INL/EXT-16-40094

Light Water Reactor Sustainability Program

Enhancements to the RAVEN code in FY16

Andrea Alfonsi, Cristian Rabiti, Daniel Maljevoic, Diego Mandelli, Joshua Cogliati

September 2016

DOE Office of Nuclear Energy

INL/EXT-16-40094

Light Water Reactor Sustainability Program

Enhancements to the RAVEN code in FY16

Andrea Alfonsi,
Cristian Rabiti,
Daniel Maljevoic,
Diego Mandelli,
Joshua Cogliati

September 2016

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov/lwrs

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

EXECUTIVE SUMMARY

The RAVEN code has been under development at the Idaho National Laboratory since 2012. Its main goal is to create a multi-purpose platform for the deploying of all the capabilities needed for Probabilistic Risk Assessment, uncertainty quantification, data mining analysis and optimization studies. RAVEN has currently reached a good level of maturity in terms of deployed state-of-art and advanced capabilities.

The main subject of this report is to show the activities that have been recently accomplished:

- Implementation of ensemble modeling for time-series, and
- initial implementation of model validation for surrogate models, and
- advanced visualization capability for topology based data analysis

The development of ensemble modeling for time-series has been performed in order to begin tackling the needs of those RISMC applications that need to communicate 1-Dimensional information (e.g. power histories, etc.) among different models. In this document the implementation details and an application example is reported.

The second subject of this report is about the initial development of methods, within the RAVEN framework, to assess the validity of the predictive capabilities of surrogate models. Indeed, after the construction of a surrogate tight to a certain physical model, it is crucial to assess the goodness of its representation, in order to be confident with its prediction. In this initial activity, a cross-validation technique has been employed. This report will highlight the implementation details and proof its correct implementation by an application example.

The final subject of this report is about the implementation of advanced visualization capability in RAVEN, for interactive data analysis. Indeed, RAVEN offers several post-processing capabilities that can structurally decompose data extracted from experimental results offering both data clustering/partitioning and dimensionality reduction techniques. A disadvantage of the workflow available in RAVEN is that it treats these as black box operations and the user is expected to know specific information about their data including the number of partitions to expect in some cases or the “correct” parameter settings for a particular algorithm. In order to overcome this limitation, it has been added an interactive user interface that can be run in RAVEN to explore different parameter settings on the fly specifically for the topological post-processor and allows the user to explore the data interactively. The design of this user interface is generalizable to other algorithms and can be used as a model to generate more dynamic and user-friendly visualization capabilities within RAVEN.

CONTENTS

FIGURES

TABLES

ACRONYMS

DOE	Department of Energy
INL	Idaho National Laboratory
LWRS	Light Water Reactor Sustainability
MOOSE	Multiphysics Object Oriented Simulation Environment
NPP	Nuclear Power Plant
PRA	Probabilistic Risk Assessment
R&D	Research and Development
RISMC	Risk-Informed Safety Margin Characterization
FOM	Figure of Merit
ROM	Reduced Order Model
SM	Surrogate Model

Enhancements to the RAVEN code for FY16

1. INTRODUCTION

The overall goal of the Risk Analysis in a Virtual ENvironment (RAVEN) software is to understand the probabilistic behavior/response of complex systems. In particular RAVEN is focused on risk analysis. The capability to predict the likelihood that the system under consideration will end in a particular region of the output space, which is the base of the risk analysis, is already an available feature of the RAVEN code (i.e., the "limit surface" approach). Those types of analyses rely on the availability of probability distributions and sampling strategies, according to which the input space of the probabilistic systems are explored, and statistical post processors used to understand the dispersion of the system response.

In FY 15 RAVEN capabilities were extended to provide the tools to the engineers to better understand the reasons/drivers of the system responses by adding advanced sampling strategies (e.g. hybrid dynamic event tree and adaptive hybrid dynamic even tree) [1], advanced static data mining capabilities (e.g. clustering, principal component analysis, manifold learning) [2], and ways to connect multiple reduced order model (able to reproduce scalar figure of merits) in order to create ensemble of models.

While these added capabilities showed a potential in terms of analysis workflows, it was clear that some additional enhancements were needed in order to ease the analysis process for the user.

This is the motivation why multiple parallel activities have been performed this year:

- Implementation of ensemble modeling for time-series:

  As already mentioned, a technique to connect multiple reduced order models representing different connecting physics has been performed in FY15. This new capability showed a benefit in terms of aggregated analysis for UQ and PRA. The implementation was limited to transferring information among the different models that could be collapsed in scalar quantities (e.g. maximum temperature, maximum pressure, thermal conductivities, etc.). This approach is not enough to be able to combine models whose boundary conditions or input parameters are 1-Dimensional (e.g., power histories, etc.). For this reasons, the ensemble modeling has been upgraded in order to be able to transfer complex Figure of Merits of vectorial types of information.

- Initial implementation of model validation for surrogate models:

  After the construction of a surrogate tied to a certain physical model, it is crucial to assess the goodness of its representation, in order to be confident with its prediction. In this initial activity, a cross-validation technique has been employed. This new RAVEN attribute allows analysts to create complex models and have a method to determine the overall fidelity of these models.

- Advanced visualization capability for topology based data analysis:

  RAVEN offers several post-processing capabilities that can decompose data extracted from experimental results offering both data clustering/partitioning and dimensionality reduction techniques. A disadvantage of the workflow available in RAVEN is that it treats these information entities as black box operations and the user is expected to know specific information about their data including the number of partitions to expect in some cases or the "correct" parameter settings for a particular algorithm. In order to overcome this limitation, a new feature has been added in RAVEN that provides an interactive user interface to explore different parameter settings on the fly specifically for the topological post-processor and allows the user to explore the data interactively. The design of this user interface is generalizable to other algorithms and can be used as a model to generate more dynamic and user-friendly visualization capabilities within RAVEN.

The report is organized as follows:

1) Section 2 reports the development of the ensemble modeling for 1-D FOMs, demonstrated via an application example.

2) Section 3 shows the newly implemented cross-validation techniques used to validate the prediction and goodness of the surrogate models. An application example is shown.

3) Section 4 is about the development and improvements of interactive visualization techniques within the RAVEN post-processing module.

4) In section 5 few other improvements are highlighted.

5) Section 6 concludes the report highlighting the future development directions.

2. ENSEMBLE MODELING FOR 1-D FOM

2.1 Introduction

As reported in previous reports, RAVEN is able to construct multi-targets Reduced Order Models [4], which are aimed to represent the response of a system (in a fixed configuration) for multiple Figures of Merits (FOMs) and time-dependent ROMs. These capabilities represented the initial steps for a larger implementation about the interaction of multiple models. In fact, in several cases, multiple models need to interface with each other since the initial conditions of one are dependent on the outcomes of another.

To better understand the problem that here is solved, it is useful to consider two simple examples:

1) The following problem is considered: a weather forecast simulation code "a" is used to compute the external (i.e., ambient) temperature in a certain location. A second model "b" is inquired to compute the average temperature in a room having as boundary condition, among several others, the external ambient temperature. In this case, the response of the model "b" depends on the outcome of the model "a" through the temperature boundary condition;
2) Two different simulation codes are considered: a) a code that is meant to compute the thermal conductivity of the ceramic Uranium Dioxide (UO_2) as function of the Temperature, and b) a Thermal-hydraulic (TH) code that is used to compute the Temperature field of a reactor, whose heat conduction depends on the thermal conductivity value. As easily inferable, the two models are mutually dependent, as determined in a non-linear system of equations;

The two examples are only aimed to illustrate the reason why the creation of a framework to make interact different models was and is a key development for the advancement of RAVEN as a comprehensive calculation flow driver. Before reporting how the ensemble-models have been implemented and their extension in order to process 1-Dimensional FOMs, it is necessary to briefly describe the representative Model "entities" that are available in RAVEN.

2.2 Models in RAVEN

The Model entity, in the RAVEN environment, represents a "connection pipeline" between the input and the output space. The RAVEN software itself does not own any physical model (i.e., it does not possess the equations needed to simulate a system), but implements APIs by which any generic model can be integrated and interrogated. In the RAVEN framework four different model categories (entities) are defined:

- Codes;
- Externals;
- ROMs;
- Post-Processors.

The Code model represents the interface object that establishes the communication pipe between RAVEN and any driven code. Currently, RAVEN has APIs for several different codes:

- RELAP5-3D, a safety analysis code (thermal-hydraulic) developed at INL;
- RELAP-7, a safety code under development at the INL;
- any MOOSE-based application;
- SAS4A/SASSYS-1, safety analysis code for fast reactors developed at Argonne;
- Modelica, object-oriented, declarative, multi-domain modeling language for component-oriented modeling of complex systems;

- MELCOR, engineering-level computer code that models the progression of severe accidents in light-water reactor nuclear power plants (coupling under development by the University of Rome “La Sapienza”);
- MCNP, general-purpose Monte Carlo N-Particle code that can be used for neutron, photon, electron, or coupled neutron/photon/electron transport (the interface for this is still under development);
- MAAP5, computer code that models the progression of severe accidents in light-water reactor nuclear power plants (coupling performed by the Ohio State University).

The data exchange between RAVEN and the driven code can be performed either by direct software interface or by files such as input files. If the system code is parallelized, the data exchanging by files is generally the way to follow since it can be much more optimized in large clusters.

The External model allows the user to create, in a Python file (imported, at run-time, in the RAVEN framework), its own model (e.g. set of equations representing a physical model, connection to another code, control logic, etc.). This model will be interpreted/used by the framework and, at run-time, will become part of RAVEN itself.

The ROM (Reduced Order Model) represents an API to several different algorithms. A ROM is a mathematical representation of a system, used to predict a selected output space of a physical system. The creation and sub-sequential usage of a ROM involves a procedure named “training”. The “training” is a process that uses sampling of the physical model to improve the prediction capability (capability to predict the status of the system given a realization of the input space) of the ROM. More specifically, in RAVEN the ROM is trained to emulate a high fidelity numerical representation (system codes) of the physical system.

The Post-Processor model is aimed to manipulate the data generated, for example, employing a sampling strategy. In RAVEN several different post-processors are available: 1) Statistics Post-Processor, aimed to compute all the statistical figure of merits (e.g. expected values, variance, skewness, covariance matrix, sensitivity coefficients, etc.); 2) Limit Surface, which computes the Limit Surface, inquiring a goal function (i.e. a function that determines if a certain coordinate in the input space led to a failure or success), and so many others.

2.3 Current EnsembleModel implementation

As already mentioned, in several cases multiple models need to interface with each other since the initial conditions of some are dependent on the outcomes of others. In order to face this problematic in the RAVEN framework, a new model category (e.g. class), named *EnsambleModel*, was implemented in FY15. This class is able to assemble multiple models of other categories (i.e. Code, External Model, ROM), identifying the input/output connections, and, consequentially the order of execution and which sub-models can be executed in parallel.

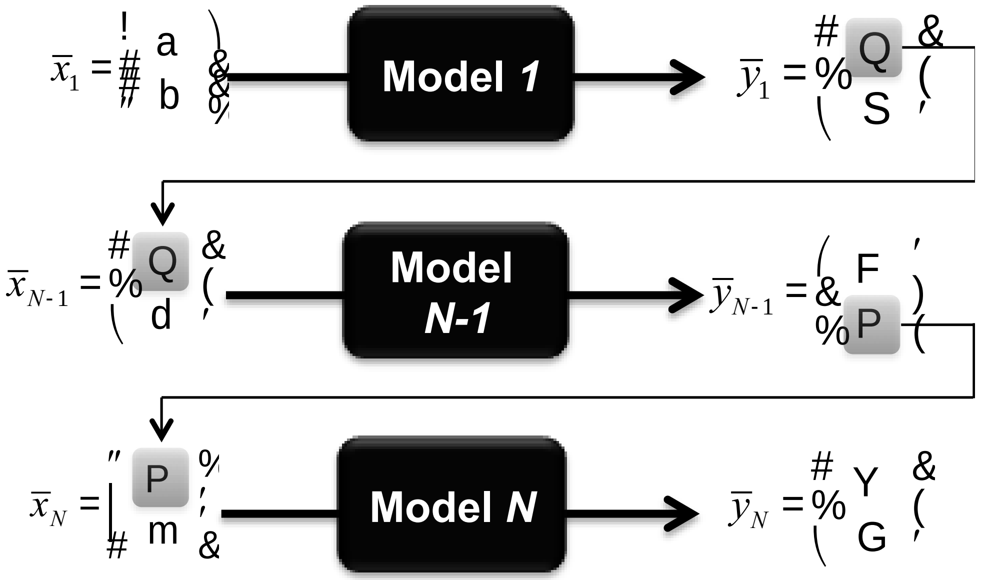

Figure 1 - Example of an Ensemble-Model constituted by 3 sequential sub-models.

Figure 1 reports an example of a Ensemble-Model that is constituted by 3 sub-models (ROMs, Codes, or External Models). As it can be noticed:

- The *Model 2* is connected with the *Model 1* through the variable Θ (Model 1 output and Model 2 input);
- The *Model 3* is connected with the *Model 2* through the variable Π (Model 2 output and Model 3 input);

In this case, the Ensemble-Model is going to drive the execution of all the sub-models in a serial sequence, since each model (except the *Model 1*) is dependent on one of the outcomes of previously executed.

In several cases, the input of a model depends on the output of another model whose input is the output of the initial model. In this situation, the system of equation is non-linear and an iterative solution procedure needs to be employed. The Ensemble-Model entity in RAVEN is able to detect the non-linearity of the sub-models' assembling and activate the non-linear solver: Picard's iterative scheme. **Figure 2** shows an example of when the Ensemble-Model entity activates the Picard's iteration scheme, which ends when the residue norm (between an iteration and the other) falls below a certain input-defined tolerance.

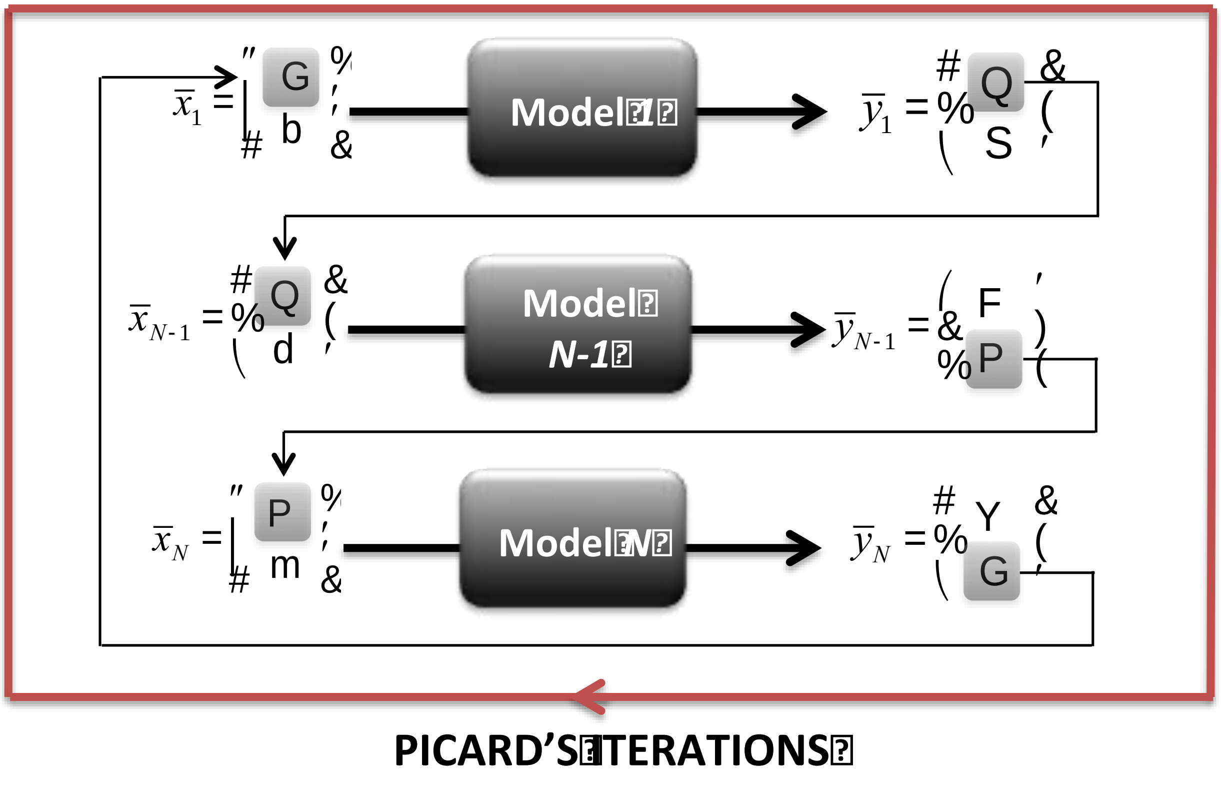

Figure 2 – Ensemble-Model resolving in a non-linear system of equations – Picard's iterations.

In RAVEN all the models' outputs (e.g. whatever code output, etc.) are collected in an internal containers (named *DataObjects*) that are aimed to store time-series and input/output data relations in a standardized fashion; in this way, the communication of the output information among different entities (i.e. Models) can be completely agnostic with respect to the particular type of output generated by a model. The Ensemble-Model entity fully leverages this peculiarity in order to transfer the data from a Model to the other(s).

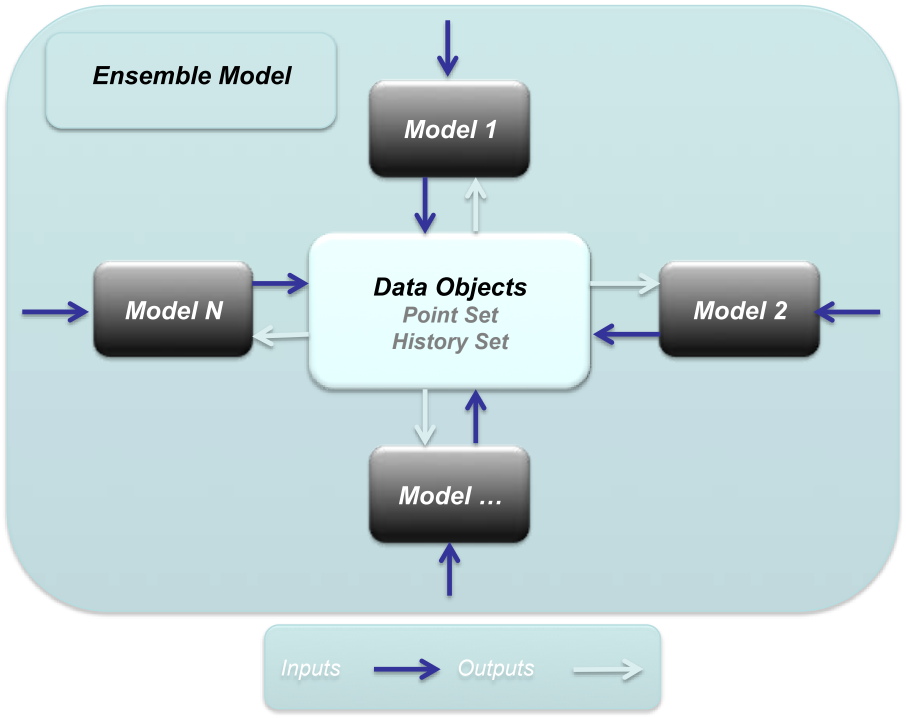

Figure 3 - Meta-model data exchange among sub-models.

Based on the Input/Output relations of each sub-models, the Ensemble-Model entity constructs the order of their execution and, consequentially, the links among the different entities.

2.4 EnsembleModel for 1-Dimensional Figure of Merits

The infrastructure of the *EnsembleModel* that has been developed in FY15 was able to transfer information among different models just in case of scalar quantities (e.g. peak clad temperature, constant thermal conductivities, etc.). This limitation was connected to the fact that in RAVEN the concept of "input realization" was limited to scalar uncertainties (i.e. the RAVEN code was able to perturb the input space in terms of scalar quantities). Since the increasing interest in using RAVEN also oriented to interconnection among heterogeneous Models' entities, the concept of treatable "input realizations" has been revised, including the possibility to process 1-D realizations.

Figure 3 schematically shows the communication piping established by the *EnsembleModel* entity. It can be noticed how the sub-models share information (inputs/outputs data) using the *DataObjects* entity as communication network.

