

## FUEL RELIABILITY ANALYSIS USING BISON AND RAVEN

C. Rabiti, J. Cogliati, G Pastore, R. J Gardner, A. Alfonsi

*Idaho National Laboratory, 2525 Fremont Ave,  
Idaho Falls, ID, 83401  
cristian.rabiti@inl.gov*

*The investigation of operational limits for nuclear fuel is a challenging subject and at the same time important to ensure reliability and safety of the nuclear energy production. The parametric and probabilistic analysis of failure is a very expensive process that might require thousands of simulations while exploring the input space by altering the input parameters. This paper will demonstrate how RAVEN is a suitable tool to perform such analysis. While RAVEN is fully capable to perform the needed parametric analysis, it can also use artificial intelligence to speed up the reliability/safety evaluation. In particular artificial intelligence algorithms are used to accelerate the search of the limit/reliability surfaces.*

*This paper will review the concept of limit surface and its numerical representation, and will explain how support vector machine type algorithms are used to speed up the limit surface search. The limit surface location is then used to perform evaluation of the failure probability. This last step is examined in detail from the point of view of its numerical implementation.*

### I. INTRODUCTION

The investigation of operational limits for nuclear fuel is a challenging subject and at the same time important to ensure reliability and safety of the nuclear energy production. The simulation tools used to perform the analysis, presented in this paper, are the RAVEN<sup>1,2</sup> and the BISON<sup>3</sup> codes. RAVEN is a code developed at the Idaho National Laboratory (INL) founded by the Nuclear Energy Advanced Modeling and Simulation<sup>4</sup> (NEAMS) DOE program in support of the Light Water Reactor Sustainability<sup>5</sup> DOE campaign. BISON is a fuel performance code developed at INL, currently sponsored by several DOE program as NEAMS and the Consortium for Advanced Simulation of Light water reactors<sup>6</sup> (CASL).

The analysis of a fuel rod undergoing a power transient is considered, and 3 input variables are selected for the calculation of the limit surface, namely, fuel grain radius, rod linear power, and fuel thermal expansion coefficient.

The limit surface sought in this work is the 3-dimensional surface in the parametric input space, identified by the values of the above input variables for which the maximum Von Mises stress in the clad does not exceed an imposed threshold. The search of the limit surface (aka reliability surface) location could be very computationally expensive; for this reason, RAVEN provides the capability to spawn several BISON runs in parallel and accelerate the search of the limit surface by means of Support Vector Machines (SVM). SVMs are a class of artificial intelligence algorithms mostly used as classifiers (forecast Boolean outcomes). Few initial runs of BISON are used to train the SVM algorithm that is used to forecast which regions of the parametric input space are inside/outside the reliability surface. This forecast is used to decide where further exploration of the input space is needed. The search stops when the exploration of additional points of the input space will not lead to changes in the location of the limit surface.

Once the limit surface is found, RAVEN will compute the probability of exceeding the maximum Von Mises stress by performing an integral of the volume surrounded by the input space weighted by the probability associated to the value of the input parameters. The limit surface so obtained will be also used to help the interpretation of the physical phenomena relating the input space and the max Von Mises stress in the clad.

### I. LIMIT/REALIABILITY SURFACE

#### I.A. Limit Surface Definition

Let's consider a system, which could be represented in the phase space by:

$$\bar{y} = H(\bar{x}, t, \bar{p}) \quad (1)$$

Where  $\bar{y}$  is the coordinate representing the system in the phase space, and  $(\bar{x}, t, \bar{p})$  are the independent variables that have been separated in spatial, temporal and parameters independent variables, respectively,  $(\bar{x}, t, p)$ . The distinction between  $(\bar{x}, t, \bar{p})$  is based solely on engineering considerations, while no mathematical differentiation exists .

Now it is possible to introduce the failure function,  $G$ .  $G$  is a Boolean function that – based on the values of the system phase space coordinate, time, spatial coordinate, and possible system parameters – has either the value 0 to indicate system properly functioning and 1 for system failure:

$$G = G(\bar{y}, \bar{x}, t, \bar{p}) = G(H(\bar{x}, t, \bar{p}), \bar{x}, t, \bar{p}) = G(\bar{x}, t, \bar{p})$$

For simplicity it is possible to assume, without loss of generality, that  $F$  is time independent (e.g.  $G = \int_{t_0}^{t_{end}} dt f(\bar{x}, t, \bar{p})$ ).

$$G = G(\bar{x}, \bar{p}) \quad (2)$$

Now, two more hypotheses are needed to properly introduce the concept of limit surface.

