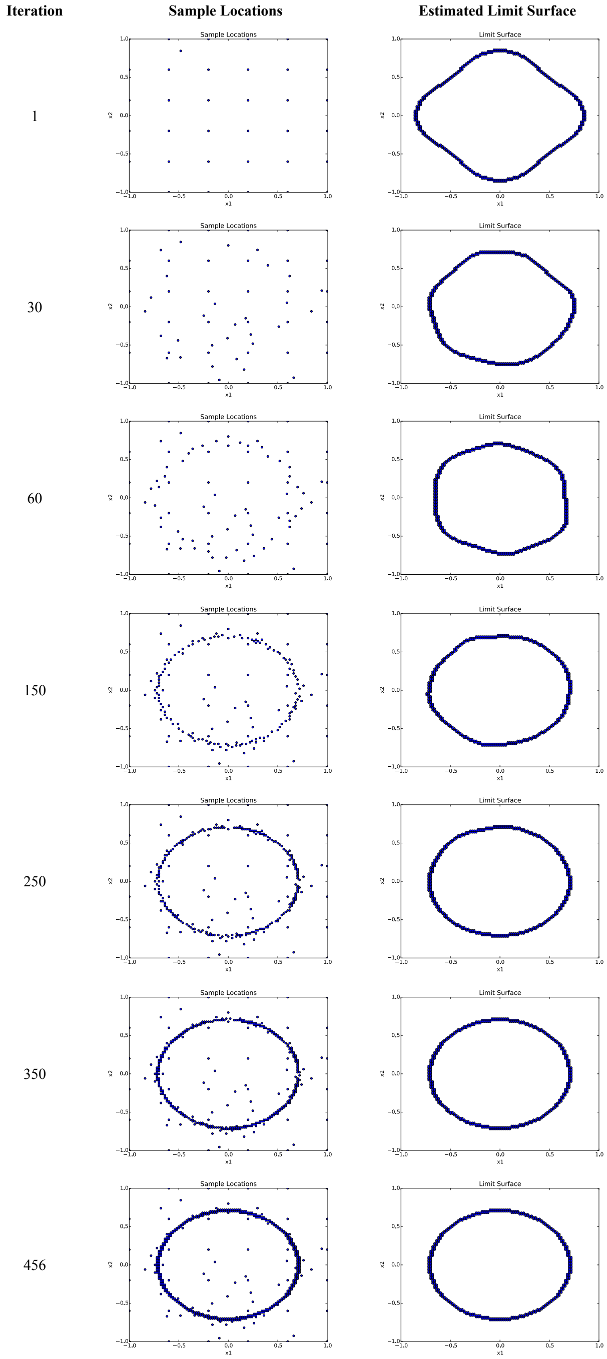


Fig. 11. Convex limit surface: sample locations and the estimated limit surface for different adaptive sampling iterations.

3. At a certain time the DGs fail and, thus, conditions of SBO are reached; ECCS systems is subsequently off-line. Without the ability to cool the reactor core, its temperature starts to rise
4. In order to recover AC electric power, a plant recovery team is assembled in order to recover one of the two DGs
5. If AC power is recovered prior reaching code damage condition (CD), the auxiliary cooling system (i.e., ECCS system) is able to cool the reactor core and, thus, core temperature decreases

In this case, we limit the analysis to two stochastic variables:

1. Time of loss of diesel generators (DGs) after LOOP
2. Recovery time of DGs

The RELAP-7 PWR model has been set up based on the parameters specified in the OECD main steam line break (MSLB) benchmark problem [20]. The reference design for the OECD MSLB benchmark problem is derived from the reactor geometry and operational data of the TMI-1 Nuclear Power Plant (NPP), which is a 2772 MW two loop pressurized water reactor (see the system scheme shown in Fig. 12). An example of PWR SBO scenario generated using RELAP-7 is shown Fig. 13.