2.5 Application example

In order to test the newly developed capability to process 1-D FOMs in an *EnsembleModel* configuration, two models have been considered:

- A pump controller model (***Model B***) for a hypothetical simplified PWR model (see Figure 4) has been used. It consists of the following components:
    - Reactor core (RX)
    - Motor operated pump
    - Pump digital controller

- Heat exchanger (HX)

This system is responsible to remove the decay heat generated from the core (RX) in order to avoid damage of the core itself. While we assumed that both the HS and the pump are perfectly reliable components (i.e., no failure can be introduced), using [Ref] as a references, the digital pump controller reliability model has been performed using a continuous time Markov Chain formulation.

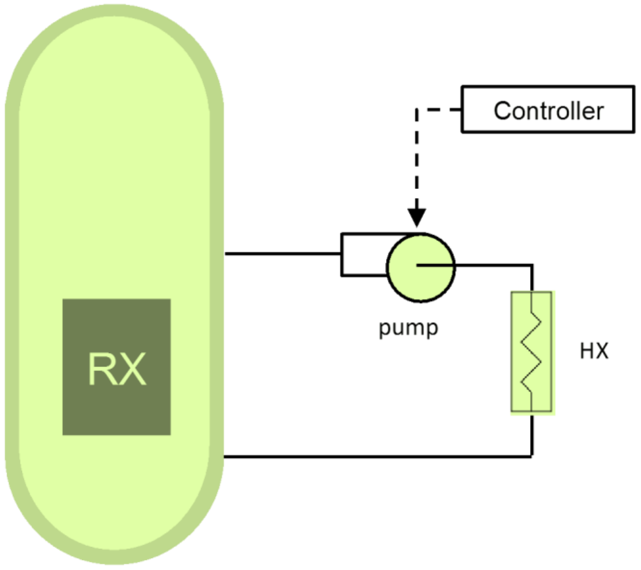

Figure 4 – Pump controller model scheme

In more detail, the controller has been modeled using 4 states (Figure 5):

- Operating: controller operating as designed
- Failed closed: controller failed by sending close signal to pump (i.e., pump not running)
- Failed stuck: controller failed by sending oldest valid signal to pump
- Failed random: controller failed by sending close signal to pump

Since the scope of this exercise is to show the new capability in RAVEN, an additional

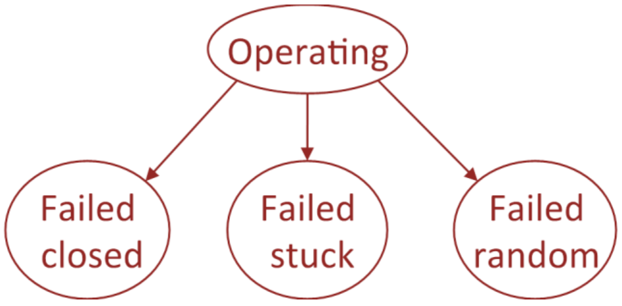

Figure 5 - Continuos time Markov model for the pump controller

In order to perform such analysis the model has been coded as a RAVEN external model (see Sect. 2.2) which determine the temporal profile of core temperature give the two stochastic parameters:

- Pump controller failure time ($\boldsymbol{CNTR_{f,time}}$)

- Pump controller failure mode ($\boldsymbol{CNTR_{f,mode}}$)

The dynamic of the hypothetical system has been modeled using basic mass and energy conservation laws so no effective engineering conclusions can be gathered by this example.

- A power history generator model (***Model A***) has been used. It employs of the following simple equation:

$$Power(t) = 1{,}500 * scaling_P * \exp(-t)$$

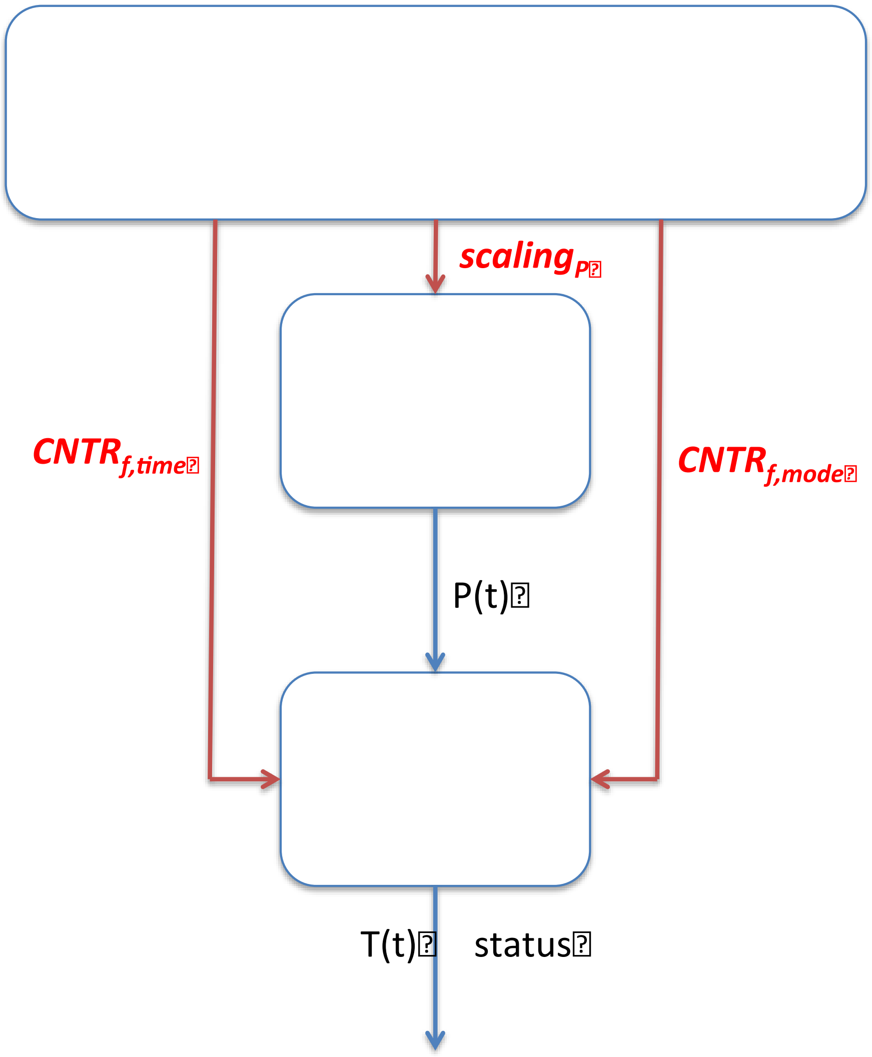

Figure 6 - Ensemble Model scheme for PWR controller example

The power multiplier (***scaling_P***) is an additional stochastic parameter in this example analysis (Uniform between 0.5 and 1.2).

The *EnsembleModel* data flow is shown in Figure 6. The Model A generates a Power history (time-dependent) that is passed into the Model B that employs the balance analysis. Even if the presented example is quite simplified, it shows the potential of this added capability.

By using RAVEN we sampled the three stochastic parameters using a Monte-Carlo algorithm and generated 1500 simulations as shown in Figures Figure 7 andFigure 8. Note in Figure 8 (i.e., Model B) that in some cases elevated temperatures are recorded due to the potential failures of the pump in the system.

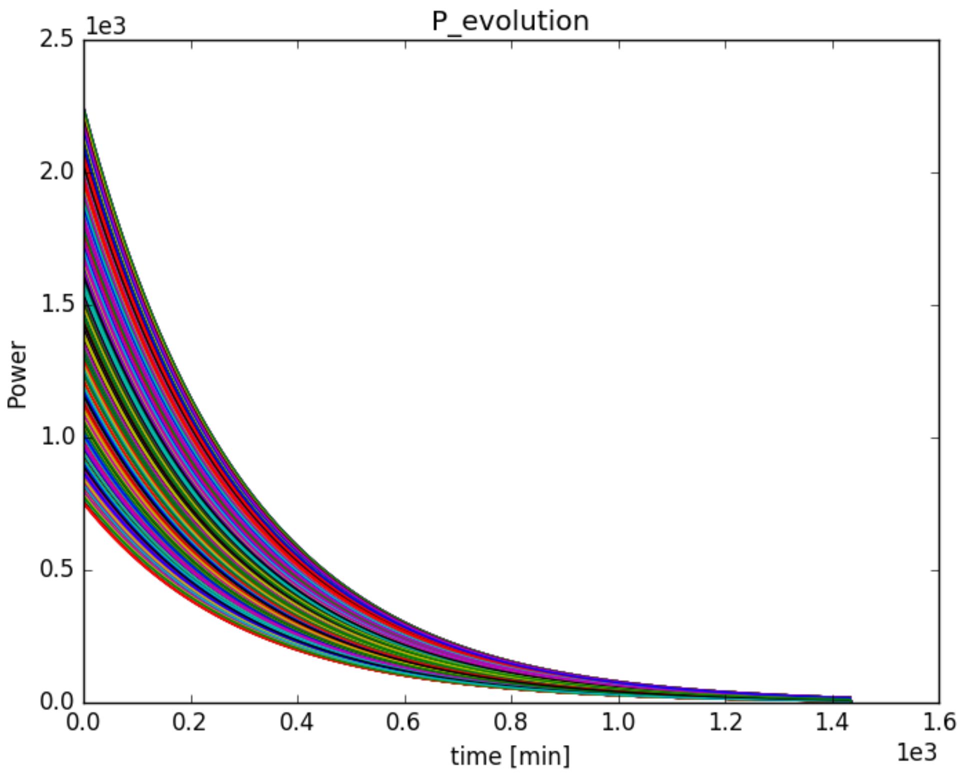

Figure 7 - Plot of the 1500 histories (Power - Model B) generated by RAVEN

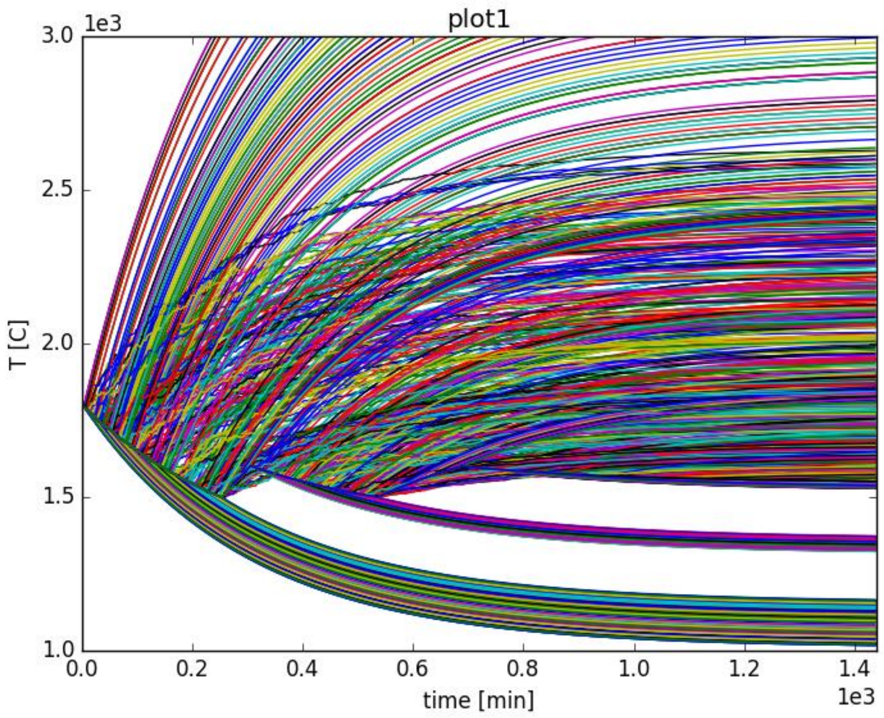

Figure 8 - Plot of the 1500 histories (Temperature – Model A) generated by RAVEN

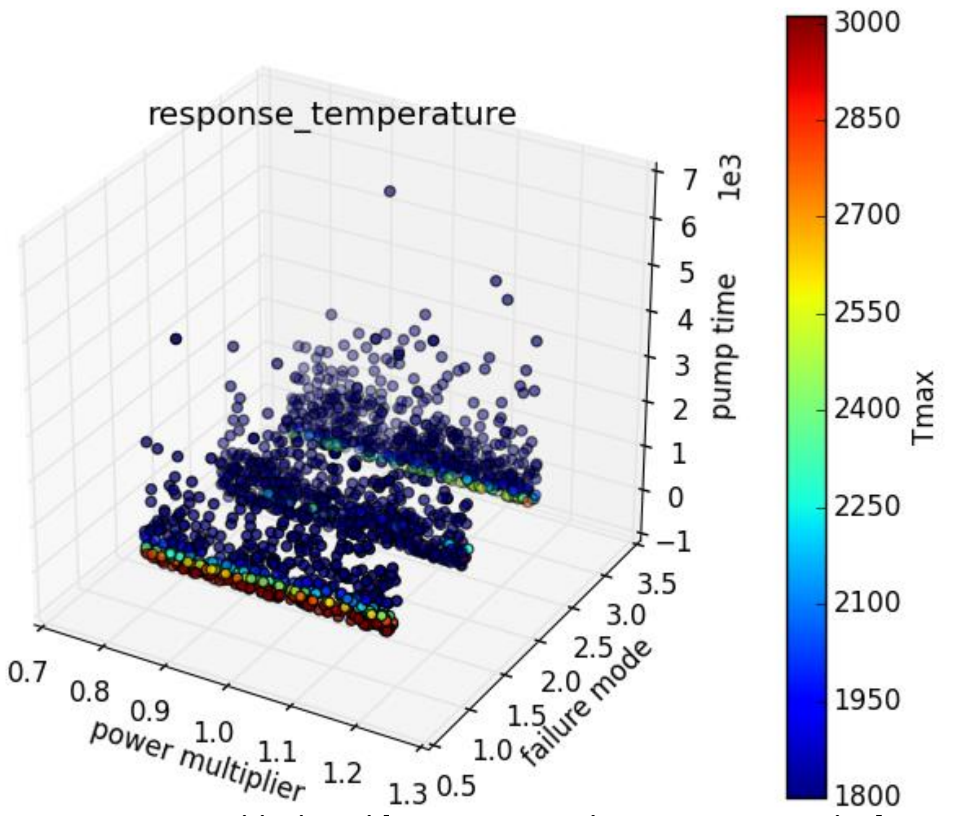

Figure 9 – Input space partitioning with respect to maximum temperature in the system

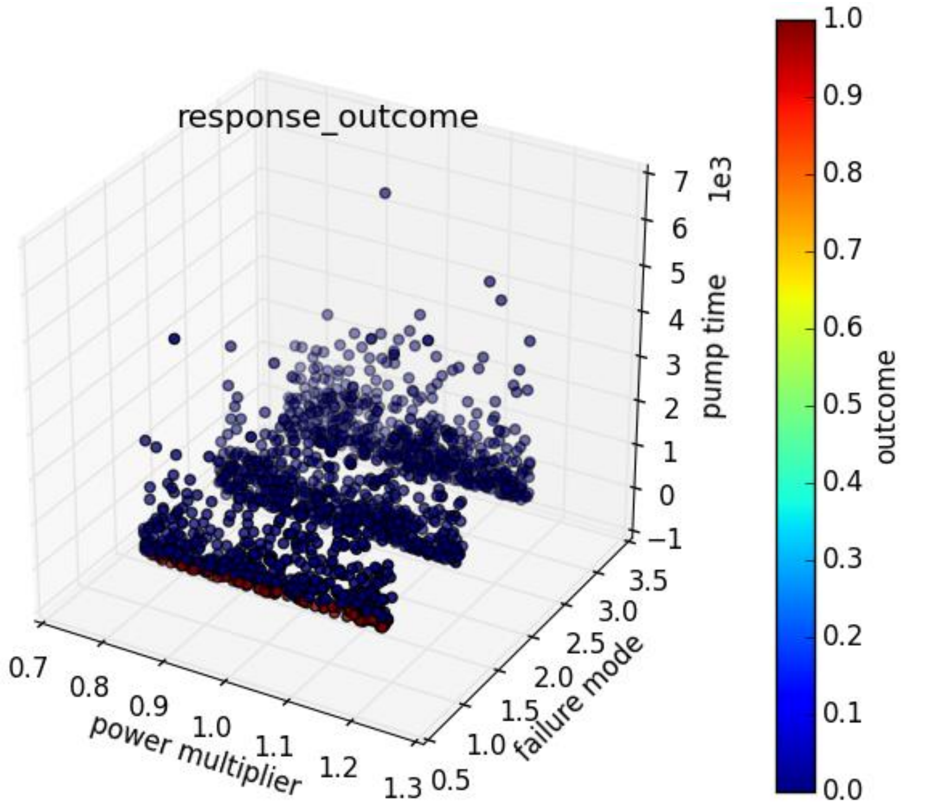

Figure 10 - Input space partitioning with respect to the outcome of scenario (failure/success)

In Figures Figure 9Figure 10 it can be seen as most of the failures happened in failure mode 1 (i.e. controller failed by sending close signal to pump (i.e., pump not running).

This simple example testifies how RAVEN can share 1-Dimensional Figure of Merits, abstracting the concept of “input realization” from scalars (e.g. uncertainty on thermal conductivity) to vectors (e.g. power histories).

3. SURROGATE MODEL VALIDATION TECNIQUE IN RAVEN

As already mentioned in previous sections and detailed in previous reports [5,3], an important pathway in RAVEN analysis workflow is represented by the heavy usage of Surrogate Models (a.k.a. Reduced Order Model - ROM). Even if the concept of Surrogate Models (SMs) has been extensively explored in past reports, it is important to recall its meaning. A Surrogate Model is a mathematical model of fast solution trained to predict a response of interest of a physical system. The training process is performed by "sampling" the response of a physical model with respect to variations of parameters that are subject to probabilistic or parametric behavior. These samples are fed into the ROM that tunes itself to replicate those results.

In the RAVEN case the SM is constructed to emulate a numerical representation of a physical system. In fact, as already mentioned, the physical system model would be too expensive to be used to explore the whole input space.

In RAVEN, several SMs are available: Support Vector Machine, Neighbors based, multi-class models, Quadratic Discriminants, etc. All these SMs have been imported via an Application Programming Interface (API) within the scikit-learn library [6] and few have been also specifically developed (e.g. Polynomial Chaos, N-Dimensional Spline regressors, etc.).

At the FY16 begin the RAVEN user did not have the possibility to assess the validity of a constructed SM, since no metrics were available to define "how good" the prediction of a SM was. This aspect represented a large limitation on the usage of SMs for uncertainty quantification and probabilistic risk assessment, since the user was not able to fully "trust" the response of a SM, overall when the response of the system (in terms of trends) was not known. The development under RISMC for the RAVEN code has been focused on addressing this limitation, defining and implementing "validation" metrics for SM construction and usage.

In order to tackle this problem, the RAVEN team decided to implement a black-box approach that could be used for all the SMs currently available: cross-validation. As detailed in next section, the cross-validation is a statistical method of comparing and evaluating the validity of learning algorithms (i.e. classification and regression algorithms) in modeling the response of a physical system. Through this methodology is possible to assess the validity of the regression/classification algorithms in emulating the response of the system. In the following section, the concept of cross-validation is detailed and their implementation in RAVEN is reported. As a proof of concept, an application example is finally reported.

3.1 Cross Validation for assessing surrogate modeling prediction capabilities

As previously mentioned, the approach that has been chosen for providing initial SM validation capabilities in RAVEN is based on cross-validation. Cross-validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two portions: one used to "train" a surrogate model (building the surrogate model) and the other used to validate the model, based on specific scoring metrics. In typical cross-validation, the training and validation sets must crossover in successive rounds such that each data point has a chance of being validated against the various sets. The basic form of cross-validation is k-fold cross-validation. Other forms of cross-validation are special cases of k-fold cross-validation or involve repeated rounds of k-fold cross-validation.

Cross-validation is used to evaluate or compare learning algorithms as follows: in each iteration, one or more learning algorithms use k folds of data to learn one or more models, and subsequently the learned models are asked to make predictions about the data in the validation fold. The performance of each learning algorithm on each fold can be tracked using some pre-determined scoring metric (see Sect. 3.1.2). Upon completion, k samples of the performance metric will be available for each algorithm. Different methodologies such as averaging can be used to obtain an aggregate measure from these samples, or these samples can be used in a statistical hypothesis test to show that one algorithm is superior to another.

3.1.1 Background

As highlighted in [7], there are two possible goals in cross-validation:

- To estimate performance of the learned model from available data using one algorithm. In other words, to gauge the generalizability of an algorithm.
- To compare the performance of two or more different algorithms and find out the best algorithm for the available data, or alternatively to compare the performance of two or more variants of a parameterized model.

