Overview

RAVEN workshop

C. Rabiti, A. Alfonsi, S. Sen, D. Maljovec, P. Talbot, R. Kinoshita, D. Mandelli, J. Cogliati

PSA 2015 - April 26th 2015, Sun Valley (ID)

RAVEN

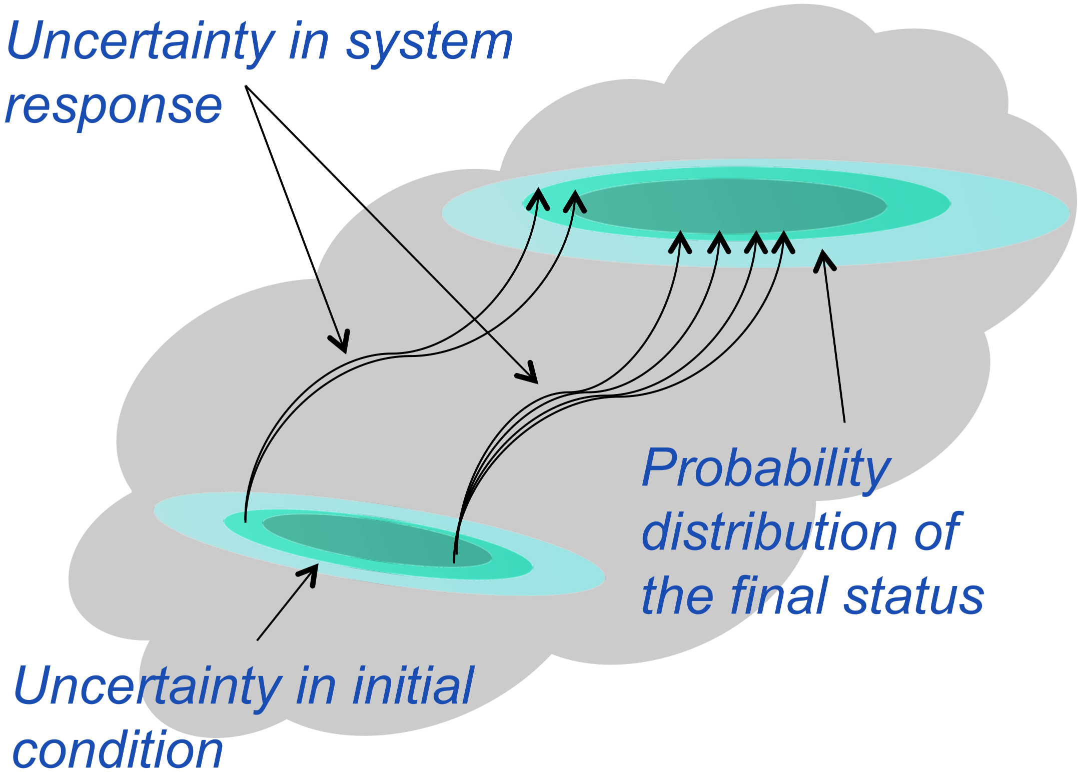
Risk Analysis Virtual ENvironment

Project Background

- RAVEN was started in early 2012
  - Supported by the Nuclear Energy Advanced Simulation (NEAMS) program
  - The Light Water Reactor Sustainability (LWRS) program is the major customer and supporter

- The overall goal was to have a tool to enable Risk Informed Safety Margin Characterization (RISMC)
  - Evaluating risk (uncertainty propagation)
  - Understanding risk (limit surface, ranking, sensitivity, data mining)
  - Mitigating risk (optimization)