First, the equation describing the time evolution of probability density function of the system coordinate in the phase space is of type Liouville, which allows ensuring that all the stochastic behaviors of the system are representable as probability distribution function in the input space. A heurist analysis of such condition is reported in reference 7 while the process of how the phase space could be expanded to accomplish such condition is reported in reference 8. This allows combining the parametric space with the initial condition space:

$$\begin{aligned} (\bar{x}) &\leftarrow (\bar{x}, \bar{p}) \\ G(\bar{x}) &\leftarrow G(\bar{x}, \bar{p}) \end{aligned} \quad (3)$$

Second, the function describing the system  $H(\bar{x}, t)$  should satisfy certain regularity requirement.

While the second condition is practically always true for the physical systems the first condition could be violated in a very specific set of problems as illustrated in reference 8, but usually not relevant for the fuel reliability analysis.

Under the above simplifications it is possible to identify the region of the input space ( $V$ ) leading to a specific outcome of the goal function. In particular we can define the failure region  $V_F$  as the region of the input space where  $G = 1$ , more specifically:

$$V_F = \{\forall \bar{x} | G(\bar{x}) = 1\} \quad (4)$$

The definition of the complementary of the failure region is of course:

$$V_F^c = \{\forall \bar{x} | G(\bar{x}) = 0\} \quad (5)$$

Its boundary is the sought limit surface:

$$L_s = \partial V_F = \partial \{\forall \bar{x} | G(\bar{x}) = 1\}$$

It is worth mentioning that  $V_F$  could be a disjoint subset of  $V$ , and unless both  $V_F$  and its complementary are not disjoint sets  $L_s$  is also a disjoint set.

### I.A. Property of the Limit Surface

Clearly, determining the location of the limit surface corresponds to the capability to identify bounding regions for which the system will not exceed the operative limit.

The location of the limit surface is important for design optimization but could also be used to extract more information to characterize the behavior of the system

from a probabilistic standpoint. For the moment lets assume that all the input space is subject to probabilistic behavior (in case some variable are not, the following derivation should be limited to only the ones that are). To be more exact let be  $\bar{x} \in V$  and  $\bar{x} \sim \bar{X}$  (i.e.  $\bar{x}$  is the random variate realization of the random variable  $\bar{X}$ ).

If  $f_{\bar{X}}(\bar{x})$  is the probability distribution function (pdf) of  $\bar{X}$  the failure probability of the system or the probability of exceeding the operative limits ( $P_F$ ) is:

$$P_F = \int_V d\bar{x} G(\bar{x}) f_{\bar{X}}(\bar{x}) = \int_{V_F + V_F^c} d\bar{x} G(\bar{x}) f_{\bar{X}}(\bar{x}) \quad (6)$$

By the definition of the  $V_F$  and  $V_F^c$  sets:

$$P_F = \int_{V_F} d\bar{x} f_{\bar{X}}(\bar{x}) \quad (7)$$

To summarize, the failure probability of the system is equal to the probability of being in the input space leading to a possible failure pattern that is also equal to the volume, probability weighted, surrounded by the limit surface.

To exemplify the concept, an example could be helpful. Lets consider the following input space:

- $h_F$ : level of the Von Mises stress for which the clad would fail
- $p$ : power scaling factor

And the corresponding cumulative distribution functions:

$$h_F \sim f_{H_F}(h_F) = \begin{cases} = 0 \text{ if } h_F < h_{F_{min}} \\ = \frac{1}{h_{F_{max}} - h_{F_{min}}} \\ = 0 \text{ if } h_F > h_{F_{max}} \end{cases} \quad (8)$$

$$p \sim f_p(p) = \begin{cases} = 0 \text{ if } p < p_{min} \\ = \frac{1}{p_{max} - p_{min}} \\ = 0 \text{ if } p > p_{max} \end{cases} \quad (9)$$

Assuming that the Von Mises stress ( $h$ ) is a linear function of the power (just to illustrate the integration domain in a simple case):

$$h = h_0 + \alpha p \quad (10)$$

To further specify the case (otherwise several alternative would be possible):

$$\begin{aligned} h_{F_{min}} &> h_0 + \alpha p_{min} \\ h_{F_{max}} &< h_0 + \alpha p_{max} \end{aligned} \quad (11)$$

The limit surface, failure region, and active part of the failure region (failure region with non zero probability) are illustrated in figure 1 (in agreement with the above assumptions):

In this simplified case the failure probability could be evaluated as it follows:

$$P_F = \int_{V_F} d\bar{x} f_{\bar{X}}(\bar{x}) = \int_0^{+\infty} dh_F f_{H_F}(h_F) \int_{\frac{h_F - h_0}{\alpha}}^{+\infty} dp f_p(p) \quad (12)$$