The above two goals are highly related, since the second goal is automatically achieved if one knows the accurate estimates of performance. Given a sample of N data instances and a learning algorithm A, the average cross-validated accuracy of A on these N instances may be taken as an estimate for the accuracy of A on unseen data when A is trained on all N instances. Alternatively if the end goal is to compare two learning algorithms, the performance samples obtained through cross-validation can be used to perform two-sample statistical hypothesis tests, comparing a pair of learning algorithms.

Concerning these two goals, various procedures are proposed:

- ***Resubstitution Validation***:

  In resubstitution validation, the model is learned from all the available data and then tested on the same set of data. This validation process uses all the available data but suffers seriously from over-fitting. That is, the algorithm might perform well on the available data yet poorly on future unseen test data.

- ***Hold-Out Validation***:

  To avoid over-fitting, an independent test set is preferred. A natural approach is to split the available data into two non-overlapped parts: one for training and the other for testing. The test data is held out and not looked at during training. Holdout validation avoids the overlap between training data and test data, yielding a more accurate estimate for the generalization performance of the algorithm. The downside is that this procedure does not use all the available data and the results are highly dependent on the choice for the training/test split. To deal with these challenges and utilize the available data to the max, k-fold cross-validation is used.

- ***K-Fold Cross-Validation***:

  In k-fold cross-validation the data is first partitioned into k-equally (or nearly equally) sized segments or olds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k 1 folds are used for learning. Data is commonly stratified prior to being split into k folds. Stratification is the process of rearranging the data as to ensure each fold is a good representative of the whole. For example in a binary classification problem where each class comprises 50% of the data, it is best to arrange the data such that in every fold, each class comprises around half the instances.

- ***Leave-One-Out Cross-Validation***:

  Leave-one-out cross-validation (LOOCV) is a special case of k-fold cross-validation where k equals the number of instances in the data. In other words in each iteration nearly all the data except for a single observation are used for training and the model is tested on that single observation. An accuracy estimate obtained using LOOCV is known to be almost unbiased but it has high variance, leading to unreliable estimates [3]. It is still widely used when the available data are very rare, especially in bioinformatics where only dozens of data samples are available.

- ***Repeated K-Fold Cross-Validation***:

To obtain reliable performance estimation or comparison, large number of estimates is always preferred. In k-fold cross-validation, only k estimates are obtained. A commonly used method to increase the number of estimates is to run k-fold cross-validation multiple times. The data is reshuffled and re-stratified before each round.

3.1.2 Scoring metrics

As previously mentioned, the cross-validation is performed dividing the training set in a "test-training" portion and in a validation one. Once the SM is trained on the "test-training" set, it is evaluated on the validation one in order to assess the capability to predict the true values of the physical model it is aimed to represent. This "splitting" approach determines the need to define (i.e. use) scoring metrics that can gives an indication of the distance between the predicted outcomes and the "true" ones. Several distance metrics can be used. In RAVEN the following ones have been considered:

- ***Classification***:
  - *Accuracy score*:

    If $\tilde{y_i}$ is the is the predicted value of the $i-th$ sample and y_i is the corresponding true value, then the fraction of correct predictions over $n_{samples}$ is defined as:

    $$accuracy(y,\tilde{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\tilde{y_i} = y_i)$$

    where $1(x)$ is the indicator function[1].
  - *Precision score*:

    The precision score is the ratio $tp/(tp+fp)$ where tp is the number of true positives and fp is the number of false positives. The precision is the ability of the classifier not to label as positive a sample that is negative.
  - *Recall score*:

    The recall score is the ratio $tp/(tp+fn)$ where tp is the number of true positives and fn the number of false negatives. The recall is the ability of the classifier to find all the positive samples.
  - *Average precision score*:

    It is a score that is based on the computation of the average precision from predictions. The score corresponds to the area under the precision-recall curve.
  - *Balanced F-score*:

    The balanced F-score can be represented as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

    $$F1 = 2 * \frac{(precision * recall)}{(precision + recall)}$$
  - *Cross entropy loss score*:

[1] Function defined on a set X that indicates membership of an element in a subset A of X, having the value 1 for all elements of A and the value 0 for all elements of X not in A

This scored is based on the loss function used in (multinomial) logistic regression and extensions of it such as neural networks, defined as the negative log-likelihood of the true labels given a probabilistic classifier's predictions. For a single sample with true label $yt\ in$ $\{0,1\}$ and estimated probability yp that $yt = 1$, the log loss is:

$$-\log P(yt|yp) = -(yt\log(yp) + (1 - yt)\log(1 - yp))$$

- ***Regression***:
  - *Mean absolute error*:

    If $\tilde{y_i}$ is the is the predicted value of the $i - th$ sample and y_i is the corresponding true value, then the mean absolute error (MAE) estimated over $n_{samples}$ is defined as:

$$MAE(y, \tilde{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \tilde{y_i}|$$

  - *Mean square error*:

    If $\tilde{y_i}$ is the is the predicted value of the $i - th$ sample and y_i is the corresponding true value, then the mean square error (MSE) estimated over $n_{samples}$ is defined as:

$$MSE(y, \tilde{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \tilde{y_i})^2$$

  - *Median absolute error*:

    If $\tilde{y_i}$ is the is the predicted value of the $i - th$ sample and y_i is the corresponding true value, then the median absolute error (MedAE) estimated over $n_{samples}$ is defined as:

$$MedAE(y, \tilde{y}) = median(|y_1 - \tilde{y_1}|, \dots, |y_n - \tilde{y_n}|)$$

  - *R^2 error*:

    The R^2 is the coefficient of determination. It provides a measure of how well future samples are likely to be predicted by the model. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

    If $\tilde{y_i}$ is the is the predicted value of the $i - th$ sample and y_i is the corresponding true value, then the R^2 score estimated over $n_{samples}$ is defined as:

$$R^2(y, \tilde{y}) = 1 - \frac{\sum_{i=0}^{n_{samples}-1} (y_i - \tilde{y_i})^2}{\sum_{i=0}^{n_{samples}-1} (y_i - \bar{y})^2}$$

    where $\bar{y} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} y_i$

3.2 RAVEN implementation

In following sections, the implementation of the previously reported methodology is explained, reporting functional schemes that provide a usage overview within the RAVEN framework.

3.2.1 RAVEN confidence estimation

In order to provide the capability to estimate the accuracy of a SM prediction and/or its global validity, a model validation entity in the RAVEN framework has been developed. This entity is aimed to apply model validation techniques to any SM currently supported in RAVEN framework. When the user activates the model validation entity, each SM is able to report the "confidence" score for each prediction. Figure 11 shows the functional connection between the SM and the model validation entity; every time the SM is evaluated for certain set of feature realizations $\bar{x}$, an accuracy score $a_{y|\bar{x}}$ is determined, associated to the prediction y.

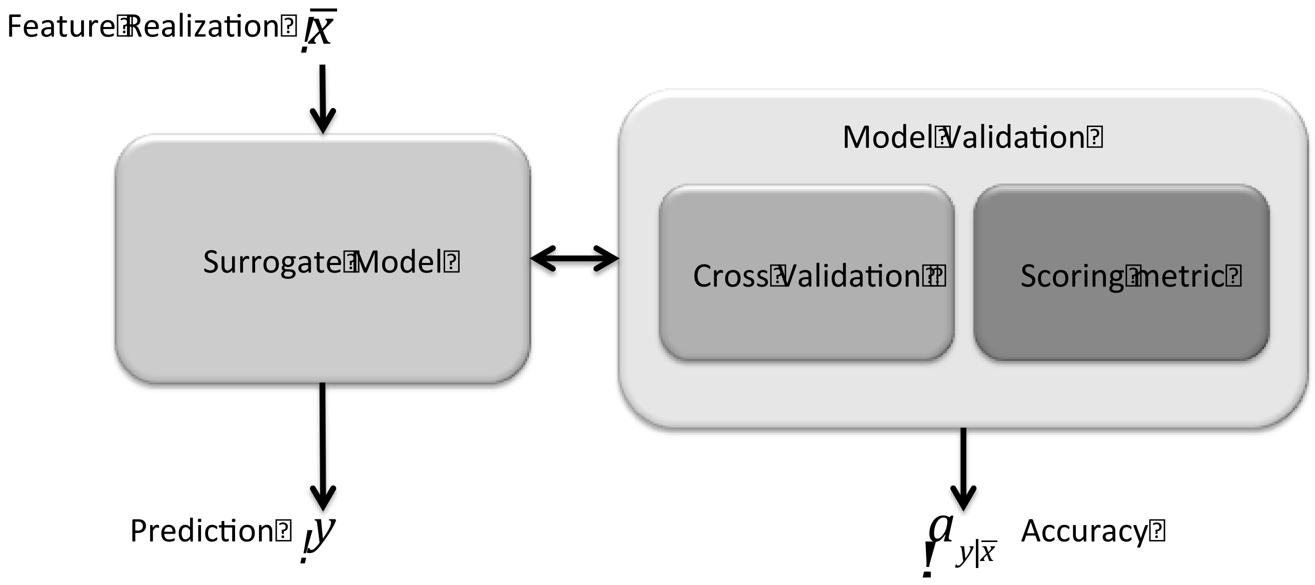

Figure 11 - RAVEN confidence estimation scheme

3.2.2 RAVEN automatic model selection

In the previous section, a schematic representation of RAVEN implementation of the model validation entity has been presented. For each SM evaluation, the model validation entity is able to determine an accuracy score based on the methodologies presented in Section 3.1. Since the implementation of this new capability, a new *Model* entity has been implemented in RAVEN. This new Model is able to combine a SM and any other Model (i.e. High-Fidelity model) leveraging the *EnsembleModel* infrastructure presented in Section *2*, deciding which of the Model needs to be evaluated based on the model validation score calculation. Figure 12 shows the functional scheme of this new entity, for a single evaluation request $\bar{x}$:

1) The SM is evaluated on the feature coordinates $\bar{x}$, producing the outcome y_{SM};
2) The model validation entity is inquired in order to estimate the validity of the prediction $a_{y_{SM}|\bar{x}}$;
3) If the prediction meets a user-define acceptability criterion, the outcome y_{SM} is kept and used as actual "true" value;
4) If the prediction does not meet the user-defined acceptability criterion, the outcome y_{SM} is discarded and the High-Fidelity model is inquired producing the "true" response y. The new "true" evaluation is added in the current SM training set and the SM is then reconstructed.

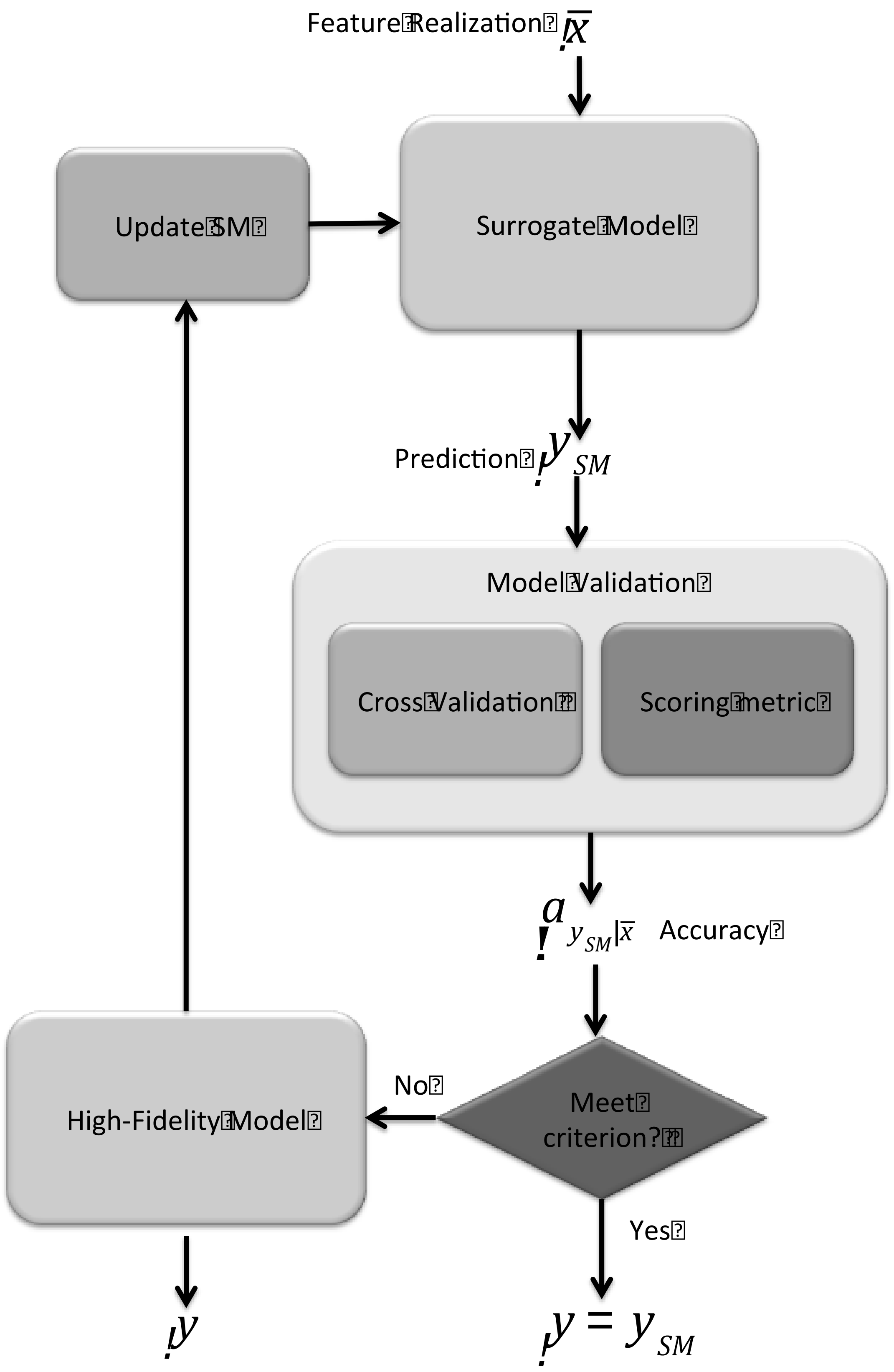

Figure 12 - RAVEN automatic model selection scheme.

3.3 Application example

In the following section, the newly developed capability is shown through an applications example, which shows how a constructed surrogate model can be used in conjunction with model validation techniques in order estimate the model validity for Uncertainty Quantification analysis.

Even if the approach presented in Section 3.2.2 is fully functional (in terms of software infrastructure), its application is not shown here since the only model validation approach so far implemented is the global cross-validation. In order to be fully effective, it requires a local validation method that is going to be the focus for FY17.

3.3.1 Application: RAVEN confidence estimation

As previously mentioned, an initial infrastructure for model validation has been developed. In order to show how the newly implemented capability can be used to construct a usable surrogate model for uncertainty quantification, a simple example has been employed. For the sake of this example, the same Pump controller model and scenario employed in Section 2.5 has been used (see Figure 4 for the model scheme). Similarly to the this scenario, 2000 Monte-Carlo samples has been generated obtaining the reference solution in terms of final temperature reached by the system using the “high-fidelity” model. Figure 13 shows the obtained final temperature distribution, which represents the reference distribution in this exercise.

The SM that has been chosen for showing how to construct a valid and, consequentially converged, regression model is based on a nearest neighbors’ regression algorithm named K-Nearest neighbors. In this type of algorithm the target (i.e. the Figure of Merit for which it has been trained for) is predicted by local interpolation of the targets associated of the nearest neighbors in the training set. Table 1 shows the setting used for this algorithm.

Table 1 - K-Nearest SM settings.

Parameter	Short description	Value used
# Neighbors	Number of neighbors used in the local interpolation	5
Weights	Weighting strategy	Uniform
Metric	The distance metric to use for the tree.	Minkowski
p	Power parameter for the Minkowski metric	2 – Euclidean distance

The assessment of the SM validity has been performed employing a K-Fold (3 folds) cross-validation strategy using a R^2 scoring metric. The SM has been trained with a variable number of MC samples, monitoring the evolution of its prediction accuracy both in terms of final temperature distribution, R^2 and first two statistical moments.

Figure 14 and Figure 15 show the evolution of the predicted final temperature distribution (2000 MC samples) for the 6 training sets and the convergence history of the SM as function of the number of MC samples, respectively. It can be noticed, how the R^2 score metric represents a good estimator of the performance of the SM, completely in agreement with the final temperature distribution results.

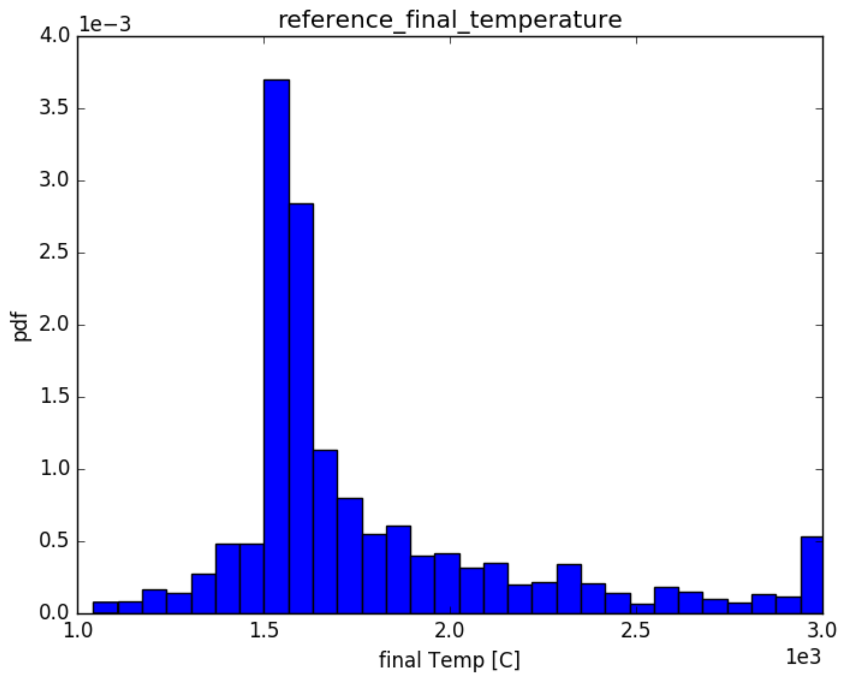

Figure 13 - Reference Final Temperature distribution.

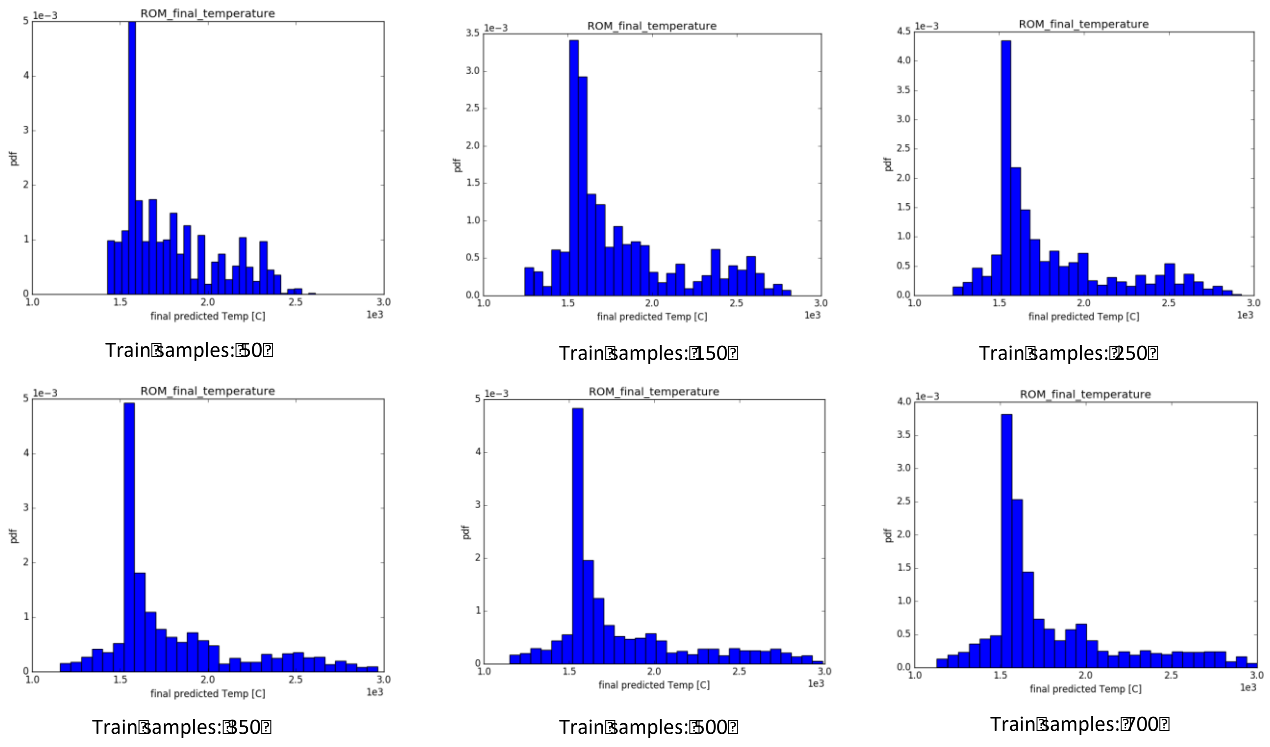

Figure 14 - Evolution of Final Temperature predicted by the SM.