For the scope of this article we wanted to show one of the capabilities of RAVEN to generate ROMs and perform statistical analysis on them. For this case we collected the actual simulated data by RELAP-7 in [18], generated a ROM from such data and performed adaptive sampling on the ROM instead of the RELAP-7 code. In more detail, we performed the following steps:

1. Retrieved the hdf5 data generated by sampling RELAP-7 in [18]
2. Trained a ROM given the data retrieved in Step 1
3. Sampled on a 2-dimensional Cartesian grid the ROM obtained in Step 2
4. Performed adaptive sampling and limit-surface search

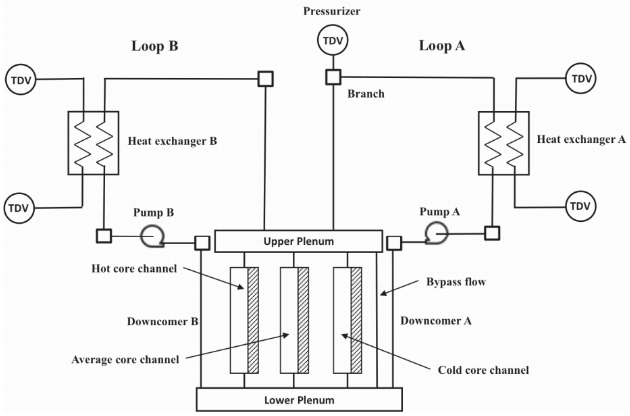


Fig. 12. Scheme of the TMI PWR benchmark.

We performed the adaptive sampling analysis for this test case following an initial 6×6 Cartesian grid sampling for training. The sample locations and the estimated limit surface are shown for different steps of the sampling process, i.e. at iteration 1, 10, 30, 60, 100, 150 and 185 (see Fig. 14) past the training sampling. For each iteration note how the sample locations are quickly approaching the exact location of the limit surface and the estimated limit surface is converging.

Table 4 compares the number of samples required to evaluate this limit surface by using classical Monte-Carlo and adaptive sampling.

We repeated the analysis for the case where the reactor power is set to 120% (i.e., a 20% power uprate). The scope of evaluating this new limit surface is to determine the reduction of the time to recovery for DGs.

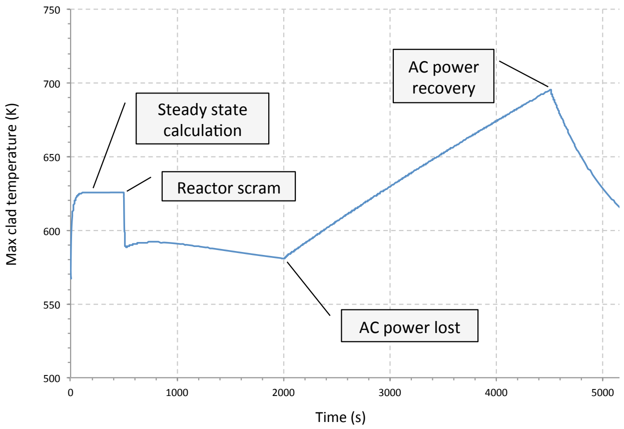


Fig. 13. Example of LOOP scenario followed by DGs failure to run using the RELAP-7 code.

TABLE 4. Number of samples required to evaluate the RELAP-7 PWR SBO limit surface by using classical Monte-Carlo and adaptive sampling (convergence in value is equal to 5 10^{-5})

	Number of Samples
Monte-Carlo	~10^7
Adaptive	185

A 20% reactor power increase implies that clad temperature is increasing at a higher rate and thus the clad is reaching its melting temperature (2200 F) much faster. We performed Steps 1 though 4 for the new data set and evaluated the new limit surface for the 120% test case and the results are shown Fig. 15.