The integral domain could be restrain to the areas with not zero contribution:

$$P_F = \int_{h_{F_{min}}}^{h_{F_{max}}} dh_F \frac{1}{h_{F_{max}} - h_{F_{min}}} \int_{\frac{h_F - h_0}{\alpha}}^{p_{max}} dp \frac{1}{p_{max} - p_{min}}$$

$$P_F = \frac{2(\alpha p_{max} + h_0) - (h_{Fmax} + h_{Fmin})}{2\alpha(p_{max} - p_{min})} \quad (13)$$

While the analytical solution carries low importance per se, this simple example is useful to grasp how the limit surface is defined in a practical case and how the hyper-volume should be properly weighted by the probability in the input space.

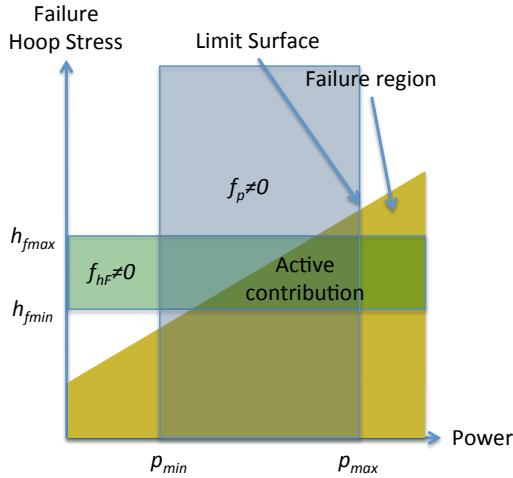


Figure 1: Limit surface and active integration domain.

## I. NUMERICAL SEARCH OF THE LIMIT SURFACE

**II.A. Numerical Representation of The Limit Surface**

Similarly to every numerical process the location of the limit surface would be known given a certain tolerance derived from its numerical representation. The tolerance is given by the grid that is used to locate the limit surface. To simplify the explanation, lets assume that each of the  $N$  dimensions of the input space is discretized with the same number of equally spaced point  $M$ . The Cartesian grid so built has therefore  $M^N$  points and each point is uniquely characterized by the vector index  $\bar{j} = (j_1, \dots, j_i, \dots, j_N)$  with  $j_i \leq M \forall i$ , and for each dimension is defined the  $h_i$  step size. Assuming that the goal function  $G(\bar{x})$  is known on each point of the grid, and a zero-th order approximation is used the approximation of the goal function could be written as:

$$G(\bar{x}) = \sum_{j=\bar{l}}^{\bar{M}} \theta_j(\bar{x}) G(\bar{x}_j) \quad (14)$$

Where  $\theta_j$  is the characteristic function of the hyper-volume surrounding the point  $\bar{x}_j$ :

$$\theta_j(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x} \in \prod_{i=1}^N [x_{j_i} - h_i/2, x_{j_i} + h_i/2] \\ 0, & \text{if } \bar{x} \notin \prod_{i=1}^N [x_{j_i} - h_i/2, x_{j_i} + h_i/2] \end{cases} \quad (15)$$

$\bar{l}$  is the vector  $(1, \dots, 1)$ , and  $\bar{M} = (M, \dots, M)$

Same approximation is used to represent the probability of the parameters in the input space:

$$f_{\bar{X}}(\bar{x}) = \sum_{j=\bar{l}}^{\bar{M}} \theta_j(\bar{x}) f_{\bar{X}}(\bar{x}_j) \quad (16)$$

Using the above-described approximations the failure probability could be expressed as:

$$P_F = (\prod_{i=1}^N h_i) (\sum_{j=\bar{l}}^{\bar{M}} f_{\bar{X}}(\bar{x}_j) G(\bar{x}_j)) \quad (17)$$

The formula needs proper adjustment to account for the edges of the grid where the weight it is not anymore the full hyper-volume  $\prod_{i=1}^N h_i$  but needs to be reduced to account for the half size cells.

A question that has not yet been addressed is the accuracy of this approach. Unfortunately the goal function and in some cases also the probability distribution functions are discontinuous and therefore an estimation of the asymptotic error bound is not easily achievable by using higher order Taylor expansions.

Under some assumption the following derivation could be used to seek some error bounds for the accuracy of the probability integral.

Firstly, it is needed to exclude the case of discrete parameters in the input space, second, when considering truncated distributions the grid should be constructed such as the discontinuous points lay on the edges of the grid.