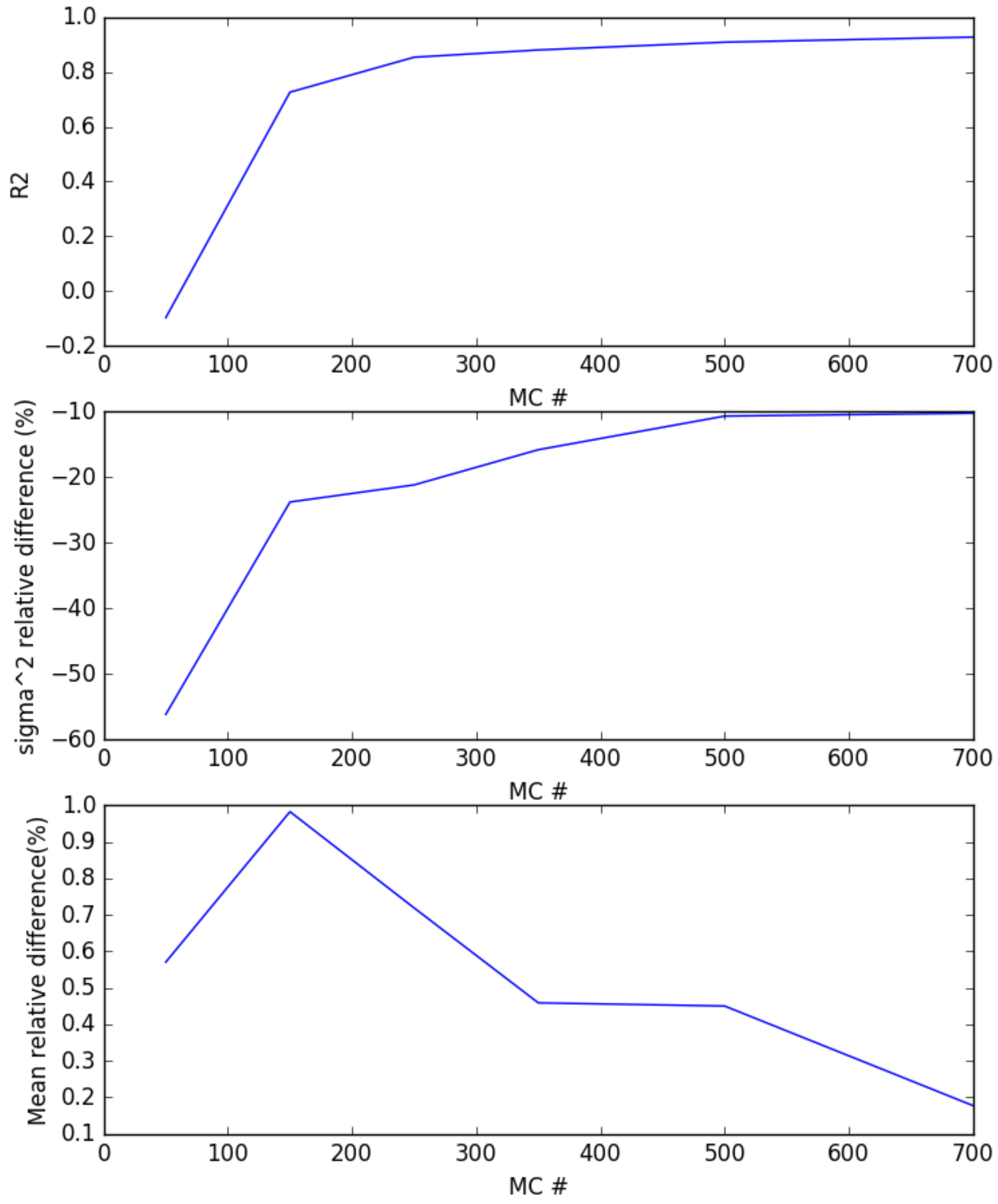

Figure 15 - Converge evolution of the SM as function of the number of MC samples

3.4 Future work

As already mentioned, the implementation of this initial infrastructure for model validation is crucial for creating confidence in the usage of surrogate models. The current infrastructure is general and represents a development framework for adding additional model validation techniques. The current cross-validation technique is quite powerful in assessing the validity of the SMs in their globality but does not provide indication of the local confidence (i.e. error estimation on a particular prediction). Indeed, in FY17 the RAVEN team is going to define and implement local confidence metrics for all the SMs.

In addition, the model validation techniques are going to be used to develop advanced sampling strategies for the construction of converged SMs.

4. VISUAL ANALYSIS CAPABILITIES IN RAVEN

RAVEN offers several post-processing capabilities that can decompose data extracted from experimental results offering both data clustering/partitioning and dimensionality reduction techniques. A disadvantage of the existing workflow available in RAVEN is that it treats these data as black box operations and the user is expected to know specific information about their data including the number of partitions to expect in some cases or the "correct" parameter settings for a particular algorithm. These can be complicated to determine and often a user benefits from visualizing and interacting with the data and the particular method's settings. The visualizations provided in RAVEN are typically static plots, which cannot be adjusted without modifying a text file and rerunning the script. To this end, we have added an interactive user interface that can be run in RAVEN to explore different parameter settings on the fly specifically for the topological post-processor and allows the user to explore the data interactively. The design of this user interface is generalizable to other algorithms and can be used as a model to generate more dynamic and user-friendly visualization capabilities within RAVEN. Section 4.1 describes the implementation details of the new user interface and how it can be generalized to fit more generic RAVEN needs. Section 4.2 provides more augmentation on the Morse-Scale complex. Section 4.3 applies the technique to a BISON nuclear fuel simulation where RAVEN is used to perturb the parameters to generate an input space for analysis. In Section 4.4, we discuss future directions for this development.

4.1 Implementation Details

We have incorporated a new visualization system into the post-processing capabilities of RAVEN in order to allow the user to interactively explore and modify parameters for the topological post-processor. This system utilizes the PySide binding of Qt in order to build a multiple coordinated view [60] system. Such a system allows the user to build views on demand and each view is context aware providing a unified view of the data. For example, if a user selects a specific partition of the data in one view then other views will reflect that selection in some cases providing a more focused view of the data in the given selection. Furthermore, if the user adjusts the threshold parameter, then the underlying partitioning will change and all of the views will be updated using Qt's signaling system. In a similar fashion, the user can choose to build local models on each partition and again the signaling system will handle dispatching updates to each separate view.

Note that this visualization system can be embedded in any GUI environment (e.g. Peacock, MOOSE GUI).

The system we have built is specific to the Morse-Smale complex utilized by the topological post-processor in RAVEN, and so we have evaluated an existing software tool, HDViz [33], and designed a system that provides similar functionality, but is more user-friendly, compliant with the coding standards and environment used by RAVEN, and targets a specific task faced by users of the RAVEN software, namely sensitivity analysis and parameter screening. We now turn our attention to the theory behind this methodology and how we arrived at our specific implementation.

4.1.1 Background

We apply Morse-Smale regression (MSR) [35] in the context of SA. MSR builds upon a domain partitioning of a dataset induced by the Morse-Smale complex (MSC) and employs a linear regression fit within each partition, thereby exploiting its monotonicity. The MSC itself has been successfully utilized in visual exploration of simulation data modeled as high-dimensional scalar functions [33,43,49,50].

Figure 16a shows the MSC of a 2D test function with four maxima and nine minima. The MSC decomposes the domain into 16 partitions such that each partition can be well approximated by a linear model as shown in Figure 16b.

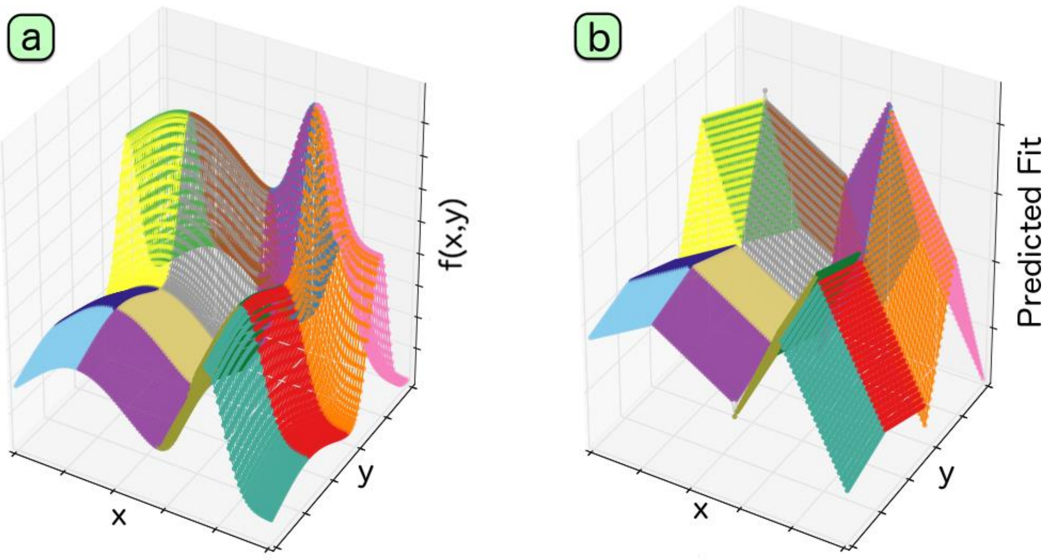

Figure 16 - (a) Morse-Smale complex of a 2D height function that induces a partitioning of the domain. (b) Linear models are fit to each monotonic partition.

The topological characteristics of the MSC are at the core of MSR. Here, we give a high-level description; see [31] for details. The MSC partitions the domain of a scalar function into monotonic regions, where points in each region have gradient flow that begins at the same local minimum and ends at the same local maximum of the function. Furthermore, the MSC can be simplified based on the notion of topological persistence [29,32]. For a fixed scale, the main idea behind the simplification is to merge its corresponding partitions based on a measure of their significance (i.e., persistence); see Figure 17 below for an example applied to the MSC of a 2D height function.

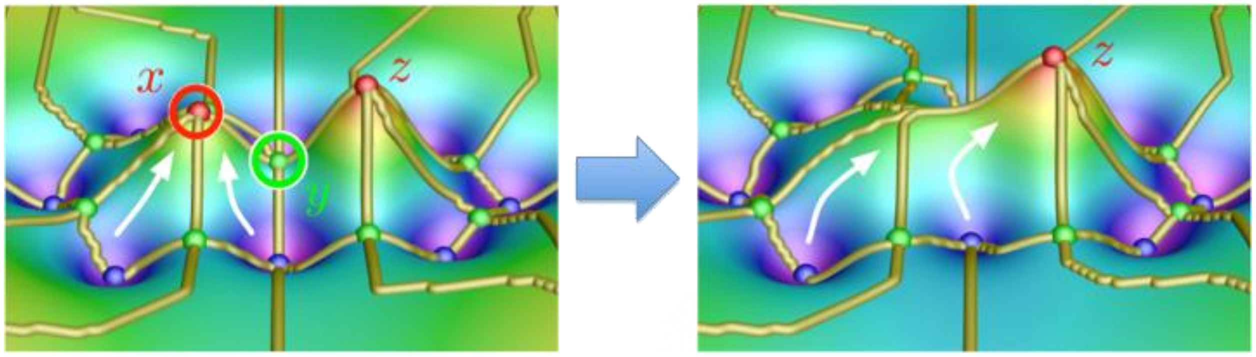

Figure 17 - Example of persistence simplification of a local maximum (x) by pairing and cancelling it with the saddle point (y). The result is shown at right where the gradient is simulated to flow to the more persistent local maximum (z).

For point cloud data, the MSC can be approximated [18,33], enabling MSR to be applied in high dimensions. Points are connected by neighborhood graphs such as the k-nearest neighbor (kNN) graph, and gradients are estimated along the edges of the graph. In our context, we utilize similar approximation schemes [33,35], where points are connected using the relaxed Gabriel graph, which was shown to give superior results in extracting topological features [25] compared to the kNN graph.

For our analysis, we use least square linear regression to fit each partition. To obtain the coefficient estimates, such a least squares fitting minimizes the sum of squared residuals. For a given partition with n data points, let **y** = $[y_1,..., y_n]^T$ be the n-by-1 vector of observed response values, **X** be the n-by-m design matrix of the model (that is, X_{ij} is the *j*-th dimension of the *i*-th data point), and β be the m-by-1 vector of coefficients. We minimize the error estimate:

$$s(\beta) = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{m} X_{ij}\beta_j\right)^2$$

In matrix form, we obtain the coefficient estimates $\hat{\beta}$ in the following way:

$$\hat{\beta} = arg\min_{\beta} s(\beta) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

We use the regression coefficients $\hat{\beta}_i$ $(1 \le i \le m)$ to evaluate the sensitivity of the *i*-th dimension.

For a given partition fitted with a linear regression model, it is important to evaluate how well the data points fit the model by computing the *coefficient of determination*, or the R^2 score. Given a partition with n data points, for the *i*-th data point, y_i is the observed response value and $\hat{y}_i = \sum_{j=1}^{m} X_{ij}\beta_j$ is the fitted response value. The coefficient of determination is computed as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

We extend such a notion by ranking and considering *how many* input dimensions are sufficient to provide an optimal fit. We select a subset of input dimensions for the *n* data points, apply least square linear regression on these points with reduced dimensions, and evaluate the R^2 score of the linear fit. The closer the value of R^2 is to 1, the better the linear regression fits the data with the selected subset of input dimensions.

4.1.2 Problem Characterization

We follow the core stages of a design study [63], namely, discover, design, implement, and deploy, in this application-driven research. During the discover stage, we focus on problem characterization and abstraction. We learn the targeted domain of study through close interactions with stakeholders and problem analysis. In this process, we study the existing workflow available and implement design requirements to better enable knowledge discovery via analysis and visualization. Those requirements are enumerated below.

4.1.2.1 Structure-based Domain Partitioning Amenable to Local SA

As suggested by Gerber et al. [35], we first considered numerous forms of partition-based regression that can be applied for local SA. However, most of these are based on greedy, local operations focusing on optimizing some quality metric. Since the task is to understand trends occurring in the data rather than accurately (potentially over fitting) the data, we selected MSR, as it considers the global structure of the data.

In fact, MSR is particularly suited to this task, as each partition is assumed monotonic and thus well described by a set of simple sensitivity coefficients. In addition, its flexibility and robustness to noise allow it to be used in an exploratory task such as SA where data may be sparse. Existing local SA methods are restricted to be point-wise and any domain partitioning is done manually via a time-consuming process; while our proposed SA, built upon MSR, applies automatically and efficiently to partitions at multiple scales. Using HDViz [33] as a starting point (see Figure 18), it has been an ongoing process for us to define new visual design requirements as well as refining existing ones in order to satisfy the various SA tasks.

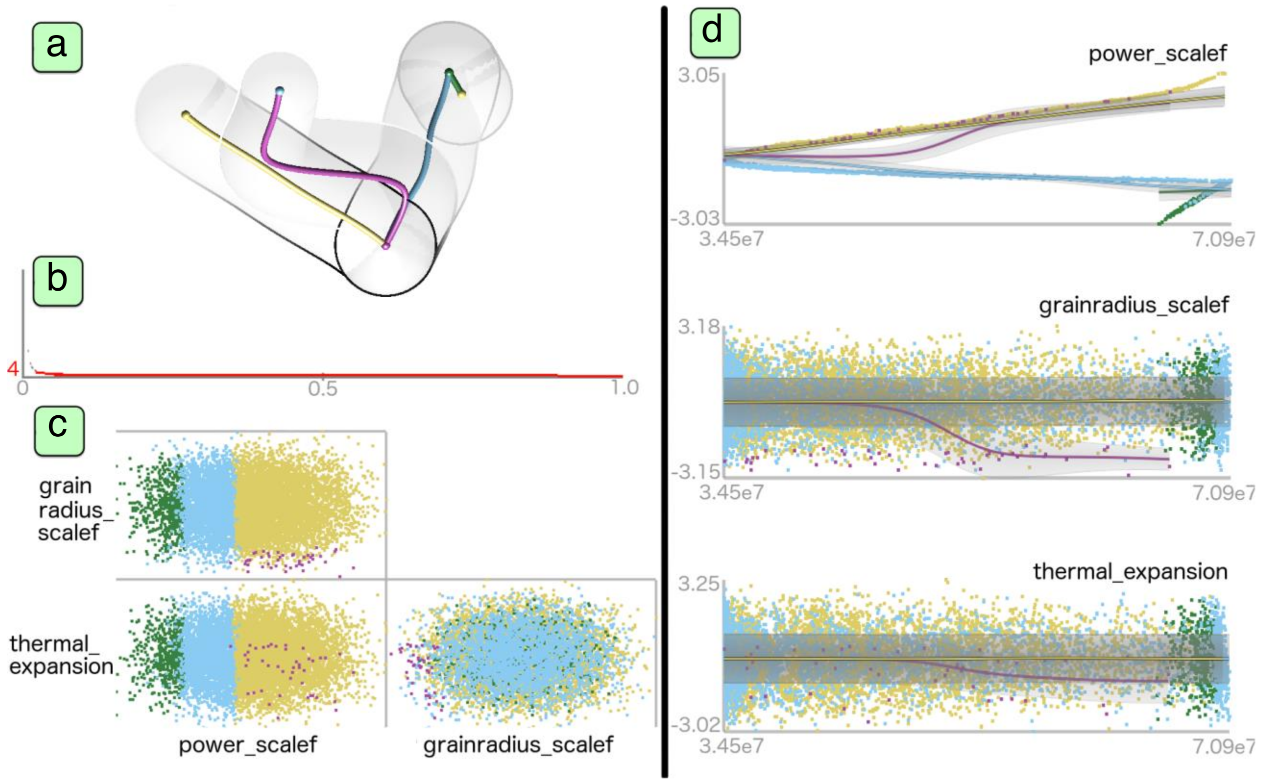

Figure 18 - HDViz applied to the nuclear fuel dataset described in Section 4.3. (a) Topological skeleton: each summary curve in the visual space corresponds to a partition of the data; transparent tubes capture the spread (width) and density (luminance) of the partition. (b) Persistence chart: number of partitions plotted as a function of scale. (c) Scatterplot matrix of partitioned data. (d) Inverse coordinate plots: summary curves and partitioned data are projected onto a 2D plot where the x-axis represents the output dimension, and each y-axis represents an input dimension.

4.1.2.2 Intuitive Visualization of Data Hierarchy

Existing visualization using HDViz has a very steep learning curve and has very limited support for the SA pipeline. For example, much of the information may seem relevant, but it is obfuscated to the untrained eye. Part of our design has been to simplify the information encoded in some respects, and in others to augment with more pertinent information. For example, the tube view in Figure 18a does encode information on the spread and density of the data in high-dimensional space, but often this information is more clearly seen in specific projections of the data where the “spread” can be attributed to specific dimensions. In addition, the sensitivity information encoded in Figure 18d violates the typically notion of placing the independent variable on the horizontal variable on the horizontal axis and the dependent variable on the vertical axis and the view is entirely qualitative with no metrics for strength of relationship or confidence.

Selecting the appropriate scale (i.e., persistence level) from the persistence chart remains an unintuitive process when the scientists are rarely aware of how many noisy features exist in the data. Understanding how users interact with the topological view (Figure 18a) has greatly influenced its design in our iterative process. We understand that the scientists are primarily interested in a high-level picture of their data; therefore, we simplify our visualization to an abstract 2D representation that enables easier selection and manipulation of partitions and supports well-integrated, interactive selection of scales. Finally, the new visual representation should convey information about different scales within the data hierarchy, but also provide context about the partitions within a fixed scale. The topology map described in Section 4.1.3 is designed to fulfill such requirements.