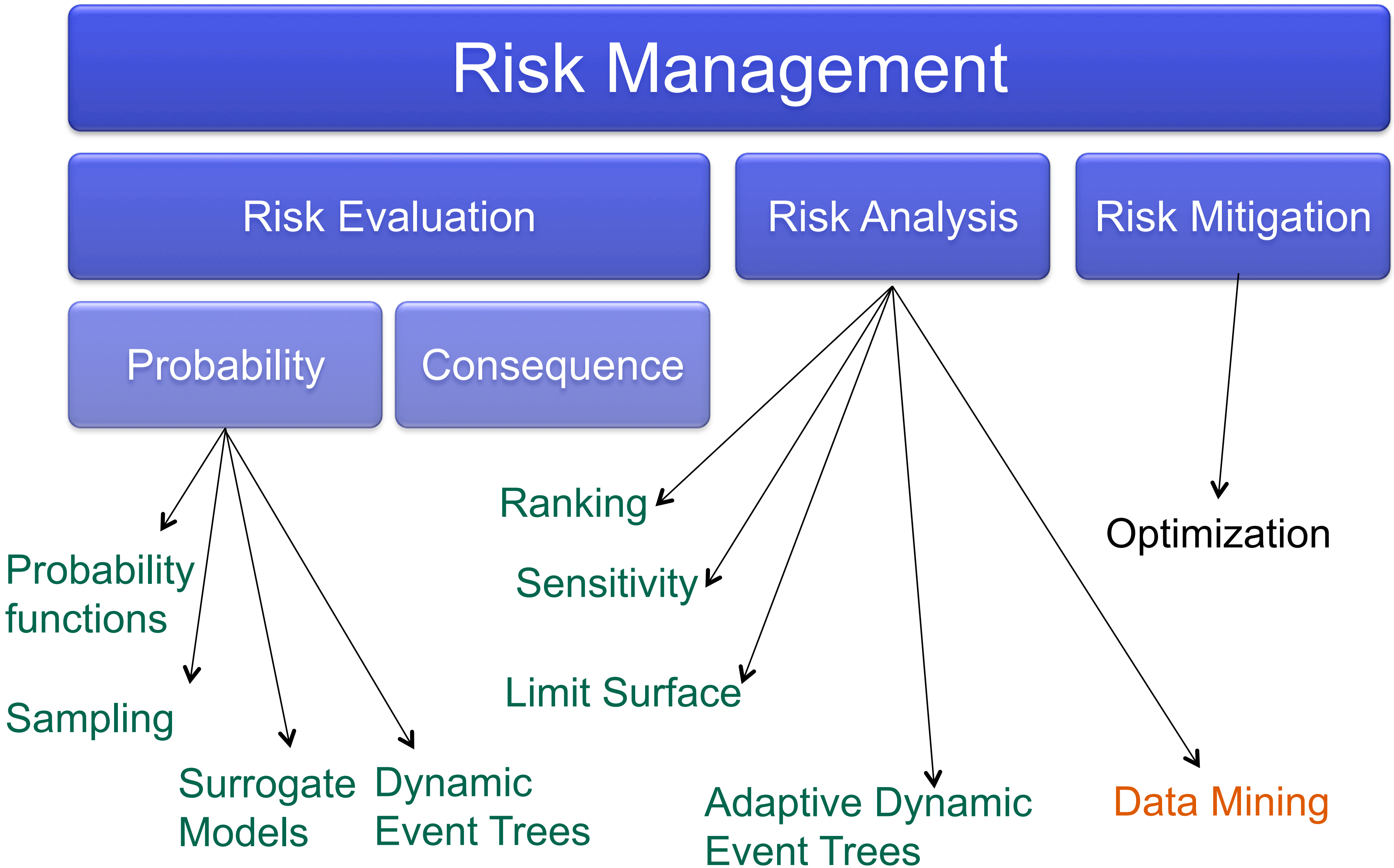
For Which Type of Systems



- Uncertainty in initial and boundary condition
- Uncertainty in model parameters
- Stochastic events
- Possibly dealing with discrete variable
- Highly non linear systems