These choices bring the definition of the active contribution hyper-volume directly into the definition of the integration space eliminating the need to account for possible discontinuities in the probability distribution functions. If the region of the input space limited to the support of the distribution functions of the input parameters is defined as the active hyper-space ( $V_A$ ) the following holds:

$$\begin{aligned} P_F &= \int_V d\bar{x} G(\bar{x}) f_{\bar{X}}(\bar{x}) = \\ &\int_{V_A} d\bar{x} G(\bar{x}) \left[ \sum_{j=\bar{l}}^{\bar{M}} \theta_j(\bar{x}) \left( f_{\bar{X}}(\bar{x}_j) + \sum_{i=1}^N \frac{\partial f_{\bar{X}}}{\partial x_i} \Big|_{\bar{x}_j} (x_i - x_{j_i}) \right) \right] \end{aligned}$$

$$P_F = \sum_{j=\bar{l}}^{\bar{M}} \int_{\bar{x}_j - h/2}^{\bar{x}_j + h/2} d\bar{x} G(\bar{x}) \left( f_{\bar{X}}(\bar{x}_j) + \sum_{i=1}^N \frac{\partial f_{\bar{X}}}{\partial x_i} \Big|_{\bar{x}_j} (x_i - x_{j_i}) \right) \quad (18)$$

Now it will be useful to split the point of the grid, already in the active part of the domain, between the ones having at least neighbor with a different value of the goal function ( $V_A \cap V_{\partial V_F}$ ), the ones in the active domain with zero value of the goal function and surrounded by point with the same value of the goal function  $V_A \cap V_{G(\bar{x})=0}$ , and the points in the active domain with value of one of the goal function and surrounded by point with the same value of the goal function  $V_A \cap V_{G(\bar{x})=1}$ .

Now the two following relationships hold:

$$\begin{aligned}
& \sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{G(\bar{x})}=0}}^{\bar{N}} \int_{\bar{x}_j - \bar{h}/2}^{\bar{x}_j + \bar{h}/2} d\bar{x} G(\bar{x}) \left( f_{\bar{X}}(\bar{x}_j) \right. \\
& \quad \left. + \sum_{i=1}^N \frac{\partial f_{\bar{X}}}{\partial x_i} \Big|_{\bar{x}_j} (x_i - x_{j_i}) \right) = 0 \\
& \sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{G(\bar{x})}=1}}^{\bar{N}} \int_{\bar{x}_j - \bar{h}/2}^{\bar{x}_j + \bar{h}/2} d\bar{x} G(\bar{x}) \left( f_{\bar{X}}(\bar{x}_j) + \sum_{i=1}^N \frac{\partial f_{\bar{X}}}{\partial x_i} \Big|_{\bar{x}_j} (x_i - \right. \\
& \quad \left. x_{j_i}) \right) = \\
& \sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{G(\bar{x})}=1}}^{\bar{N}} \int_{\bar{x}_j - \bar{h}/2}^{\bar{x}_j + \bar{h}/2} d\bar{x} \left( f_{\bar{X}}(\bar{x}_j) + \sum_{i=1}^N \frac{\partial f_{\bar{X}}}{\partial x_i} \Big|_{\bar{x}_j} (x_i - \right. \\
& \quad \left. x_{j_i}) \right) \quad (19)
\end{aligned}$$

Consequently:

$$\begin{aligned}
P_F &= \sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{G(\bar{x})}=1}}^{\bar{N}} (\prod_{i=1}^N h_i) f_{\bar{X}}(\bar{x}_j) + O(h^{N+1}) + \\
&\sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{\partial V_F}}}^{\bar{N}} \int_{\bar{x}_j - \bar{h}/2}^{\bar{x}_j + \bar{h}/2} d\bar{x} G(\bar{x}) \left( f_{\bar{X}}(\bar{x}_j) + \sum_{i=1}^N \frac{\partial f_{\bar{X}}}{\partial x_i} \Big|_{\bar{x}_j} (x_i - \right. \\
&\quad \left. x_{j_i}) \right) \quad (20)
\end{aligned}$$