4.1.2.3 Integration of Common Practices with New Designs

We incorporate visual tools that are familiar and ubiquitous to non-visualization experts (e.g., scatterplots and bar charts) in order to facilitate the knowledge discovery process. For example, a user may select an MSC-based partition and observe its associated point cloud clustered in a geometrically coherent space in a corresponding scatterplot. Such integration gives the users some intuition of how the topological partitioning is being performed. In addition, users find that the scatterplots also provide some sense of data density within each partition – such information is vital to the users as it is directly related to the confidence associated with the derived sensitivity information. Knowing such details also allows us to encode density information into other representations not prone to the occlusion problem.

4.1.2.4 Presentation of Comparative and Quantitative Information

Initial visualizations of the sensitivity information focus on a comparative analysis that uses shapes of different sizes to convey differences among individual partitions in the data. Though it is useful for quickly detecting major trends, numeric measurements of each partition give an increased amount of accuracy necessary for decision-making. In addition, selection of the scale parameter (i.e., persistence) can be an art given the potential scarcity in the input data. As such, a user will typically evaluate standard plots for persistence selection (see the persistence barcode and persistence diagram below) in order to determine a clear distinction between feature and noise. In absence of this, one may evaluate the number of data points in each partition by looking at scatterplots of the data. Another important piece of the puzzle is how well the partitions are described by their linear fits. To this end, we provide this as both comparative and quantitative information in the fitness view described in Section 4.1.3.

4.1.2.5 Scalable Analysis and Visualization

Our proposed SA using MSR scales well in high dimensions. Given point cloud data, the analysis performed by Gerber et al. [35] has shown the approximate Morse-Smale complex algorithm provides a good approximation of the true MSC of an underlying function when the smallest feature (signal) of f has a persistence that is an order of magnitude larger than the standard deviation of the noise. The running time of the MSC does not depend on the ambient dimension of the input data, but rather the topological complexity (e.g., number of local extrema) of the underlying function. Therefore, we employ a typical assumption on our targeted datasets such that they have moderate topological complexity, and their topological features can be well approximated. In addition, we require that the new visual design remains intuitive and informative even with increasing dimensionality.

4.1.3 Design

In the design stage, we focus on data abstraction, visual encoding, and interaction mechanisms [63]. We have proposed multiple visual encodings where the feedback from nuclear scientists helps us narrow them down to a few usable solutions. These solutions are integrated into a linked view system with multiple visual components providing interactive functionalities, as illustrated in Figure 19.

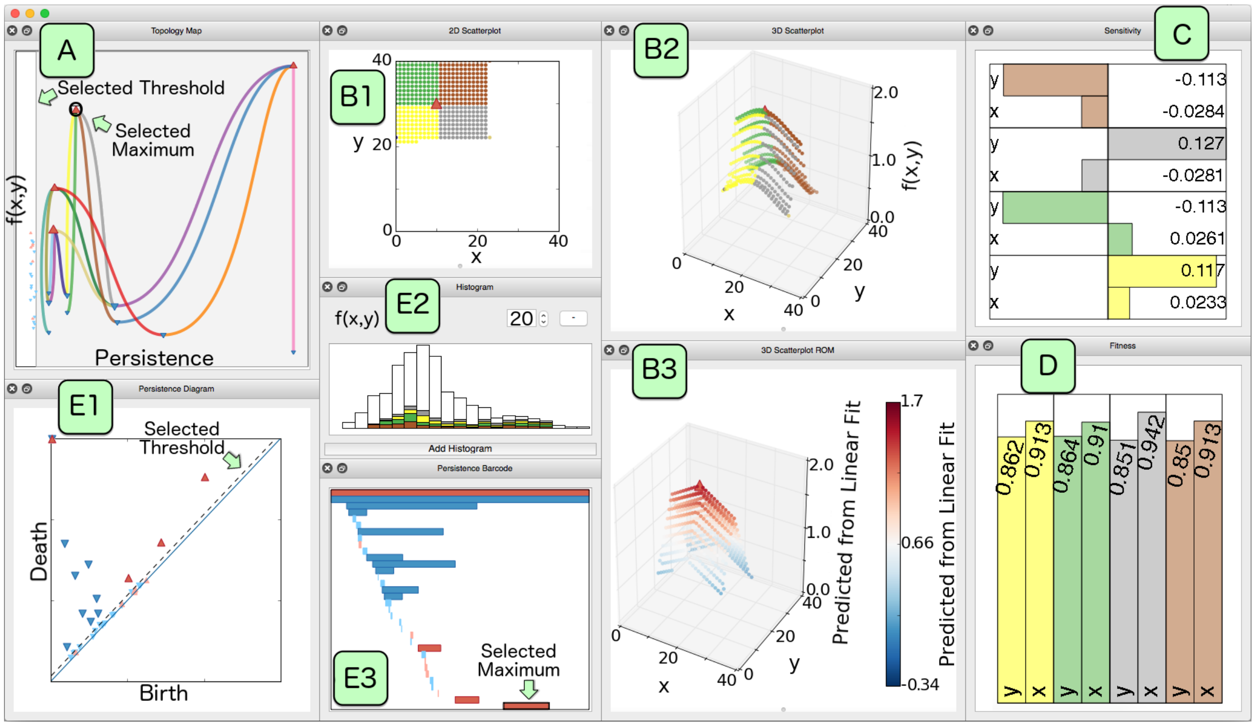

Figure 19 - The linked view visualization system. The system includes (A) topology map, (B1-B3) scatterplots, (C) sensitivity view, (D) fitness view, (E1) persistence diagram, (E2) histogram, and (E3) persistence barcode

A typical workflow begins with the *topology map*, where a user can navigate through the partitioning hierarchy at different scales, and at a chosen scale, explore the structure of the partitions. Within this view, one can understand at a glance the number of local extrema for a given partitioning, their relative importance encoded by persistence, and their connectivity. Traditional topological visualizations such as the *persistence diagram* [23] and the *persistence barcode* [16] can also be used to validate the appropriate choice of scale.
At a fixed scale, the user then selects a subset of partitions for further SA. 2D or 3D scatterplots can be constructed for selected partitions by choosing any two or three input/output dimensions, where the output dimensions include both observed and predicted values. Subsequently, the user can choose to build a *histogram* of any chosen input/output dimension. Such a histogram can also be used for selecting and filtering data for further analysis. Finally, sensitivity information is computed when requested and then visualized on a per dimension basis using the *sensitivity view*, and linear fitness information in terms of R^2 score is given in the *fitness view*. We give a detailed description of the *primary* visual encodings below: topology map, scatterplot projection, sensitivity view, and fitness view followed by a brief introduction to the *secondary* visual encodings: the persistence diagram, the persistence barcode, and histograms.

4.1.3.1 Topology Map

The topology map is a key data abstraction within our design study. It is a 2D representation of the data that highlights its topological structure. We generate and validate such an abstraction through an active and cyclic design process with the scientists to arrive at its current form. The topology map encodes the locations of local extrema defined by their persistence and function values, as well as their connectivity (i.e., curves describing flow from a local minimum to a local maximum). Such visualization is, in spirit, similar to the topological skeleton proposed by Gerber et al. [33] and applied to nuclear probabilistic risk assessment [49,50]. However, based on the new requirements outlined in Section 4.1.2, we have completely redesigned such a visual encoding to provide a topological summary that omits geometric information but preserves the underlying structure essential to understanding the data partitioning, thereby greatly improving its usability and scalability.

As illustrated in Figure 19A, each local extremum is mapped onto a 2D plane, whose horizontal axis represents its *persistence* and vertical axis corresponds to its *function value* (i.e., $f(x, y)$). Therefore, local maxima (minima) move toward the top (bottom) of the display, respectively; more robust features appear to the right, whereas noisy ones tend to the left. Encoding function value to the vertical axis is well aligned with the scientists' understanding of typical 2D function plots where the dependent variables are mapped to the y-axis; in fact, the inverse coordinate plots used in HDViz (Figure 18Figure 16d) are often misinterpreted for violating this common notion. In addition, the local maxima (minima) are upward (downward) pointing red (blue) triangles for fast differentiation and counting.

Selecting the appropriate scale (i.e., persistence level) for data partitioning is essential during the exploratory process as it helps the user to understand the data hierarchy and sensitivity information. The above design also provides a natural separation between features and noise. The user is able to choose a scale by clicking anywhere in the plot, which creates a vertical line passing through the cursor location, separating the local extrema into those that represent robust topological features (grey region on the right) and those that represent topological noise (white region on the left). Extrema in the grey region are visualized together with their connectivity among other extrema. Their sizes correspond to the point densities in their surrounding regions. Extrema in the white region are rendered with a default minimum size and therefore do not draw the user's attention away from more salient features but still provide context.

Selection of the scale parameter is an interactive process that is hard to automate, where the user is free to explore and construct ROMs at arbitrary levels. In *persistence simplification* [23,32], features and noise are assumed to exist on two separable scales, providing a heuristic to choose a scale (vertical line) that is a clear divider between well-separated clusters of extrema. Additional visual cues also help guide such a selection process. For example, in the fitness view, higher R^2 scores typically correspond to better local fits of the model at a chosen scale. On the other hand, density information encoded by the sizes of the extrema influences the interpretation of the R^2 score for a given partition, where a high R^2 score for a partition with low point density may not be trustworthy.
Closed user feedback loops help us refine our visual encoding of the topological map. For example, the extrema in the grey region are connected via color-coded, user-adjustable, cubic Bezier curves, each of which represents a Morse-Smale cell (i.e. a data partition). Such a representation affords a level of flexibility to counteract visual clutter. Each partition is identified by one of nine colorblind-safe colors [64] that are used throughout the visual interface. The topological map is used as a high-level atlas to orient users in the data exploration process. It enables efficient comparisons among partitions across multiple scales. The user can select local extrema and partitions via its interactive interface for in-depth analysis via the remaining views. We also defer additional geometric information to other views to avoid sensory overload.

4.1.3.2 Scatterplot Projection

The scatterplot projection is a common tool familiar to nuclear scientists. It allows the user to select up to three dimensions to be mapped to spatial coordinates, and a potential fourth dimension to be mapped to color. The dimensions of choice include the input parameters, the observed and predicted output parameters, and the residuals of the ROM. The users are therefore provided with detailed spatial information on demand. The 2D height function example in Figure 19 includes a 2D scatterplot (involving two inputs x and y in B1), a 3D scatterplot (involving x, y, and an observed output $f(x, y)$ in B2), and a 3D scatterplot for the local ROMs (involving x, y, and the predicted output).

4.1.3.3 Sensitivity View

We show per dimension coefficients associated with each partition of the data in signed, horizontal bar chart format. The coefficients used include the linear coefficients, the Pearson correlation coefficients, and the Spearman rank correlation coefficients. The user therefore gains first derivative behavior for each selected partition. There are two important pieces of sensitivity information to convey in our visual encoding: the sign and the relative magnitude of the different coefficients. Using signed bar charts, this information becomes available at a glance, and the ubiquity of bar charts ensures their universal interpretation. These bar charts are clustered by partition for easy comparison.

4.1.3.4 Fitness View

For each partition, the fitness view reports the fit accuracy as well as its incremental improvement by increasing the dimensionality of the fit. Based on the linear coefficients described in the sensitivity view, the dimensions are ordered by their magnitude and visualized via vertical bar charts. Based on this sorted order of dimensions as a heuristic, we build lower-dimensional linear fits beginning with only the dimension with the largest coefficient in magnitude. We then iteratively add one dimension at a time and recompute an updated linear model. For an *m*-dimensional dataset, we will end up with *m* different linear fits and for each fit; we compute the R^2 score given in Section 4.1.1. The use of vertical bar charts conveys the stepwise improvement of adding a dimension to the regression model. Typically (but not always), the largest changes in R^2 score occur at the beginning, and we are interested to know at what point the value added by increasing the dimensionality becomes negligible, signifying potentially extraneous dimensions.

4.1.3.5 Secondary Visual Encodings

We include a few secondary visual encodings to enhance and validate insights obtained from the primary ones. The persistence diagram is the classic tool in the form of a 2D scatterplot used for separating signal from noise [29]. In a nutshell, a point in the persistence diagram corresponds to a topological feature represented by the pairing of two critical points. The *persistence* of a point (and its corresponding feature) in the diagram is its distance to the diagonal. When selecting an appropriate scale for partition-based SA, the chosen vertical line in the topology map corresponds to a dotted line in the persistence diagram; a large separation between clusters of extrema in the topology map corresponds to a large separation between clusters of points in the persistence diagram. The persistence diagram provides a standard and less cluttered view of the distribution of topological features in the data, and offers a complementary and validating alternative when selecting the appropriate scale. In addition, we also include a persistence barcode [16] that encodes similar information as the persistence diagram as another alternative for visual comparison.
Histograms are provided on demand and allow the user to visualize the distribution of inputs, outputs, and various computed values on a per partition basis. Histograms also provide an interactive filtering system. This filtering system allows the user to sub-select portions of the data based on ranges of individual dimensions by clicking and dragging to select bins of a single histogram. These alternative visual encodings are included as part of our prototyping stage, and they are implemented in our final tool to offer diversity and increase efficacy.

4.2 Augmenting the Approximate Morse-Smale Complex

It is important to note that the underlying Morse-Smale complex is approximated according to available data. The approximation error can result in somewhat unexpected results when using persistence simplification alone to select the appropriate partitioning of the data. The Morse-Smale complex and its related topological structures are prone to artifacts resulting from variable sampling density, gradient estimation, boundary effects, etc. Standard ε-persistence simplification can thus simplify truly significant features before more minor ones caused by any one or more of the issues listed. This can lead scientists to focus time and effort describing non-salient features in the data while neglecting more important ones. By gaining a better understanding of the particular issues that can arise in the point cloud approximation of the Morse-Smale complex and providing alternative modes of augmenting and navigating the hierarchy, we hope to empower scientists to more quickly identify artifacts and turn their attention to more meaningful trends.

Generalized topological simplification exists for the low-dimensional Morse-Smale case [36,64,67] and Reeb graph shattering [40] demonstrates one way of further segmenting the data to circumvent the issue of disjoint segments for the Morse-Smale approximation in high-dimension. As part of an ongoing development, we are working to more formally enumerate issues occurring as a result of the approximation scheme and offer a more user-guided approach to traversing the Morse-Smale hierarchy. In addition, we plan to explore ways of augmenting the topological partitioning with more traditional partition-based regression techniques based on splitting the data via axis-aligned or projection aligned cuts all while maintaining a coherent hierarchy. The goal is to demonstrate these artifacts and the proposed solution on both synthetic and real-world datasets existing in both low and high (> 3) dimensional spaces.

4.2.1 Related Works

The proposed methodology combines concepts from topology and partition-based regression, as such, we review the existing literature of both and their ties to prior works in visualization, so that we can properly motivate our work.

4.2.1.1 Topology-based Methods

Topological constructs of scalar field data such as contour trees [47], Reeb graphs [58], and merge trees [54] are useful for understanding the evolution of connected components over the levelsets of the data and can all be applied in arbitrary dimensional domains. However, when studying the sensitivity and gradient of a system, as is the focus of this section, they act as poor summaries of sensitivity information. This is due to the fact that these constructs encode an entire connected component into a single visual element, however the gradient direction and magnitude can vary drastically within each component of a levelset.

The Morse-Smale complex on the other hand partitions the data into monotonic patches at the finest level. Persistence simplification can be used in this context to smooth the patches and thus the data can still be treated as being well approximated as a monotonic region. The algorithm used by Gerber et al. [33] can be used to approximate this structure in aribtrary dimensional domains. In fact, Gerber et al. [35] demonstrated the effectiveness of the Morse-Smale complex in fitting simple and intuitive local models to data. This concept was extended to performing sensitivity analysis studies on non-monotonic domains in the work of Maljovec et al. [51]. The same underlying algorithm has been used to construct extremum graphs which are able to help scientists better understand the connectivity and shape of extrema (sharp peaks vs. plateaus) in high-dimensional domains [26].

Several works have focused on manipulating the extracted topology of a dataset in order to conform to a user's needs. Tierney et al. discuss applications where standard ε-persistence is not intuitive and have proposed methods for arbitrarily removing allowable features in the low-dimensional setting [66,67]. Gyulassy et al. allowed users the ability to edit an existing MS-complex again in the low-dimensional setting [36]. We seek to extend this notion to the high-dimensional approximation in order to provide a similar level of interactivity.

Our work is also closely related to the concept of Reeb graph shattering [40], and in fact, Reeb graph shattering could easily be incorporated into our methodology. Reeb graph shattering removes 1-cycles from each of the Morse crystals in order to create more geometrically coherent partitions. By utilizing user-defined cuts, we seek to create better local linear fits of the data while maintaining a coherent global map of the data where partitions are shown as connected when they share a boundary.

4.2.1.2 Partition-based Regression Methods

Partition-based regression techniques can be separated into fully automatic techniques such as regression trees [16] and the user-driven techniques more commonly seen in visualization systems such as in May et al. [52], and Muhlbacher et al. [54]. Automated techniques systematically partition the domain, typically in a greedy fashion, by splitting the data along either an original axis-aligned dimension or a reduced axis after performing dimensionality reduction. The criterion for splitting the data varies from minimizing numerical error [8,16,19,23,33] to more geometry and topology-based criteria [35,44].

The aforementioned methods that rely on axis-aligned cuts tend to over-segment the data especially in cases where the “natural” cut does not lie along a hyperplane in the data. For this reason, Li et al. [44] utilized a dimensionality reduction technique called principal Hessian directions to define a direction that best captures variation in the derivative of the data in order to identify directions across which the data can be cut. A problem with this method is that it only utilizes the average Hessian matrix over the entire data, and so struggles with periodic data where the positive and negative gradients effectively cancel each other out. In addition, the greedy nature of this and many of these algorithms does not take into account the global structure of the data. In this way, the Morse-Smale complex is more robust since it supports boundaries of arbitrary shape, but can also be less interpretable in high-dimensional spaces. For this reason, we begin with the Morse-Smale complex in order to automatically find such undetectable features and then allow the user to use more interpretable cuts where necessary.

Due to their simple cuts/decisions at each level, regression trees are typically visualized using decision trees (dendrograms) [10,13,55]. However, regression trees are amenable to most hierarchical visualization modes such as sunbursts [13] and treemaps [69]. In this work, we provide a visual map that allows users to visualize the current set of partitions geometrically in addition to visualizing the entire hierarchy via a multiple coordinated views approach.

The goal of our process is to create more effective partitions for understanding the local differences in sensitivity of a dataset. So, we include here the works on visualizing regression models and their associated sensitivity information.

HDViz [33] and its extensions [48,49,50] encode sensitivity and density information into a skeleton of the topology and provide auxillary views for more refined analysis. The work of Maljovec et al. [51] more explicitly encodes the sensitivity and fitness of the partitioned data into bar charts. The remaining works focus on a single regression model rather than partitioning the data and providing local fits.

Many methods provide point wise, local sensitivity, which is more amenable to drill-down tasks such as optimization rather than the summarized information we seek to provide. Such methods can either not provide enough information when the visualization is focused around a particular location [11,14,60] or create unnecessary clutter when trying to extract trends in the data [19,20,36].

HyperMoVal [60] allows users to visually validate and explore a model built using support vector regression (SVR) [64] against real data in order to identify and understand regions of poor fitting. Berger et al. [11] focus instead on local optimization by providing local sensitivity information on and around a focal point in a focus+context visualization. Vismon [14] uses sensitivity information to perform trade-off analysis. Glyph-based methods augment scatterplots with pointwise sensitivity information [19,20,36].

On the other side of the spectrum, global sensitivity analysis is not amenable to understanding non-monotonic domains. The heliograph [29] and Slycat [27] are both examples that utilize canonical correlation analysis to provide global sensitivity information. When the data is decomposed appropriately these methods can be applied on each partition and then compared as in Maljovec et al. [51] where linear sensitivity coefficients, Pearson coefficients, and Spearman rank correlation coefficients are calculated.