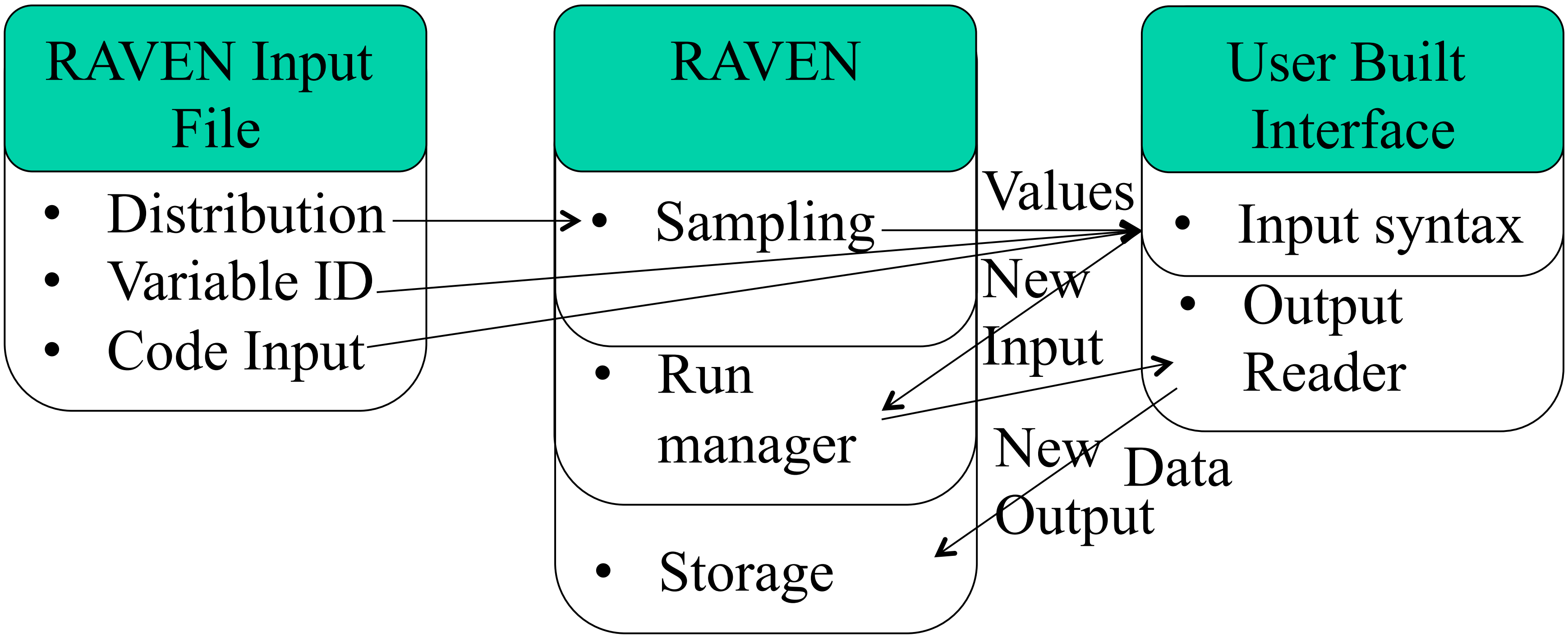
Idaho National Laboratory

Which Tools



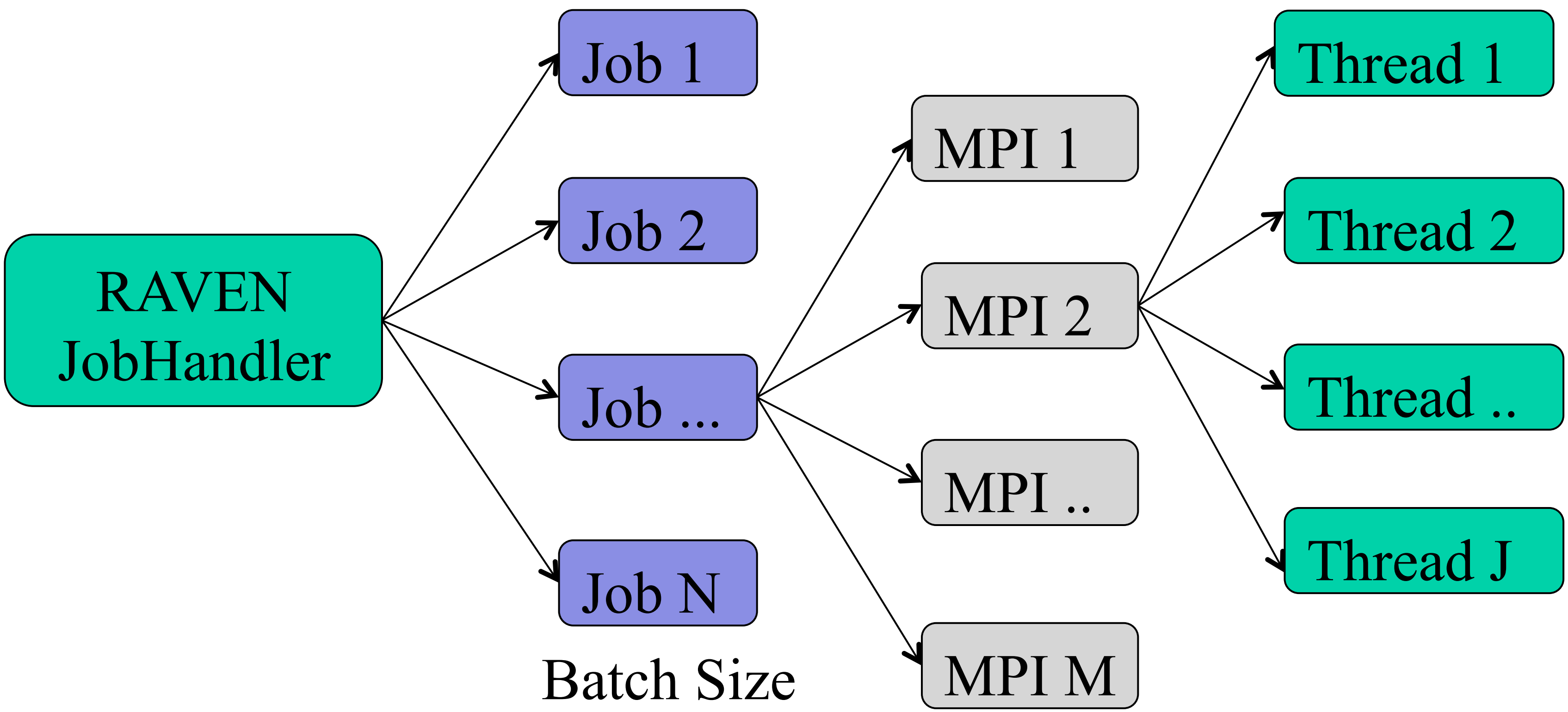
How

- RAVEN could interact with a physical model by:
  - Wrapping the code simulating the physical moment
  - Exchanging information at each time step (only for RELAP-7 and MOOSE based application soon)
  - Leveraging already existing control logic structure in the code (e.g. RELAP5-3D for Dynamic Event Trees)



Parallel Environment

- Sampling is a very expensive computational operation
- Modern codes are already parallel using MPI and multithreading
- RAVEN manages concurrent runs of already parallel codes

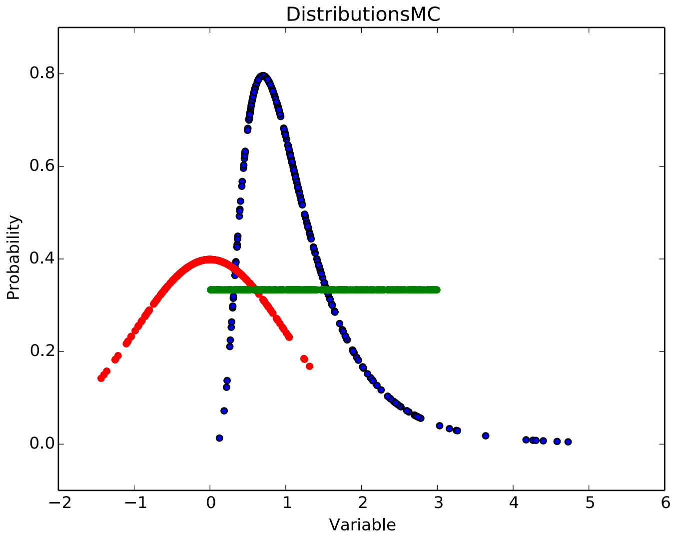


The Building Blocks

- RAVEN is organized in a modular fashion where "**entities**" can be used to construct the sought workflow
- The building blocks are:
  - **Distributions**: describe the probabilistic behavior of variables
  - **Samplers**: associate the distribution with a variable and a sampling strategy
  - **Models**: represent the system to be explored or more generally an input output relationship
- The actions (**Steps**):
  - **MultiRun/Run**: feeds an input to a model to retrieve an output, if a sampler is present the model is sampled according
  - **Postprocessor**: investigate outputs to determine their statistical characterization and relate the output to the input
  - **IO**: export import data where the output are stored

Probability Distribution Functions (1D)

- Most common used 1D distributions (Boost)