It is unfortunate that the second summation is a  $O(h^N)$  given the fact that the discontinuity of  $G(\bar{x})$  does not allow the Taylor expansion to converge consequently in a worst-case scenario analysis:

$$\begin{aligned}
P_F &= \sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{G(\bar{x})}=1}}^{\bar{N}} (\prod_{i=1}^N h_i) f_{\bar{X}}(\bar{x}_j) + O(h^{N+1}) + \\
O(h^N) &\approx O(h^N) \quad (21)
\end{aligned}$$

concluding that the process could be completely undetermined. In reality this is not true if

$$\begin{aligned}
&\sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{G(\bar{x})}=1}}^{\bar{N}} (\prod_{i=1}^N h_i) f_{\bar{X}}(\bar{x}_j) \gg \\
&\sum_{\substack{j=\bar{I} \\ \bar{x}_j \in V_A \cap V_{\partial V_F}}}^{\bar{N}} \left| \int_{\bar{x}_j - \bar{h}/2}^{\bar{x}_j + \bar{h}/2} d\bar{x} f_{\bar{X}}(\bar{x}_j) \right| \quad (22)
\end{aligned}$$

This condition is satisfied when the volume to surface ratio is in favor of the volume and when the probability distribution function is not strongly peaked around the limit surface.

Actually to take advantage of the last observation the grid is built in the probability transformed space. In this space the measure  $d\bar{x}$  is replaced by  $d\bar{\mu} = f_{\bar{X}} d\bar{x}$ .

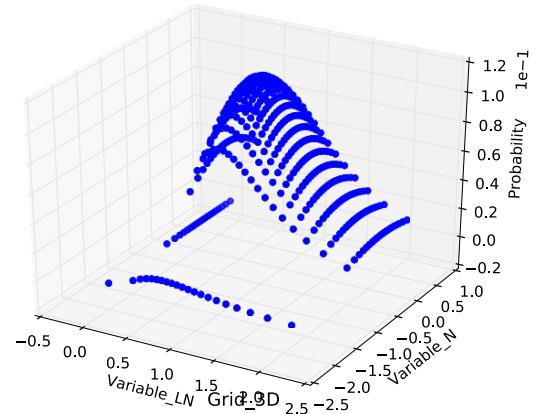
The equally spaced grid, in the new metrics, corresponds to a grid where the distance between the points is equally spaced in probability. The advantage of this approach is that the last inequality is reduced to the condition:

*Number of grid point on the failure region  
 >> number of point on the failure region frontier*

In conclusion, when using the equally probability spaced grid the error is bounded by the amount of probability contained in each grid cell times the number of cell on the frontier of the limit surface. It is worth to notice that given that the integral of the probability on the full domain of the input space is always one the probability content per computational cell is:

$$P_{\bar{J}} = \frac{1}{M^N} \quad (23)$$

Figure 2 illustrates how the point on regular grid in value or in probability will differ. The blue dots are the location of the grid nodes. The Variable\_LN uses an equally spaced grid in value and its probability distribution function is a LogNormal. The Variable\_N uses an equally probability spacing approach using a normal distribution as weighting function. In this case in the Z axis it is reported the product of the two probabilities.



**Figure 2: Comparison of equally value and equally spaced grid points.**

## II.B. Acceleration of the Limit Surface Search

So far it has been determined the meaning, and the possible numerical approximation of the limit surface but its search still remains an open challenge.

Each evaluation of the goal function in one of the grid point implies the evaluation of the BISON code (or any other) for a set of entry in the input space. Each BISON run could take several hours and, even if RAVEN would handle the parallel sampling of the grid points, the number of runs needed could be in the thousands to contain the error and this could turn out to be too computationally expensive. This process is accelerated by means of usage of supervised learning algorithms.

A supervised learning algorithm, in short and in our context, is a set of equations which parameters could be ‘trained’ so to reproduce with a certain amount of confidence the response (output) of a much more complex system of equation modeling a physical system (in or case

BISON). The output that the supervised learned algorithm tries to reproduce is the value of the goal function. Being this type of output a Boolean value, a sub class of supervised learning algorithms are used that are usually referred to as classifiers.

RAVEN has several different types of classifiers. The most successfully tested during these scoping studies where the Support Vector Machines (SVM) with radial basis function based kernels.

What RAVEN implements is, in reality, what is commonly referred to as a active learning process which ultimate results is the training of a classifiers capable to predict the outcome of the goal function for any given point of the input space.

In an active learning process a supervised learning algorithm is combined with a criteria to choose the next point of the input space to explore using the complex physical model. This process is repeated until, under a proper metric, the prediction capabilities of the supervised learning algorithm will not improve by adding further training points.