4.2.2 Approximation Errors

Boundary artifacts can occur whenever the data is restricted to a manifold with boundary, however the arbitrary arrangement of points can make such issues more prevalent as often the neighborhood graphs do not contain the convex hull of the point set and so there is not a smooth integral line along the boundary. Therefore, such "jagged" boundaries can create many small local minima or local maxima. Typically, these problems can be alleviated using persistence simplification, but as real-world examples have shown, e.g. the data used in both Maljovec et al. [51] and by Davani-Tavakol [64], the combination of variable sampling density and neighborhood graph selection can result in these features being more persistent than true features in the data. This boundary problem can also happen internally in the data where artificial voids are created 1. As noted above, these issues occur due to the sampling dispersion of points, which is typically beyond our control, and not something we can modify. The selection of neighborhood graph can also be considered the source of such artifacts. A sparse, under connected graph may result in such large virtual voids in the data and selecting a denser neighborhood graph can alleviate this issue, but can also cause create other issues as seen below.

A denser graph can over connect the domain causing the Morse-Smale extraction algorithm to skip over a local extrema failing to identify it at all. Alternatively, an over connected graph may also create long edges that connect points on the interior of adjacent or even non-adjacent cells extending the identified boundary far into the interior of these cells. This results in false saddles whose function value is far below or above the actual boundary between cells. In addition, a dense graph can create crossing edges that can lead to crossing integral lines. Crossing integral lines can result in intersecting partitions though typically this occurs only at the boundaries.

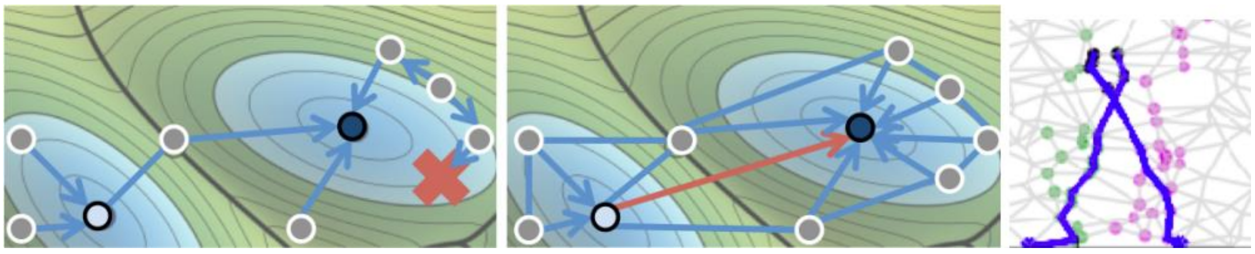
Figure 20 - Left: An under connected graph falsely identifying a point as a minimum; Center: An over connected graph directly connects two local minima causing the left one to not be identified as such; Right: An example of a graph with crossing edges resulting in crossing estimated gradients.

We acknowledge these problems exist and that some of them are unavoidable using the given algorithm. Some can be alleviated through pre-processing such as ensuring that no edges intersect, but others are hard to determine a priori such as whether a void actually exists in the data or is created by the imposed graph. The intention is to make scientists aware of these issues and provide methods for identifying and circumventing such problems through interactive exploration keeping in mind the goal is not always preserving the topology exactly, but decomposing the data into coherent structures for local analysis.

4.2.3 Methodology

In order to support interactive data splitting and merging in a sensible manner, we have extended the software explained in Section 4.1.3, which utilizes a multiple coordinated views approach that allows us to evaluate the effectiveness of the current partitioning via several methods and exposes faults in the current partition readily.

4.2.3.1 Visual Encoding

A typical workflow of our system will begin with the user exploring the persistence hierarchy of the extracted topology. For this purpose, we utilize the topology map to show how partitions connect to extrema and having their persistence being visually represented by their leftmost coordinate. Additionally, we provide a contextual tooltip (see Figure 21) that displays the number of samples in a given partition along with information regarding the span and dispersion of samples in each dimension.

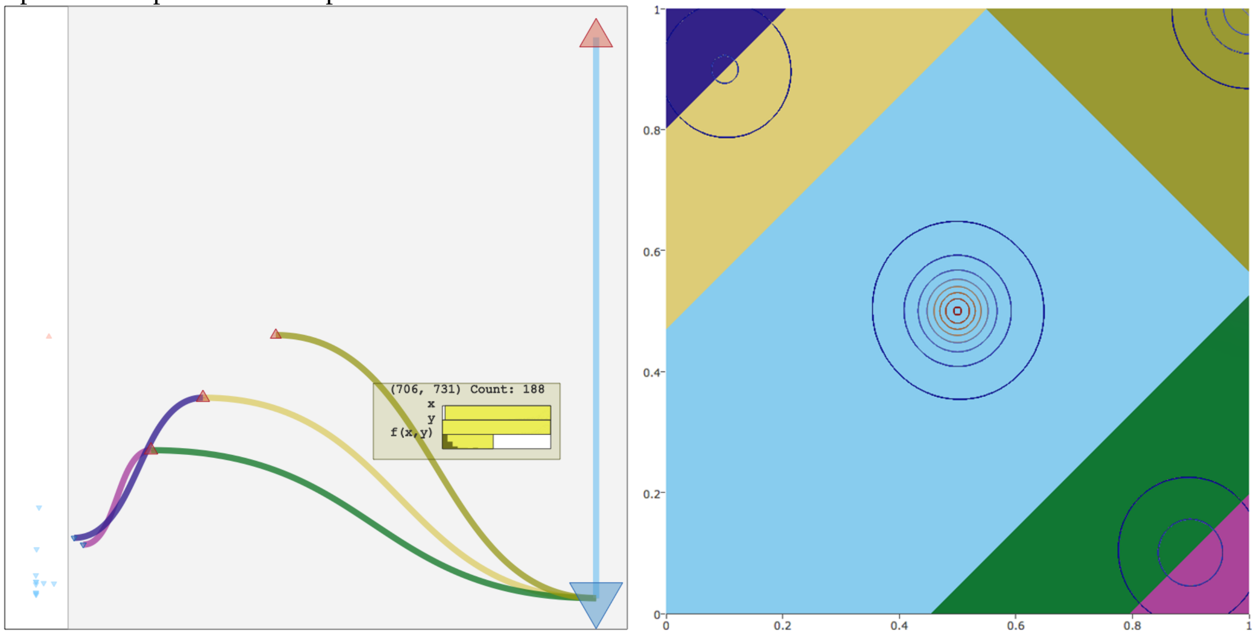

Figure 21 - Left: The topology map of a 2D analytic test function. Right: A contour map showing how the 2D domain is partitioned.

In terms of sensitivity analysis, partitions with few data points or poor fitness levels are both less desirable for summarizing the data. However, both of these pieces of information are lacking in the topology map view. To this end, we introduce a new view of the data that associates a glyph with each partition and arranges the glyphs in a way that provides some global geometric context. The glyphs are sized according to the number of data points in the corresponding partition, and encode sensitivity and overall fitness information in order to inform when a partition should be merged back into its parent, or split to obtain finer granularity. An example of this layout is given in Figure 22.

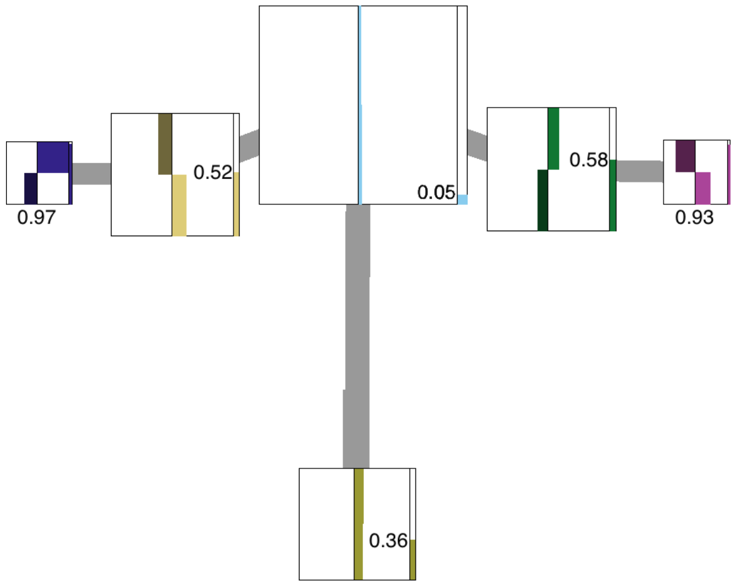

Figure 22 - The proposed graph layout view showing the connectivity of the partitions from Figure 21 with their sensitivities and linear fitnesses embedded.

The glyphs are laid out in a three-step process. First, we identify the centroids of the partitions. Next, these centroids are projected into 2D using PCA to initialize their positions. Lastly, we connect the glyphs using weighted edges and apply a mass-spring layout to obtain the final positions. The edges between glyphs/partitions are constructed/weighted according to the number of points on the shared boundaries obtained from the neighborhood graph imposed previously in the Morse-Smale computation. The thicknesses of these edges are proportional to the number of points belonging to the corresponding boundary. In higher dimensional spaces such a graph can become dense very quickly as the data points are provided more directions to connect with one another. To mitigate the effects of this, we provide a threshold on the minimum number of edges needed to report a boundary. Alternatively, we have been experimenting with using the extremum graph first introduced in the topological spines work [26] to connect the partitions according to how their extrema are connected.

The glyphs are each represented as a bar chart and are augmented to show the overall linear fitness of the partition with a juxtaposed vertical bar and numeric value overlaid. Figure 23 shows in detail the information encoded in this glyph.

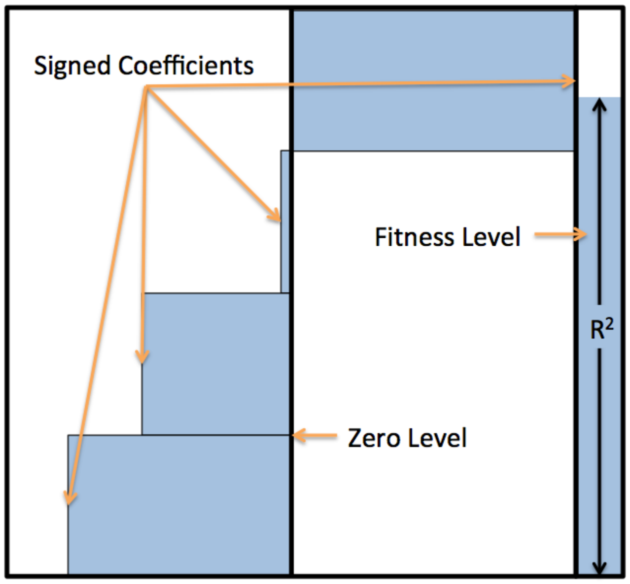

Figure 23 - A bar glyph representing sensitivity through a stack of signed horizontal bars and linear fitness through a juxtaposed vertical bar.

Lastly, we provide users with a dendrogram view to show the hierarchical relationship between the partitions identified by the Morse-Smale construction as well as where any manual cuts (described below) in the data were made. The user can arbitrarily traverse any sub-tree to any depth they choose without affecting the remainder of the tree. In this way, we begin to explore how to partition the data besides using a single persistence threshold. This is useful in a regression setting as smaller features may cover more area and thus have more data to provide a more trustworthy fit than can be achieved on a sharp spike with only few data points. Also, we provide the users with an interface for creating manual cuts based on various projections of the data. This notion is borrowed from work on partition-based regression trees. One goal in this work is to understand the implication of creating manual cuts and when and how the remainder of the topology can be maintained. This is discussed in more detail in the next section.

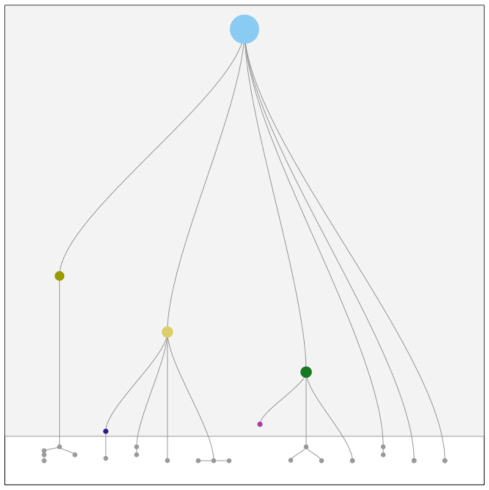

Figure 24 A dendrogram representing the topological hierarchy for the test function presented in Figure 21.

4.2.3.2 Data Splitting

In addition to allowing users to navigate to arbitrary depths of the persistence hierarchy on any branch of the tree, we also allow the users to augment the hierarchy by injecting their own splits at any level in the hierarchy. In order to maintain coherency within the hierarchy, we should make every effort to track the effects the new

cuts have on the topology within these newly created segments, and attempt to conform the topology to match the user's needs. It is important to note that this process is similar to the approach given by Gyulassy et al. [36] in that only the boundaries are affected. A boundary is the set of all points such that the label of a point p differs from the label of at least one of its neighbors. So, we must only conform the boundary created by the new cut. In this way, any changes we make to the topology of the sub-domain will not affect the rest of the topology as long as the global minimum and maximum of the sub-tree remain.
The window shown in Figure 25 demonstrates the working prototype that allows the user to project the data onto a single dimension that is either part of the original space or determined by one of the principal Hessian directions allowing users to interactively decide where to next split their data. Currently, these manual cuts are placed into the topological hierarchy and override any sub-tree associated to the partition that has been split.

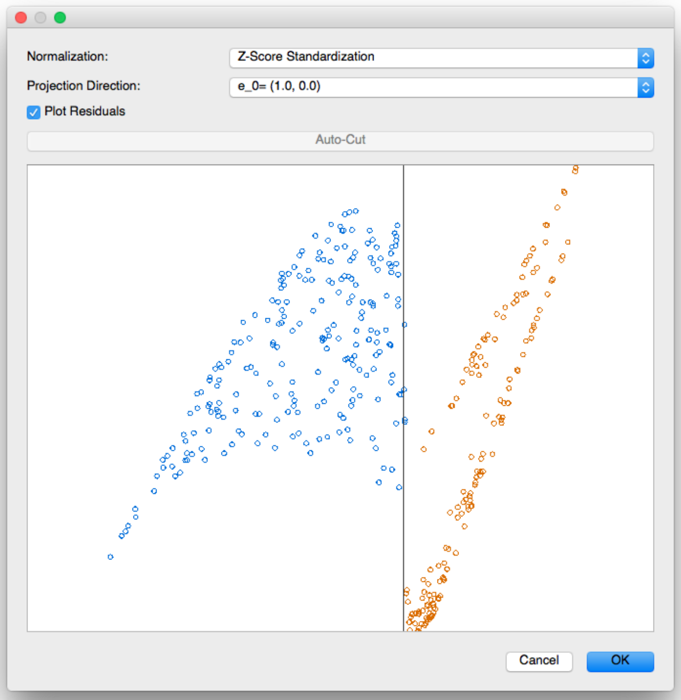
Normalization: Z-Score Standardization
Projection Direction: e_0= (1.0, 0.0)
☑ Plot Residuals

Figure 25 - A demonstration of the manual cutting interface where the user may select from a pre-defined list of projections and can then place a cut orthogonal to that projection. A preview of the user's cut is shown.

4.2.3.3 Evaluation

As a first step, we have begun exploration of several different synthetic and real-world datasets that can benefit from interactive exploration of the Morse-Smale hierarchy. The example provided here is a general dimension flattened sinusoidal function:

$$f(x) = \left(\prod_{i=0}^{d} \cos(\pi x_i)^{\frac{1}{7}}\right)(0.1 + K(\boldsymbol{x}))$$

$$K(\boldsymbol{x}) = (1 - \|\boldsymbol{x}\|_2^3)^3$$

The power on the cosine creates alternating plateau and valley-like shapes in the data and the combination with the tricubic kernel function, K, dampens these features away from the origin, but the offset 0.1 keeps them from going away entirely. We evaluate this function on the domain $[0, 1]^3$, so as to capture the large structure at the origin and its neighboring features in one quadrant of the data. The Morse-Smale algorithm should detect the sinusoidal undulations in the data, however it will not be able to obtain accurate linear fits due to the "*S*" shape of each monotonic patch in the data. This function thus provides us a good example for being able to use manual cuts in order to obtain more accurate linear fits. Note that beginning with the Morse-Smale decomposition allows the PHD algorithm to operate locally on monotonic data, which is an ideal condition for the PHD.

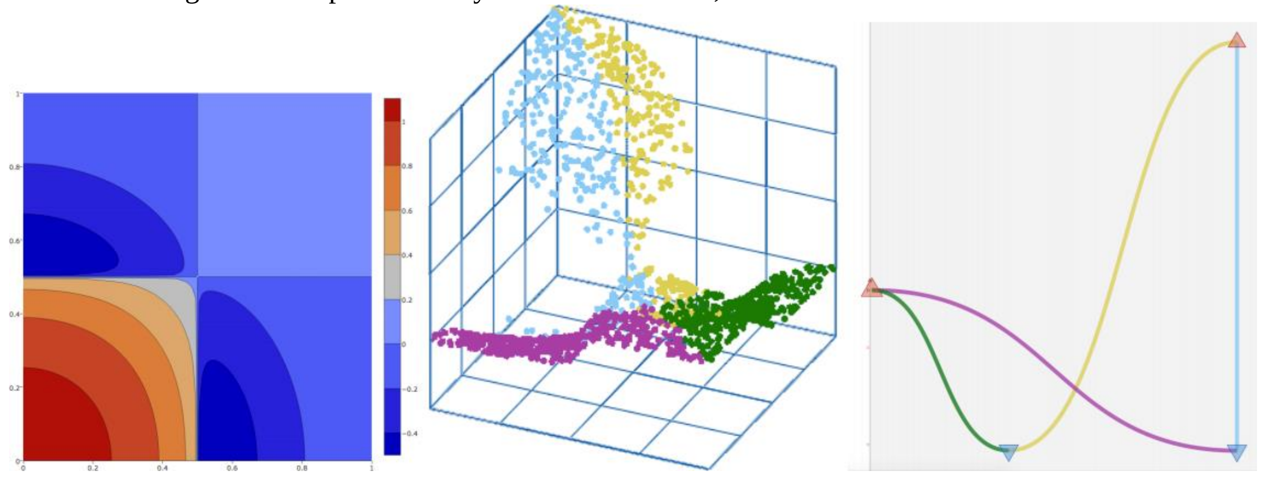

Figure 26 - A 2D example of the flattened sinusoidal function amenable to the new proposed framework. From left to right: the contour heatmap, a 3D view where the data is partitioned according to an initial setting of the Morse-Smale complex hierarchy, and the corresponding topology map showing how the extrema connect.

Figure 26 and Figure 27 show an analysis of the two-dimensional version of the flattened sinusoidal function in 2D. Figure 26 demonstrates that the topology map shows good separation between feature and noise at a level with four Morse-Smale crystals connecting each of two local maxima and two local minima. As predicted, though, the linear fitness associated to each partition (reported in the top left image of Figure 27) is low. We then perform manual cuts iteratively using the interface shown in Figure 25 to arrive at partitions of increased accuracy and refinement, as shown in the second and third columns of Figure 6.18. This work is ongoing and is to be extended to work on real-world examples from nuclear engineering.