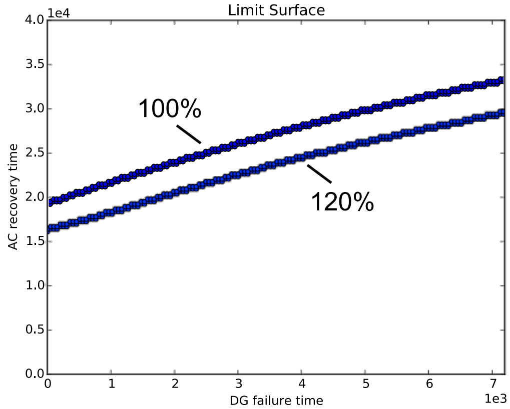


Fig. 14. Limit surface obtained for two different levels of core power: 100% and 120%.

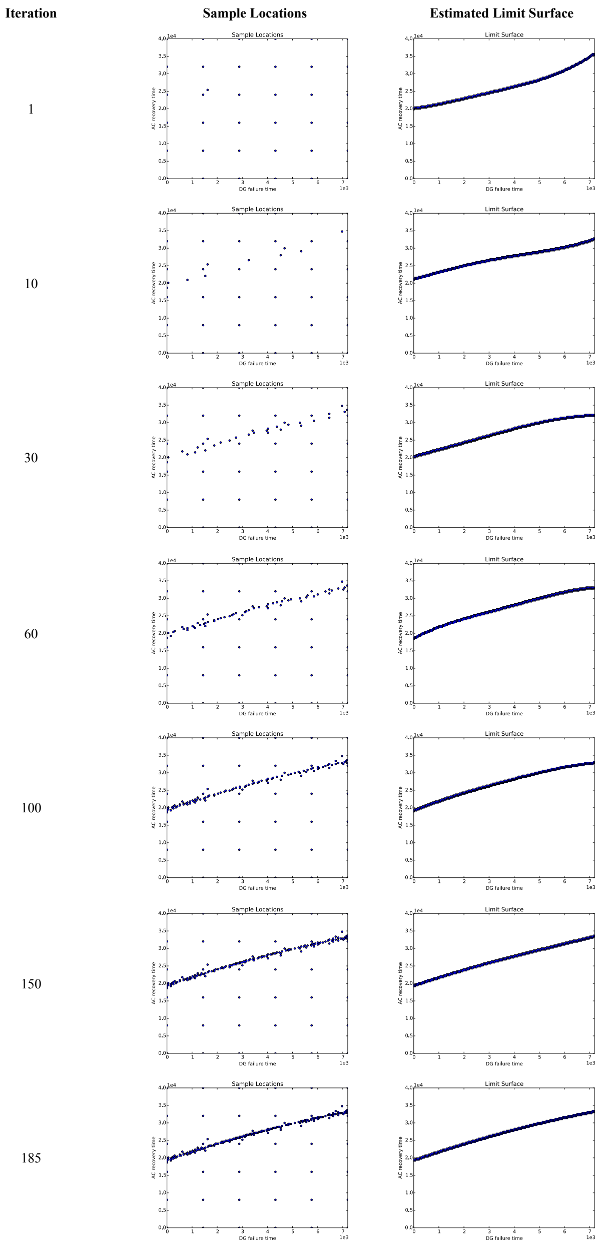


Fig. 15. RELAP-7 limit surface: sample locations and the estimated limit surface for different adaptive sampling iterations.

VII. CONCLUSIONS

In this report we have given an overview of adaptive sampling techniques that can be used to perform PRA analyses using the RISMC toolkit. Classical simulation based approaches rely on either stochastic (e.g., Monte-Carlo or LHS) or deterministic (e.g., DET) sampling. As part of the RISMC Pathway, the type of results that can be obtained via simulation goes beyond the evaluation of probability of occurrence of certain events such core damage and containment breach.

The RISMC approach aims to determine observable outcomes in order to understand potential vulnerabilities and the limitations of the system under consideration. In order to do so, there is a need to deeply explore the space of possible events.

For complex systems such as nuclear power plants, such exploration may require a large number of computationally expensive simulation runs which can be infeasible unless very large high-performance computing resources are used.

Adaptive sampling techniques aim to reduce the computational costs of this kind of analysis by carefully selecting what are the most meaningful simulation runs to be performed. We have shown how such reduction can be achieved for both analytical and more complicated cases. In addition we have shown the kind of information that can be obtained by employing system simulator codes (e.g., RELAP-7) and stochastic analysis tools (e.g., RAVEN) that is unavailable if classical PRA tools (event-tree and fault-tree based) are used.