In more detail the iterative scheme could be so described:

1. Generation of a set of point in the input space and computation of the corresponding goal function value. This constitutes what is usually referred to as the initial training set
 
$$\{\bar{x}_k\}, k = 1, \dots, \text{size training set}$$

$$\{G_k = G_k(\bar{x}_k)\}, k = 1, \dots, \text{size training set}$$
2. The pairs  $\{(\bar{x}_0, G_0)\}_k$  are used to train the SVM ( $\tilde{G} = \tilde{G}(\bar{x})$ )
3. The SVM is evaluated on each point of the grid  $\bar{J}$  leading to the determination of  $\tilde{G}_{\bar{j}}$ , for  $\bar{j} = \bar{I}, \dots, (M, \dots, M)$
4. The values of  $\tilde{G}_{\bar{j}}$  are used to determine the predicted location of the limit surface
5. A new point is chosen to be added to the training set and a new pair is generated and added to the training set. The algorithm for the choose of the next point is explained in the next sections
6. The procedure is repeated starting from point 3 until convergence is achieved. The convergence is achieved when there is no changes in the location of the limit surface between a certain number of consecutive iteration

A final remark on the limit surface searching algorithm is about the requirement of location stability for more than two consecutive iterations. The reason for this choice is determine by the attempt to limit the effect of the build of non-linear bias in the searching pattern. In fact as it will be explained in the next paragraph the searching algorithm might focus too intensively on certain region of the limit surface while putting too few point in

other zones and completely hiding undiscovered topological features of the limit surface.

Another concern that will need to be addressed in the future is the impact of the convergence test on the projection of the limit surface obtained by surrogate model. In fact when the stability of the limit surface is tested in reality what is tested is the stability of the prediction of the surrogate model, which is equivalent to the projection of the original problem into a lower dimensional one. Most of the time practically all the points located on the limit surface are tested if a reasonably large number is used for the persistence test (few tens) and therefore in this case the convergence test on the limit surface succeed only when the location within the tolerance of the grid, is exact.

#### *I.B.1. Selection of the next training point*

The natural choice for the selection of the next point to be explored is given by the point already predicted as being part of the limit surface. This is a direct consequence of choosing, as convergence criteria, the stability of the location of the limit surface.

Among the points located on the limit surface two competing criteria are used. The point on the limit surface are ranked based on the distance from the closest training point already explored (the larger is the distance the higher is the score for the candidate point), and based on its persistence (the larger is the number of time the prediction of the goal function for that point have changed the higher is the score).

The two competing ranking system are combined as it follows (z is the iteration index):

$$(Score)_{\bar{j}, z} = \left\{ \begin{array}{l} \text{Distance closest} \\ \text{training point} \end{array} \right\} * \left\{ \frac{1}{\text{persistence}_z} \right\}$$

$$(persistence)_{\bar{j}, z+1} =$$

$$\text{persistence}_z + \left\{ \begin{array}{l} 1 \text{ if } \tilde{G}_z(\bar{x}_{\bar{j}}) = \tilde{G}_{z+1}(\bar{x}_{\bar{j}}) \\ 0 \text{ if } \tilde{G}_z(\bar{x}_{\bar{j}}) \neq \tilde{G}_{z+1}(\bar{x}_{\bar{j}}) \end{array} \right. \quad (24)$$

Since this approach creates a set of prioritized candidates could be used also in the parallel implementation of the searching algorithm. When several training point re run in parallel, it could happen that the evaluation of one additional point does not alter substantially the location of the limit surface. Consequently the new candidate with the highest score is being already submitted for evaluation, just its computation has not being yet completed. In this case, to avoid to submit two time the same evaluation point the code unsure that the point has not yet been selected and in case moves down along the point ranking until it finds a point that has not been explored yet. In the unlikely case that all possible point on the limit surface have been explored or submitted for evaluation the code will use a Monte Carlo sampling strategy to pick the next point.

### III. EXAMPLE

#### III.A. BISON Test Case Description

The case examined is a simplified axis-symmetric LWR fuel rodlet, composed by ten UO<sub>2</sub> pellet, Zr-4 cladding, gap and upper plenum. The problem is power rump up transient in which the linear power changes according to Table 1. The fully detailed description of the case geometry could be found in section 3.1.

TABLE I. Linear Power Time Evolution

Time (s)	Linear Power (W/m)
0	0
10000	3.50E+04
150000000	3.50E+04

What has been investigated is the maximum Von Mises Stress dependence from three input parameters.

The input parameters and their assumed probabilistic distribution is given in table 2.

TABLE II. Probabilistic Input Description

Variable	Distribution	Min	Max
Grain Radius	Uniform	0.4	1.5
Fuel Thermal Expansion Coefficient	Uniform	9E-6	1.1E-5
Linear Power Scaling Factor	Uniform	0.95	1.05

#### III.B. BISON Test Case Results

##### III.B.1. Parametric studies

To perform parametric studies the most natural exploration techniques of the input space is the grid-based exploration. In the case here analyzed the grid chosen was a equally probability spaced grid with 21 points by dimension leading to a total of 9261 sampling point.