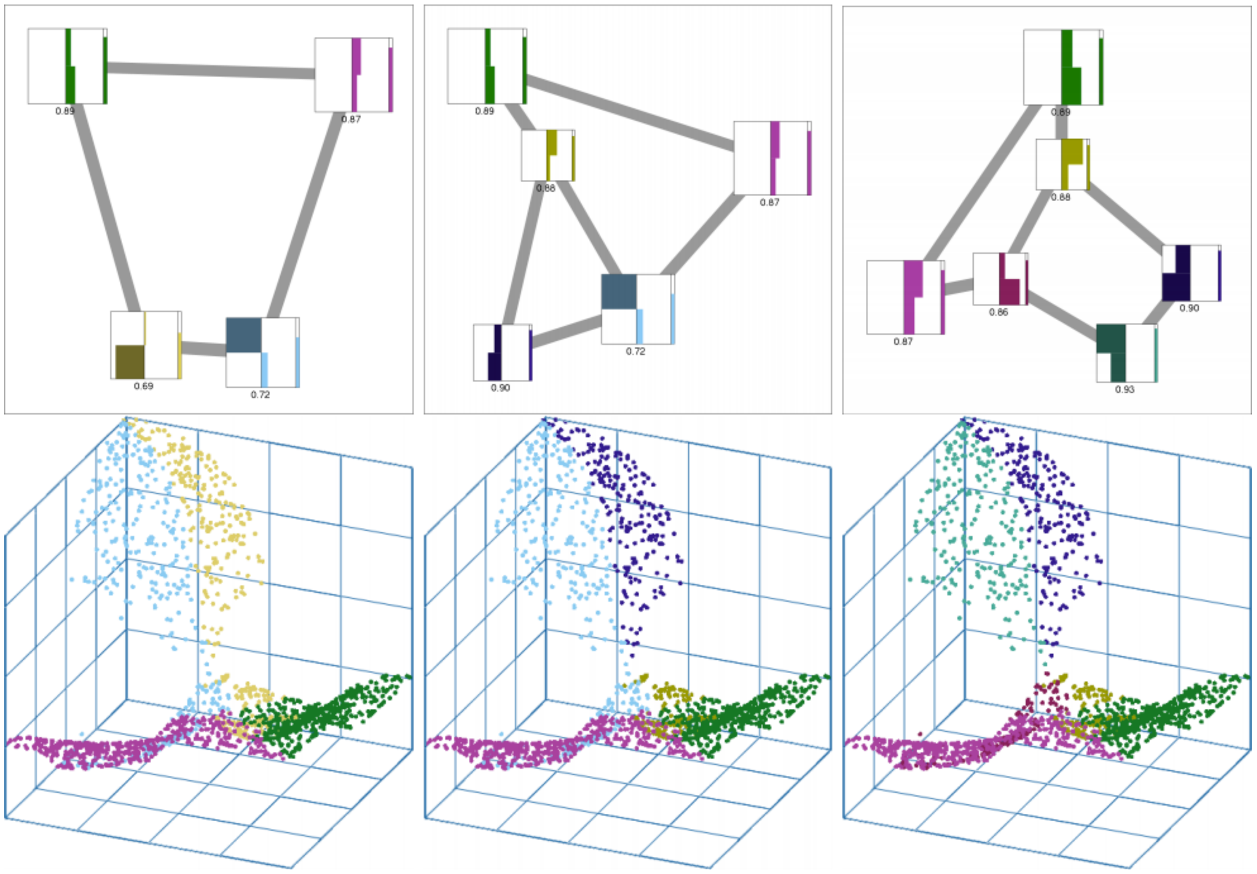

Figure 27 - Iterative refinement of the function introduced in Figure 26 using manual cuts of the data. In this case two additional partitions were discovered that increased the R^2 values to exceed 0.85 for each partition. This interface allows us to increase the fidelity without over-segmenting the data.

4.3 Application

We consider an analysis of a nuclear fuel rodlet simulation performed using the BISON software [38], a modern finite-element based nuclear fuel performance code. The rodlet is axisymmetric and composed of ten stacked UO_2 pellets surrounded by a zirconium alloy shield known as the cladding. We track the midplane von Mises stress occurring on the 3D cladding. High stress can cause the cladding to crack and allow radioactive gas to leak into the plant environment. The simulation looks at a ramping of the linear power in the reactor from time t=0 to t=10000 (seconds), whereupon the power level maintains a constant value for the remainder of the simulation. The linear power begins at 0 W/m at t=0 and linearly climbs to a value of 25000 W/m before leveling off.

The goal is a better understanding of the physics happening at the *contact point* that occurs when the fuel rodlet expands to the point of touching the cladding. The stress on the cladding at t=0 is due to a compressive force from the water pressure outside the cladding. As fission occurs, the fuel rod expands as it is heated due to thermal expansion and swelling from the release of fission gas within the microstructure of the fuel. Once contact is made, thermal expansion (of the cladding) exerts a force that counteracts the compressive force from the water pressure, and therefore the stress in the cladding decreases until these forces reach equilibrium. After the equilibrium point, the expansive forces on the cladding become dominant, causing stress on the cladding as it expands.

In the above scenario, contact is not indicative of a failure state, and is actually expected in these types of environments. The described problems arise only when the stress in the cladding becomes too high. For this study, the scientists vary three input parameters during the simulation: (a) the linear power scaling factor (**power_scalef**), a multiplier for the previously described ramping linear power; (b) the grain radius scaling factor (**grainradius_scalef**), a multiplier for the size of microstructure elements known as the grains and related to the swelling caused by the fission gas; and (c) the thermal expansion coefficient for the fuel rodlet (**thermal_expansion**), which dictates how quickly the rod expands and thus how quickly it contacts the cladding. The output parameter of interest is the final midplane von Mises stress **midplane_stress** (a scalar quantity derived from the Cauchy stress tensor useful in determining at what point the cladding may yield/crack) recorded after a simulation time of t=10^6 seconds. All parameters are scaled using z-score standardization to align with both domain and algorithmic practices.

4.3.1 Anomaly Detection

Exploration begins with an initial dataset sampled from a uniform 10×10×10 grid of the parameter space. During the exploration process illustrated in Figure 28, the scientists notice an unexpected behavior within the sensitivity view. An increase in the **thermal_expansion** of the fuel is expected to it to expand more rapidly and come into contact with the cladding sooner. The result would cause everything to precipitate faster: a simulation reaching the equilibrium point leads to a higher stress in the cladding; a simulation ending before reaching the equilibrium point (exhibited via lower values of **power_scalef**) results in a lower stress. The expectation is that the higher the **thermal_expansion** is, the higher the stress becomes. However, the linear coefficients for **thermal_expansion** across partitions in the sensitivity view contradict such an expectation: (a) **thermal_expansion** is positive for the green and orange partitions when **power_scalef** is low, and increases when approaching the equilibrium (located at the intersection of the three partitions); (b) **thermal_expansion** is negative for the violet partition when **power_scalef** is high, and decreases past the equilibrium. The Pearson and Spearman rank coefficients (not shown here) exhibit similar behaviors.

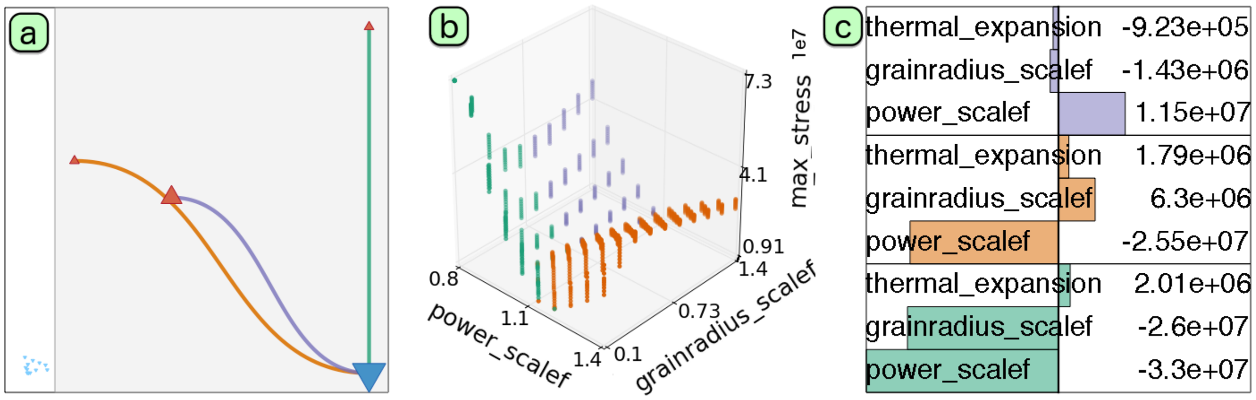

Figure 28 - SA for the initial fuel design data. (a) Topological map used to separate signal from noise. (b) Scatterplot projecting the most significant input power_scalef, grainradius_scalef, and the output max_stress. (c) Linear coefficients.

4.3.2 Workflow with New Data

After fixing the bug associated with the simulation input, a new dataset is generated for SA from 9517 simulations using Monte Carlo sampling of the three input dimensions from independent distributions. Starting with the topology map in Figure 29a, a large horizontal gap between two clusters of extrema signifies well separation between signal and noise in the data. The scientists therefore select a scale (a vertical line) that preserves the three extrema in the grey region as topological features, resulting in two partitions of the data domain. Such a selection is also supported by observing a large gap between two clusters of points within the persistence diagram (Figure 29b), where one cluster contains points near the diagonal $y = x$ and the other contains points far away from the diagonal. As a first order approximation, the scientists build local linear regression models for these two partitions.

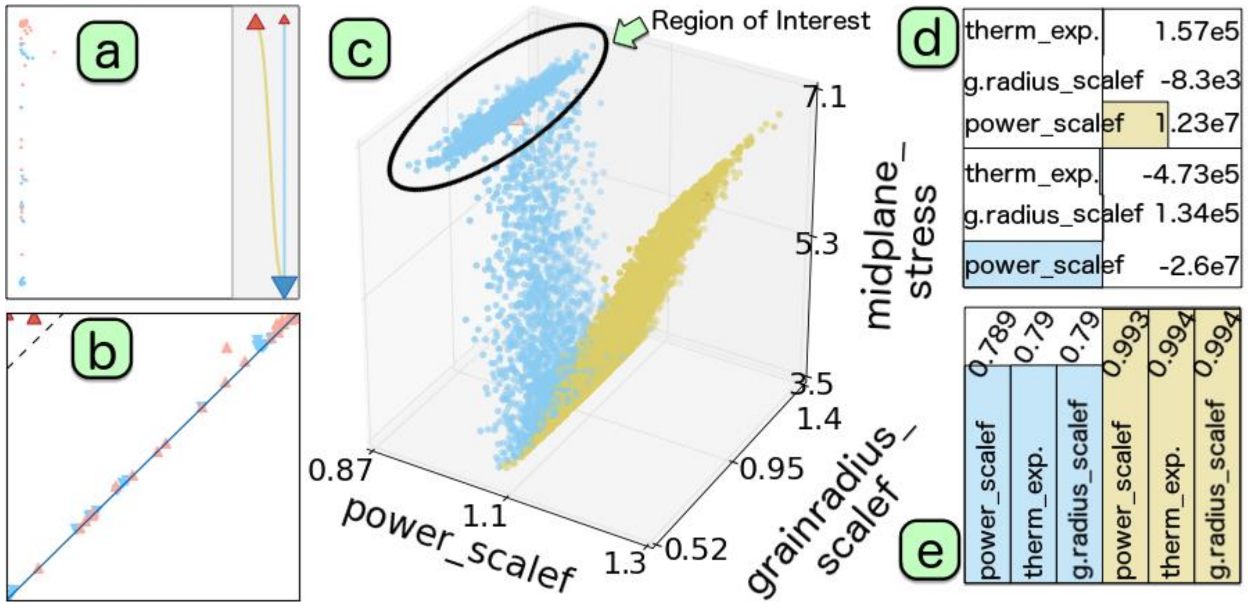

Figure 29 - SA of the new nuclear fuel dataset: (a) topology map, (b) persistence diagram, (c) linked scatter plot projection, (d) linear coefficients, and (e) fitness view with stepwise R^2 scores.

The resulting linear coefficients are shown in Figure 29d where the **power_scalef** dominates the behavior of the **midplane_stress**. Furthermore, the fitness view (Figure 29e) demonstrates that little to no information is gained by incorporating the remaining two dimensions, in either partition of the data. The fitness view also shows that the blue partition is not well described by a linear fit of any number of dimensions (i.e., with all three R^2 scores valued roughly at 0.79). The scientists investigate further by projecting the data onto a scatterplot (Figure 29c) that includes the most sensitive input dimension **power_scalef**) and the output dimension **midplane_stress**). From this scatterplot, they detect the nonmonotonic behavior of a region of interest (ROI) within the blue partition, which has low **power_scalef** and high **midplane_stress**. The scientists then focus on a more refined analysis at a finer scale to try to capture the behavior of the ROI within its own partition.

Figure 30 shows the results after decomposing the domain into three and four partitions at finer scales. The scientists, guided by both the topology map and the scatterplot view, iteratively choose finer and finer scales until the ROI separated itself from the larger blue partition. The first level of refinement produces three partitions in the data (Figure 30a). Compared to the approximation with two partitions (Figure 29c), the newly constructed magenta partition has relatively low point density (as shown in Figure 30b) and does not correspond to the ROI. Under the second level of refinement (Figure 30c), the ROI forms its own partition in green, as illustrated in Figure 30d. In addition, there is significant improvement in the fitness for the blue region. That is, the three R^2 scores increase from roughly 0.790 to 0.898 (Figure 31). Therefore, extracting the desired ROI requires two additional levels of refinements.

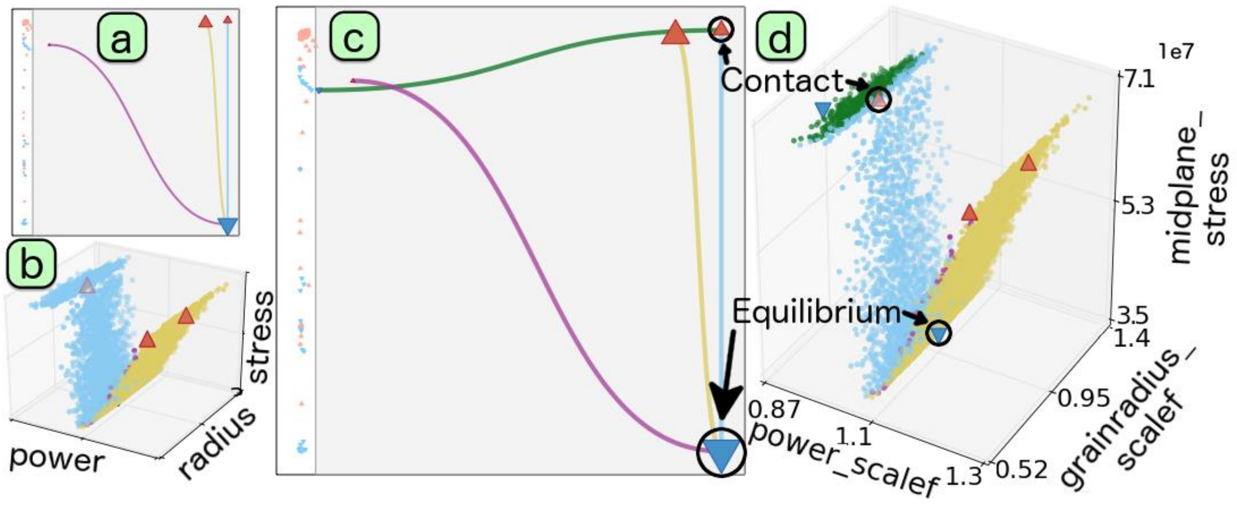

Figure 30 - SA of the new nuclear fuel data under two refined settings: topology maps and scatterplots with three partitions (a)-(b) and four partitions (c)-(d), respectively.

Under the refined setting with four partitions, the scientists are able to obtain insights that are aligned with their expectation and domain knowledge. Using topology-based domain partitioning allows them to decompose the domain in a physically meaningful way. In particular, the four partitions within the data domain are shown to correspond to various stages of the simulation. The scientists focus on examining the characteristics of each partition and how the partitions interface with one another. First, the interface between the green and blue partitions represents the *contact point* where the fuel touches the cladding (Figure 30d). Within the green partition (which has very low **power_scalef** values), the net pressure acting on the cladding originates from the external water pressure. Second, the interface between the blue and the combined magenta and gold partitions corresponds to the *equilibrium point*. The blue partition represents the simulations where the expansive forces of the cladding begin to counteract the compressive water pressure force. Within the blue partition, as **midplane_stress** decreases and **power_scalef** increases, the simulation moves closer to the equilibrium. Finally, the remaining two partitions (gold and magenta) contain scenarios after the equilibrium point between compressive and expansive stresses, and therefore within these partitions, an increase in **power_scalef** leads to an increase in the dominating expansive stress (i.e., **midplane_stress**). In terms of sensitivity information (Figure 31 left), **power_scalef** has a strong positive effect on **midplane_stress** across the partitions with the exception of the blue region, where it is strongly negative.

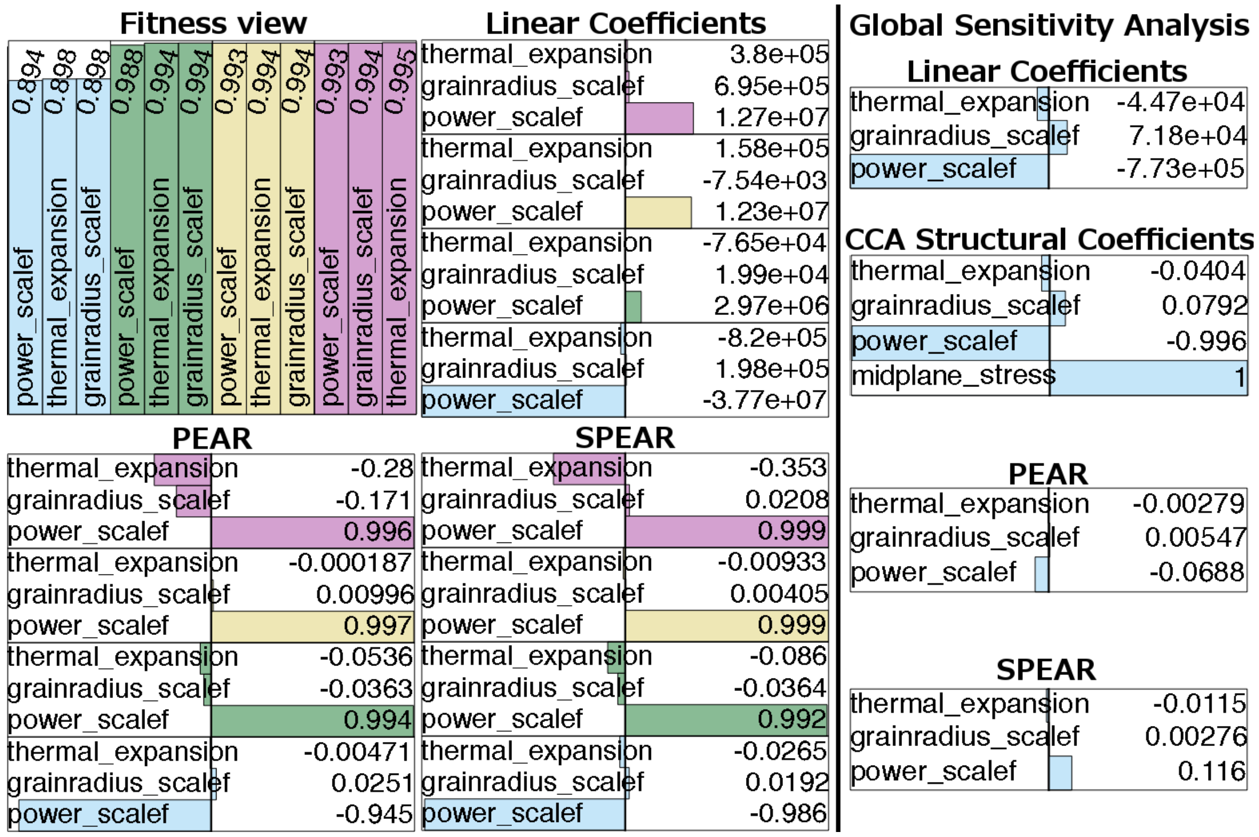

Figure 31 - Left: sensitivity information of the new nuclear fuel data under the refined setting with four partitions. Right: global SA of the same data.

Finally, the scientists compare the results from the partition-based local SA Figure 31 left) with that of a global SA (Figure 31 right). For the global SA, the linear coefficients and CCA structural coefficients [27] are used to identify **power_scalef** as the most sensitive parameter, yet its sign heavily depends on the amount of data on either side of the contact and equilibrium points. Similarly, the non-monotonic global behavior masks the sensitivity associated with **power_scalef** for both PEAR and SPEAR coefficients. On the other hand, the topology-inspired, partition-based SA is able to capture three distinct behaviors in the data, and it highlights the high sensitivity associated with **power_scalef** within each partition.
Using our framework, the scientists have validated the behaviors of a nuclear fuel simulation to be well aligned with their expectations. They are actively investigating higher-dimensional problems where topological partitioning could offer them additional insights that are not readily available via traditional visualization such as scatterplot projections. The scientists have validated our approach by performing SA faster and with higher accuracy. Our framework also offers a new paradigm in rethinking about SA via topology in nuclear engineering.

4.4 Future Development

Much of the work presented in Section 4.2 is ongoing and we hope to provide results on real-world data soon. In addition, we have shown a proof-of-concept ability to visually interact with the data generated by RAVEN while still in the RAVEN context. Moving forward, we hope to generalize this multiple coordinated view architecture to allow more robust user interaction within RAVEN. For example, the current plotting methods available in RAVEN require the user to include the full context of their desired plots within the RAVEN input. This new system allows the user to construct plots on the fly using a visual interface whereby they can interactively select portions of data and change axis and plot type without the need to exit and re-run RAVEN.