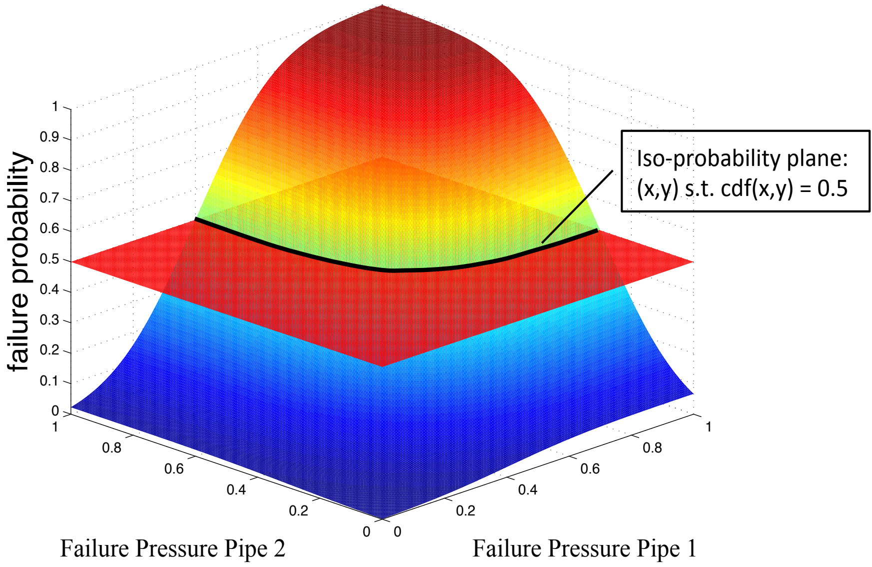
Probability Distribution Function	Truncated Form Available
Bernoulli	No
Binomial	No
Exponential	Yes
Logistic	Yes
Lognormal	Yes
Normal	Yes
Poisson	No
Triangular	Yes
Uniform	Yes
Weibull	Yes
Gamma	Yes
Beta	Yes

Probability Distribution Functions (ND)

- Multivariate distributions appear in many practical problems
- They are challenging from two points of view:
  - Construction and evaluation
  - Sampling (they are not invertible)
- RAVEN manages
  - Multivariate normal:

$$\frac{1}{\sqrt{(2\pi)^k\|\boldsymbol{\Sigma}\|}}e^{\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)}$$

  - Custom, user-provided distributions (supplied as a file of evaluated points)

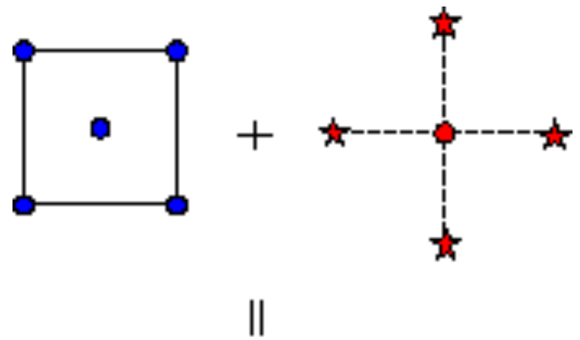


Samplers (non-Adaptive)

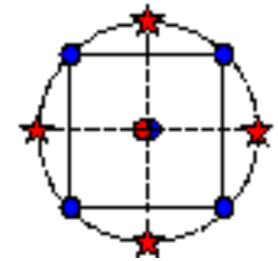
- Choose how to sample the input (parametric) space:
  - Impacts how effectively the input space is explored
  - Determines the probability associated with each realization of the output
- RAVEN supports many non-adaptive samplers
  - Monte Carlo
  - Grid
    - Equal-spaced in probability
    - Equal-spaced in value
    - Mixed
    - Custom
  - Stratified (LHS type)
    - Equal-spaced in probability
    - Equal-spaced in value
    - Mixed
    - Custom

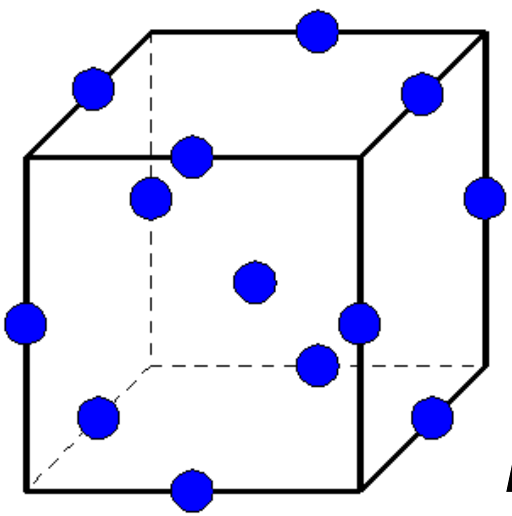
Samplers (non-Adaptive) Cont.