Figure 3 to 5 illustrate the maximum Von Mises stress as a function of the input variable sampled. From the 4-D figures (Figure 3 and 4) it is possible to appreciate the overall behavior of the stress. The two figures differ with respect the variable that is used in the fourth dimension and leading to multiple value of the max Von Mises stress for the same point in the input space. Figure 5 represent instead a projection where all the input variables have been eliminated except the initial grain radius. The figure reveals the existence of a non linear relationship between the grain radius and the max Von Mises stress, that happens (comparing also figure 4) at higher levels of power.

The relative impact of the different input parameters toward the maximum Von Mises stress could be aided by the knowledge of the sensitivity coefficients. Sensitivity

coefficients are simply the derivative of the figure of merit (max Von Mises stress) with respect the input parameters. Table III reports such coefficients computed by RAVEN using a linear regressor on the values computed on the grid points.

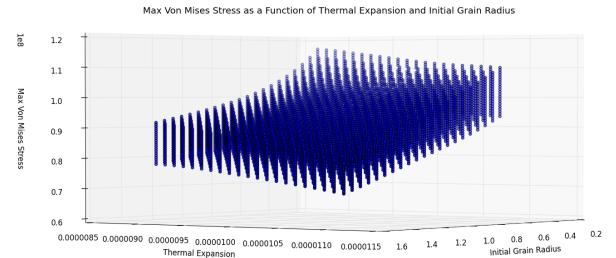


Figure 3: Max Von Mises stress as a function of the initial grain radius, and the fuel thermal expansion coefficient (hidden variable power scaling factor).

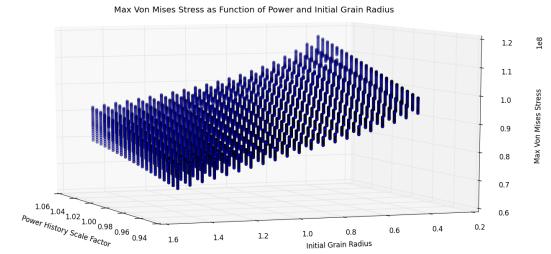


Figure 4: Max Von Mises stress as a function of the power scaling factor and initial grain radius (hidden variable fuel thermal expansion coefficient).

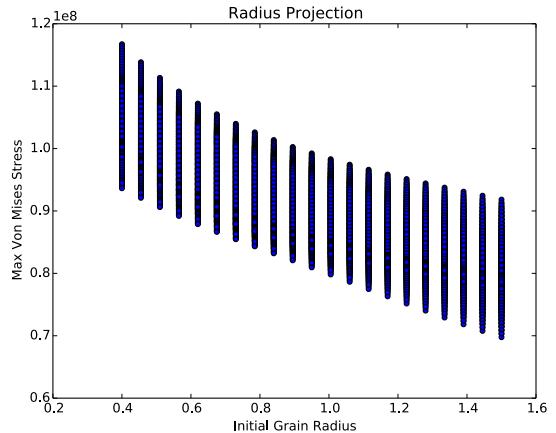


Figure 5: Max Von Mises stress as a function of the initial grain radius (hidden variables fuel thermal expansion coefficient and power scaling factor).

TABLE III. Sensitivity Coefficients

Input	Max Von Mises Stress
Power Scaling Coefficient	138596387
Grain Radius Scaling Coefficient	-20106223.4
Thermal Expansion Coefficient	-2762537190000

### III.B.2. Limit Surface Analysis

As already mentioned the limit surface search has been accelerated by means of an SVM with a radial basis function kernel. The grid has been build in probability requiring that each grid cell have a probability content of  $10^{-5}$  this is equivalent to generate an equally spaced grid (using uniform distribution) lead to grid that are both equally spaced in value and equally spaced in probability with  $\sqrt[3]{10^{-5}} \sim 47$  cell by input variable. The parameters used for the setting of the SVM and its kernel were rather standards: gamma=0.1, tolerance =0.01, and penalty factor C=10. Gamma express how strong is the influence of close by training points versus the one far away (the largest is gamma the stronger are the closest point) in determining the prediction, tolerance set the convergence criteria for the iterative search of the SVM parameters, and C is a smoothing penalty function to denoise the data. Of course more information and more specific reference could be found in the scikit learn manual<sup>9</sup>.

Figure 6 shows with respect the point sampled by the grid approach the location of the point sampled by the adaptive search of the limit surface and figure 7 shows the limit surface location. The limit surface is searched such as the maximum Von Mises stress does not exceed the value of 1.05e8. The corresponding failure probability computed was 5.1875E-02.