5. MINOR ENHANCEMENTS

5.1 Software improvements and documentation

In addition to the major modifications to RAVEN, we have implemented a few minor modifications:

- Increased number of tests that use external models instead of codes (so they can be tested more easily)
- Improved virtual machine testing and improved High-Performance Computing cluster testing.
- Improve quality assurance documentation
- Improved installation guide in user manual
- Switched RAVEN to use C++11

5.2 Minor Development on Limit Surface

INL has made a concerted effort toward developing the RISMC pathway risk analysis tool [71]. An integral part of this pathway is characterizing a simulation space into areas of system failure and system success. Consider analysis of a nuclear power plant simulation under duress such as in the station blackout scenario. Under this pathway, scientists want to understand the threshold, or limit surface, existing in the simulation input space that differentiates between system recovery and system failure. With such information at hand, they can begin to design safety systems that push against this limit surface in such a way that maximizes the safety tolerance while limiting the cost of the system. Therefore, being able to extract the limit surface with a specified tolerance is an important problem in increasing the safety of nuclear power systems. To this end, RAVEN has employed methods based on regular Cartesian grids that ensure a specified tolerance by constructing grids of adequate resolution.

As part of future development, we have begun evaluating alternative methods that utilize system topology to help seed the search for the limit surface and limit the number of evaluations to points on or neighboring the limit surface and compare the results to the existing methodology deployed in RAVEN, in terms of both memory and time.

5.2.1 Background

The limit surface problem of nuclear engineering is very similar to a well-studied problem in the field of scientific visualization and image analysis where it is known as isosurfacing. In the visualization community the problem is to visualize a 2D surface existing in a 3D scalar volume where every location is associated a function value. The standard algorithm for isosurfacing is the marching cubes [43] algorithm, which was developed to handle 3D volume visualizations. This method works on a rectilinear grid by using linear interpolation among vertices to determine the surface passing through each discretized cube of the volume. In this method, each vertex of the grid must be evaluated and then each grid cell is processed. Various optimizations of this methodology have been developed which include parallelization, GPU-based algorithms, and the use of acceleration structures such as the KD-tree [43,44,47,51,54]. Unlike in the visualization domain, however, the problems in nuclear engineering can be of arbitrary dimension. To this extent, Bhaniramka et al. [10] developed a methodology for generalized dimensionality that uses a lookup table of all 2^{2^d} configurations for a d-dimensional problem. We quickly see that keeping track of high-dimensional simplices as done in the isosurface case is infeasible and not necessarily intuitive, so instead for the limit surface problem we seek to only find points that bound the limit surface.

The methodology employed in the RAVEN framework is aimed at this later representation. That is, finding points on a discretized domain that bound the limit surface from above and/or below. RAVEN provides acceleration schemes for optimizing such limit surface searches by utilizing surrogate models and a grid-based discretization [1,60]. In d-dimensional space, this amounts to n^d evaluations of a grid with n points per dimension. Even using a surrogate model, this can become cost-prohibitive in moderate dimensionality. The existing RAVEN method thus utilizes a multi-grid approach to cut down on grid evaluation cost [38], however the time to construct this multi-grid structure can still be time-consuming for higher dimensional problems. For this reason, we are evaluating a topology-based approach and evaluating the multi-grid approach method to ensure it is optimized to create a fair comparison to the topology-based approach currently being developed.

5.2.2 Proposed Approach

Van Kreveld et al. [69] demonstrated how the contour tree can be used to extract minimal seed sets for isosurfacing. We employ this methodology in order to expedite the limit surface search problem. By using the algorithm developed by Carr et al. [18], we first extract a minimal set of seed points for each connected component of the limit surface for a specified threshold value. These seed points along with a surrogate model $\hat{p}$ can be used to trace the gradient to a grid point x of specified discretization lying on the limit surface. Each seeded grid point is added to the limit surface and placed on a queue to be processed. We then process each element of the queue, x, as follows.

We evaluate $sgn(\hat{p}\ (x))$ and $sgn(\hat{p}\ (x_n))$ where x_n is a neighboring grid point. If the two evaluate to different signs then the neighbor is added to the limit surface and pushed onto a queue. This process is repeated until the queue is empty. In this way, we only evaluate $\hat{p}$ on the limit surface points and their 1-ring of neighbors providing a near optimal algorithm. The open question in this development is whether we can reduce the number of evaluations at each grid point down from the 3^d needed for the entire neighborhood around a point on the limit surface. Currently, each grid cell that is evaluated is stored in a dictionary thus reducing redundant computation, but also making parallelization less straightforward. One potential idea is to compute the normal to the hypersurface, which should be equivalent to the gradient direction, thus requiring d evaluations per location being evaluated. In this way, neighbors along the normal direction and its negative direction can then be omitted.

The notion of determining the normal to the limit surface also can be utilized to reduce the number of points being stored, for if we can find areas of similar gradient behavior along the limit surface via clustering methods, we can begin to summarize large areas of the limit surface by annotating fewer points with their associated normal vectors ($\nabla\hat{p}$) and shape descriptors to capture the shape defined by that normal vector. Preliminary tests using the topology approach have been used to extract a relatively low-resolution limit surface existing in a seven dimensional input space. We plan to more formally compare the existing methods on low and moderate dimensional test cases.

6. SUMMARY AND CONCLUSIONS

This report presented three distinct development features that have been accomplished during FY 2016. All these activities represent improvements and upgrade of current RAVEN capabilities toward an improved and more user-oriented analysis workflow.

The upgrade of the *EnsembleModel* capability in RAVEN in order to process 1-Dimensional Figure of Merits represents a development that determined the need to abstract the concept of "input realization" in the RAVEN code. RAVEN has extended the concept of "input realization" to any scalar and 1-Dimensional Figure of Merits (i.e. parameter and vectorial uncertainties). This abstraction makes the *EnsembleModel* infrastructure capable to handle the current and future challenges and needs within the RISMC "Industrial Applications" since they rely on multiple models (i.e. codes), communicate through a common interfaced infrastructure, and combine high-fidelity codes with surrogate modeling. Future work in RAVEN is to extend the *EnsembleModel* infrastructure in order to process High-Density fields (both as input and output spaces). This development is needed in order to make the framework able to interface surrogate models that are "trained" on mesh-data (e.g. 3-Dimensional power maps, etc.).

The second portion of this report is about the development and implementation of "validation" techniques for assessing the validity of constructed surrogate models within the RAVEN framework. In order to understand the usability of complex models, it is crucial to have a metric to define "how good", in terms of predictive capabilities, is a surrogate model. We demonstrated the initial development of these validity metrics, overall leveraging the statistical approach of cross-validation. Through these techniques, the RAVEN user is now able to get estimation on how good is the surrogate model prediction and to combine a Surrogate Model and a High Fidelity code in order to accelerate the Uncertainty Quantification process. Future work in RAVEN is to extend the concept of "how good" the prediction is, implementing ad-how metrics for each supported surrogate model in order to get confidences for each evaluation instead of just a global metric (like in the cross validation approach).

The third part of this report shows how the topology-based visualization module in RAVEN can be used to ease the PRA and UQ analysis, overall providing interactive flexibility to the analysts in order to better understand complex models.

7. REFERENCES

1. A. Alfonsi, C. Rabiti, J. Cogliati, D. Mandelli, S. Sen, R. Kinoshita, C. Wang, P. Talbot, D. Maljovec, A. Slaughter, C. Smith. Dynamic Event Tree Advancements and Control Logic Improvements, INL/EXT-15-36758

2. S. Sen, D. Maljovec, A. Alfonsi, C. Rabiti. Developing and Implementing the Data Mining Algorithms in RAVEN. INL-EXT-15-36632

3. A. Alfonsi, C. Rabiti, D. Mandelli, J. J. Cogliati, R. S. Sen, and C. L. Smith. Improving Limit Surface Search Algorithms in RAVEN Using Acceleration Schemes: Level II Milestone, Jul 2015. [Online]. Available: http://www.osti.gov/scitech/ servlets/purl/1244620

4. C. Rabiti, A. Alfonsi, D. Huang, F. Gleicher, B. Wang, H. S. Abdel-Khalik, V. Pascucci, C. L. Smith. System Reliability Analysis Capability and Surrogate Model Application in RAVEN, INL/EXT-16-37243

5. D. Mandelli, C. Smith, A. Alfonsi, C. Rabiti, J. Cogliati, H. Zhao, I. Rinaldi, D. Maljovec, P.Talbot, B. Wang, V. Pascucci. Reduced Order Model Implementation in the Risk-Informed Safety Margin Characterization Toolkit. INL/EXT-15-36649

6. Pedregosa *et al.*, Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011

7. R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, International Joint Conference on Artificial Intelligence, 1995

8. W. P. Alexander and S. D. Grimshaw. Treed regression. Journal of Computational and Graphical Statistics, vol. 5, pp. 156–175, 1996.

9. A. Alfonsi, C. Rabiti, D. Mandelli, J. J. Cogliati, R. S. Sen, and C. L. Smith. Improving Limit Surface Search Algorithms in RAVEN Using Acceleration Schemes: Level II Milestone, Jul 2015. [Online]. Available: http://www.osti.gov/scitech/ servlets/purl/1244620

10. T. Barlow and P. Neville. Case study: visualization for decision tree analysis in data mining, in Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on, 2001, pp. 149–152.

11. W. Berger, H. Piringer, P. Filzmoser, and E. Gr¨oller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction, Computer Graphics Forum, vol. 30, no. 3, pp. 911–920, 2011.

12. P. Bhaniramka, R. Wenger, and R. Crawfis. Isosurfacing in higher dimensions. In Visualization 2000. Proceedings, Oct 2000, pp. 267–273.

13. BigML. (2016) Big ml. [Online]. Available: https://bigml.com/

14. M. Booshehrian, T. Moeller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. Comp. Graph. Forum, vol. 31, pp. 1235–1244, 2012.

15. A. Bouloré, C. Struzik, and F. Gaudier. Uncertainty and sensitivity analysis of the nuclear fuel thermal

behavior. Nucl. Eng. Des., 253:200-210, 2012. {SI} : CFD4NRS-3.

16. L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Monterey, CA: Wadsworth and Brooks, 1984.

17. G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas. Persistence barcodes for shapes. In Eurographics/ACM SIGGRAPH SGP, 2004.

18. H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '00. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000, pp. 918–926.

19. Y.-H. Chan, C. Correa, and K.-L. Ma. Flow-based scatterplots for sensitivity analysis, in IEEE Symposium on Visual Analytics Science and Technology. IEEE, 2010, pp. 43–50.

20. Y.-H. Chan, C. Correa, and K.-L. Ma. The generalized sensitivity scatterplot, IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 10, pp. 1768–1781, 2013.

21. P. Chaudhuri, M. ching Huang, W. yin Loh, and R. Yao. Piecewise-polynomial regression trees. Statistica Sinica, vol. 4, pp. 143–167, 1994.

22. F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Analysis of scalar fields over point cloud data. In ACM-SIAM SODA, 2009.

23. H. A. Chipman, E. I. George, and R. E. McCulloch. Bart: Bayesian additive regression trees. Ann. Appl. Stat., vol. 4, no. 1, pp. 266–298, 03 2010.

24. D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. Discrete Comput. Geom., 37 (1): 103-120, 2007.

25. C. Correa and P. Lindstrom. Towards robust topology of sparsely sampled data. IEEE TVCG, 17 (12): 1852-1861, Dec 2011.

26. C. Correa, P. Lindstrom, and P.-T. Bremer. Topological spines: A structure-preserving visual representation of scalar fields. IEEE TVCG, 17(12): 1842–1851, Dec 2011.

27. P. Crossno, T. Shead, M. Sielicki, W. Hunt, S. Martin, and M.-Y. Hseih. Slycat ensemble analysis of electrical circuit simulations. In J. Bennett, F. Vivodtzev, and V. Pascucci, editors, Topological & Statistical Methods for Complex Data, Mathematics and Visualization, pages 279-294. Springer Berlin Heidelberg, 2015.

28. W. Cui, H. Zhou, H. Qu, P.C. Wong, and X. Li. Geometry-based edge and clustering for graph visualization. IEEE Transactions on Visualization and Computer Graphics, 14 (6): 1277-1284, 2008.

29. A. Degani, M. Shafto, and L. Olson. Canonical correlation analysis: Use of composite heliographs for representing multiple patterns, in Diagrammatic Representation and Inference, ser. Lecture Notes in Computer Science, D. Barker-Plummer, R. Cox, and N. Swoboda, Eds. Springer Berlin Heidelberg, 2006, vol. 4045, pp. 93–97.

30. H. Edelsbrunner and J. Harer. Persistent Homology – a survey. Contemp. Math., 453:257-282, 2008.

31. H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. Discrete Comput. Geom., 30 (87-107), 2003.

32. H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In IEEE FOCS, Washington, DC, 2000.

33. J. H. Friedman. Multivariate adaptive regression splines, Ann. Statist., vol. 19, no. 1, pp. 1–67, 03 1991.

34. S. Gerber, P. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. IEEE TVCG, 16(6), Nov 2010.

35. S. Gerber, O. Rübel, P. Bremer, V. Pascucci, and R. Whitaker. Morse-Smale Regression. Manuscript, 2011.

36. Z. Guo, M. Ward, E. Rundensteiner, and C. Ruiz. Pointwise local pattern exploration for sensitivity analysis, in IEEE Conference on Visual Analytics Science and Technology, 2011, pp. 131–140.

37. A. Gyulassy, D. Günther, J. Levine, J. Tierny, and V. Pascucci. Conforming Morse-Smale complexes. IEEE Transactions on Visualization and Computer Graphics, 20 (12) : 2595-2603. Dec 2014.

38. W. Hackbusch, Multi-grid methods and applications, Springer series in computational mathematics. Springer-Verlag, 1985.

39. J. Hales, S. Novascone, G. Pastore, D. Perez, B. Spencer, and R. Williamson. BISON Theory Manual, 2013.

40. W. Harvey, O. Rübel, V. Pascucci, P. Bremer, and Y. Wang. Enhanced Topology-Sensitive Clustering by Reeb Graph Shattering. In Topological Methods in Data Analysis and Visualization II: Theory, Algorithms, and Applications, 77-90. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

41. K. Holtzblatt and S. Jones. Contextual inquiry: a participatory technique for system design. Lawrence Erlbaum Assoc., Hillsdale, 1993.

42. T. Ikonen and V. Tulkki. The importance of input interactions in the uncertainty and sensitivity analysis of nuclear fuel behavior. Nucl. Eng. Des., 275(0):229-241, 2014.

43. G. Johansson and H. Carr. Accelerating marching cubes with graphics hardware. In Proceedings of the 2006 Conference of the Center for Advanced Studies on Collaborative Research, CASCON ’06. Riverton, NJ, USA: IBM Corp., 2006. [Online]. Available: http://dx.doi.org/10.1145/1188966.1189018

44. K.-C. Li, H.-H. Lue, and C.-H. Chen. Interactive tree-structured regression via principal hessian directions. Journal of the American Statistical Association, vol. 95, no. 450, pp. 547–560, 2000.

45. Y. Livnat, H.-W. Shen, and C. R. Johnson. A near optimal isosurface extraction algorithm using the span space. IEEE Transactions on Visualization and Computer Graphics, vol. 2, no. 1, pp. 73–84, Mar. 1996. [Online]. Available: http://dx.doi.org/10.1109/2945.489388

46. W. E. Lorensen and H. E. Cline. Marching cubes: A high-resolution 3d surface construction algorithm. SIGGRAPH Comput. Graph., vol. 21, no. 4, pp. 163–169, Aug. 1987.

47. P. Mackerras. A fast parallel marching-cubes implementation on the fujitsu ap1000. Tech. Rep., 1992.

48. D. Maljovec, S. Liu, B. Wang, D. Mandelli, P. Bremer, V. Pascucci, and C. Smith. Analyzing simulation-based PRA data through traditional and topological clustering: A BWR station blackout case study. RESS, 2015.

49. D. Maljovec, B. Wang, D. Mandelli, P. Bremer, and V. Pascucci. Analyzing dynamic probabilistic risk assessment data through clustering. PSA, 2013.

50. D. Maljovec, B. Wang, V. Pascucci, P. Bremer, M. Pernice, D. Mandelli, and R. Nourgaliev. Exploration of high-dimensional scalar functions for nuclear reactor safety analysis and visualization. M&C, 2013.

51. D. Maljovec, B. Wang, P. Rosen, A. Alfonsi, G. Pastore, C. Rabiti, and V. Pascucci, Rethinking sensitivity analysis of nuclear simulations with topology, in Proceedings of IEEE Pacific Visualization (PacificVis), 2016.

52. T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer. Guiding feature subset selection with an interactive visualization. in IEEE VAST, 2011, pp. 111–120.

53. S. Miguet and J.-M. Nicod. Complexity analysis of a parallel implementation of the marching-cubes algorithm. International Journal of Pattern Recognition and Artificial Intelligence, vol. 11, no. 07, pp. 1141–1156, 1997. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218001497000536

54. T. Muhlbacher and H. Piringer. A partition-based framework for building and validating regression models. IEEE TVCG, vol. 19, no. 12, pp. 1962–1971, 2013.

55. M. Nason, S. Emerson, and M. Leblanc. Cartscans: A tool for visualizing complex models, Journal of Computational and Graphical Statistics, vol. 13, no. 4, pp. 807–825, 2004.

56. P. Oesterling, C. Heine, G. Weber, D. Morozov, and G. Scheuermann, "Computing and visualizing time-varying merge trees for high-dimensional data," May 2015, workshop on Topology-Based Methods in Visualization (TopoInVis).

57. V. Pascucci. Isosurface computation made simple: Hardware acceleration, adaptive refinement and tetrahedral stripping. In Proceedings of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization, VISSYM'04. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004, pp. 293–300.

58. V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas, Robust on-line computation of Reeb graphs: Simplicity and speed, ACM Trans. Graph., vol. 26, no. 3, Jul. 2007.

59. G. Pastore, L. Swiler, J. Hales, S. Novascone, D. Perez, B. Spencer, L. Luzzi, P. Uffelen, and R. Williamson. Uncertainty & sensitivity analysis of fission gas behavior in engineering-scale fuel modeling. J. Nucl. Mater., 456:398-408, 2015.

60. H. Piringer, W. Berger, and J. Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation, in Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization, ser. EuroVis'10. Eurographics Association, 2010, pp. 983–992.

61. C. Rabiti, D. Mandelli, A. Alfonsi, J. Cogliati, and R. Kinoshita. Introduction of supervised learning

capabilities of the RAVEN code for limit surface analysis. In Proceeding of American Nuclear Society (ANS), Reno (NV), 2014.

62. J. Roberts. State of the art: Coordinated multiple views in exploratory visualization. In CMV 2007, pages 61-71, July 2007.

63. M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: reflections from the trenches and the stacks. IEEE TVCG, 18(12): 2431-2440, 2012.

64. A. J. Smola and B. Scholkopf, A tutorial on support vector regression, Statistics and Computing, vol. 14, no. 3, pp. 199–222, 2004.

65. H. Tavakol-Davani. Watershed-scale life cycle assessment of rainwater harvesting systems to control combined sewer overflows, Ph.D. dissertation, University of Utah, Salt Lake City, UT, 2016.

66. J. Tierny and V. Pascucci. Generalized topological simplification of scalar fields on surfaces. IEEE Transactions on Visualization and Computer Graphics (TVCG), 18(12): 2005-2013, Dec 2012.

67. J. Tierny, D. Günther, and V. Pascucci. Optimal General Simplification of Scalar Fields on Surfaces. Topological and Statistical Methods for Complex Data: Tackling Large-Scale, High-Dimensional, and Multivariate Data Spaces, 57-71. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

68. P. Tol. Colour Schemes. Technical Note SRON/EPS/TN/09-002, SRON Netherlands Institute for Space Research, Dec 2012.

69. S. Urbanek. Different ways to see a tree-klimt, in Compstat. Springer, 2002, pp. 303–308.

70. M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore, “Contour trees and small seed sets for isosurface traversal,” in Proceedings of the Thirteenth Annual Symposium on Computational Geometry, ser. SCG ’97. New York, NY, USA: ACM, 1997, pp. 212–220. [Online]. Available: http://doi.acm.org/10.1145/262839.269238

71. R. W. Youngblood, V. A. Mousseau, D. L. Kelly, and T. N. Dinh. Risk-Informed Safety Margin Characterization (RISMC): Integrated Treatment of Aleatory and Epistemic Uncertainty in Safety Analysis, Oct 2010. [Online]. Available: http://www.osti.gov/scitech/servlets/purl/99318