Classical PRA tools give a limited representation of the system under consideration, for example the timing and sequencing of events is only loosely considered. In [21] we have performed a comparison on classical and RISMC PRA analyses for a BWR SBO test case and we have shown not only the greater amount of information that can be obtained using the RISMC approach but also major differences regarding probability of occurrences of certain event sequences.

REFERENCES

1. C. SMITH, C. RABITI, AND R. MARTINEAU, "Risk Informed Safety Margins Characterization (RISMC) Pathway Technical Program Plan", Idaho National Laboratory INL/EXT-11-22977 (2011).
2. DOE-NE Light Water Reactor Sustainability Program and EPRI Long-Term Operations Program – Joint Research and Development Plan, Revision 4, INL-EXT-12-24562, April 2015.
3. U.S. NRC, NUREG 1150, "Severe accident risks: an assessment for five U.S. nuclear power plants," Division of Systems Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC (1990).
4. M. E. PATE-CORNELL, "Fault Trees vs. Event Trees in Reliability Analysis", *Risk Analysis,* **4**, no. 3 (1984).
5. A. ALFONSI, C. RABITI, D. MANDELLI, J. COGLIATI, AND R. KINOSHITA, "Raven as a tool for dynamic probabilistic risk assessment: Software overview," in *Proceeding of M&C2013 International Topical Meeting on Mathematics and Computation*, CD-ROM, American Nuclear Society, LaGrange Park, IL (2013).
6. RELAP5 Code Development Team. RELAP5-3D Code Manual. INL, (2012).
7. A. DAVID, R. BERRY, D. GASTON, R. MARTINEAU, J. PETERSON, H. ZHANG, H. ZHAO, L. ZOU, "RELAP-7 Level 2 Milestone Report: Demonstration of a Steady State Single Phase PWR Simulation with RELAP-7," Idaho National Laboratory: INL/EXT-12-25924 (2012).
8. B. SPENCER, Y. ZHANG, P. CHAKRABORTY, S.B. BINER, M. BACKMAN, B. WIRTH, S. NOVASCONE, J. HALES, "Grizzly Year-End Progress Report", Idaho National Laboratory report: INL/EXT-13-30316, Revision 0 (2013).
9. C. RABITI, D. MANDELLI, A. ALFONSI, J. COGLIATI, AND B. KINOSHITA, "Mathematical framework for the analysis of dynamic stochastic systems with the raven code," in *Proceedings of International Conference of mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, Sun Valley (Idaho) (2013).
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*, **47**, pp. 371-382, (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*, (2012).
14. N. S. ALTMAN, "An introduction to kernel and nearest-neighbor nonparametric regression", The American Statistician, 46 – 3, pp. 175–185 (1992).
15. R. TRUDEAU, Introduction to Graph Theory, Dover Publications New York (1993).
16. 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).
17. D. MALJOVEC, B. WANG, D. MANDELLI, P.-T. BREMER AND V. PASCUCCI, "Adaptive Sampling Algorithms for Probabilistic Risk Assessment of Nuclear Simulations," *in Proceeding of International*

Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA) (2013).

18. C. SMITH, D. MANDELLI, S. PRESCOTT, A. ALFONSI, C. RABITI, J. COGLIATI, AND R. KINOSHITA, "Analysis of pressurized water reactor station blackout caused by external flooding using the RISMC toolkit," Idaho National Laboratory technical report: INL/EXT-14-32906 (2014).

19. C. J. C. BURGES, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov*. **2**-2, pp. 121–167 (Jun. 1998).

20. "Pressurized Water Reactor Main Steam Line Break (MSLB) Benchmark", Volume I: Final Specifications, NEA/NSC/DOC(99)8.

21. D. MANDELLI, Z. MA, AND C. SMITH, "Dynamic and classical PRA: a BWR SBO case comparison," 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.