It is interesting to note that the limit surface slope confirm the signs of the sensitivity coefficient previously computed.

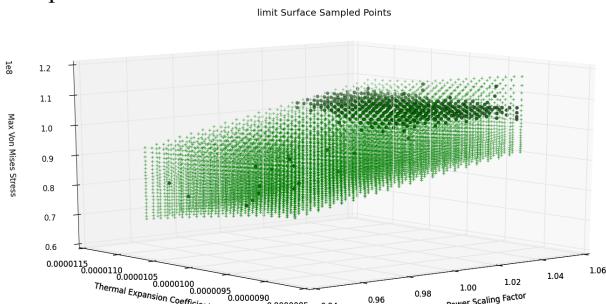


Figure 6: Points sampled by the adaptive algorithm (dots) vs. the grid based sampling (crosses).

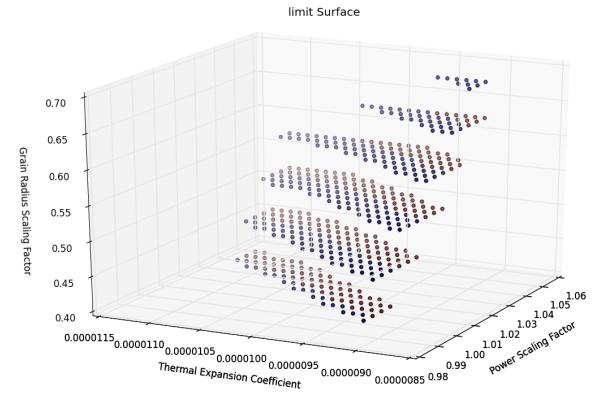


Figure 7: Limit surface location, in blue the side in the failure region, in red the side in the safe region.

## IV. CONCLUSIONS

An initial demonstration of the potential of the RAVEN probabilistic analysis code to help the design process and reliability analysis of the nuclear fuel has been presented in this paper. In particular, RAVEN has been coupled to the BISON fuel performance code, and a demonstrative evaluation of the fuel rod failure probability analysis using the concept of a limit surface has been performed. The paper presented an extensive mathematical description of the limit surface concept and of its mathematical representation. There are several issues that need to be further investigated, in particular concerning the error bounds in the representation of the limit surface and the impact of usage of surrogate models on the overall estimation error. In spite of these challenges, the methodology here presented has strong potential to become a valuable tool for improved fuel design and better understanding of operative margins in current reactors. In fact, given the stochastic nature and the uncertainty associated with some aspects of nuclear fuel analysis, it is thought that a probabilistic approach can enhance fuel performance and safety evaluations. Currently, efforts are ongoing to accelerate the limit surface search and perform inline error control. These new developments are intended to lead to a wider deployment of this approach to the point to positively impact the common practice.

## ACKNOWLEDGMENTS

This work is supported by the U.S. Department of Energy, under DOE Idaho Operations Office Contract DE-AC07-05ID14517. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

## REFERENCES

1. C. RABITI, A. ALFONSI, J. COGLIATI, D. MANDELLI, R. KINOSHITA, “RAVEN, a New Software for Dynamic Risk Analysis”, in Proceedings for PSAM 12 Conference, Honolulu (USA), 2014.
2. A. ALFONSI, C. RABITI, D. MANDELLI, J. COGLIATI, R. KINOSHITA, AND A. NAVIGLIO, “RAVEN and dynamic probabilistic risk assessment: Software overview,” in Proceedings of ESREL European Safety and Reliability Conference (2014).
3. J.D. HALES, S.R. NOVASCONE, G. PASTORE, D.M. PEREZ, B.W. SPENCER and R. L. WILLIAMNSON. BISON users manual. Technical Report INL/MIS-13- 30307, Idaho National Laboratory, September 2013.
4. “NEAMS: The Nuclear Energy Advanced Modeling and Simulation Program,” *ANL/NE/13-5* (2013).
5. Light Water Reactor Sustainability Program Integrated Program Plan
6. <http://www.casl.gov/index.shtml>.
7. H.T. Banks, S. Hu “UNCERTAINTY PROPAGATION AND QUANTIFICATION IN A CONTINUOUS TIME DYNAMICAL SYSTEM,” International Journal of Pure and Applied Mathematics, Vol. 80 No. 1 2012, 93-145
8. 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 Proc. of Int. Conf. of Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013), Sun Valley (Idaho), pp. 320–332, 2013.
9. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, David Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, “Scikit-learn: Machine Learning in Python”, Journal of Machine Learning Research 12, pp. 2825-2830, 2011.