- Factorial Designs:
  - General Full Factorial (grid)
  - 2-Level Fractional-Factorial
  - Plackett-Burman
- Response Surface Designs:
  - Box-Behnken
  - Central Composite



Central Composite

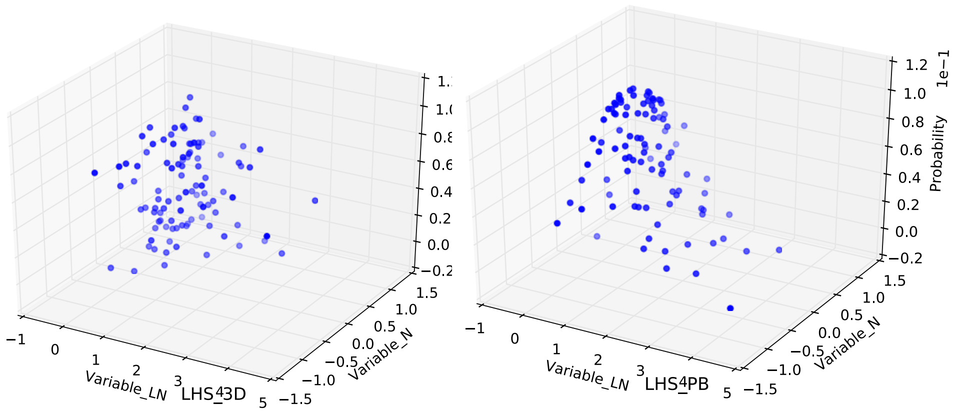




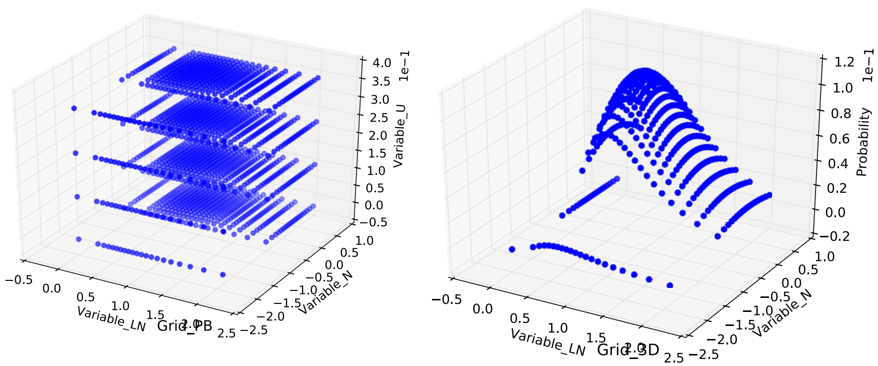
Box-Behnken

High density sampler focused on full coverage of the input space (highly discontinuous and nonlinear response, low dimensionality)

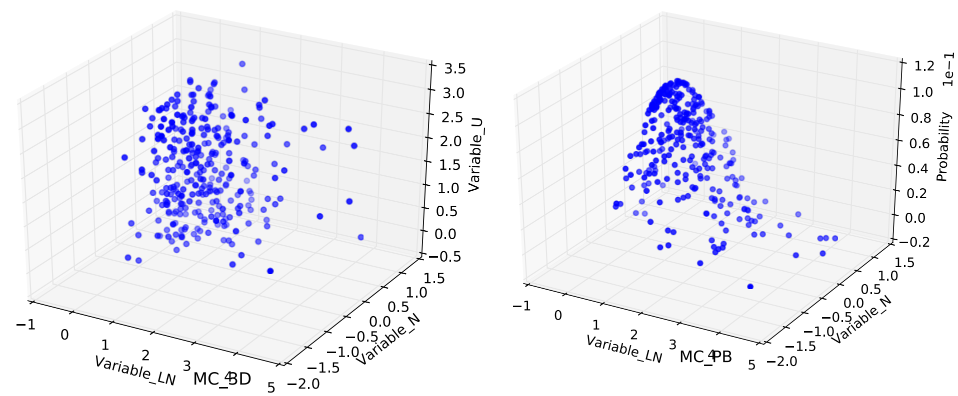
Spatial Distribution Examples



Latin Hypercube (100 points)



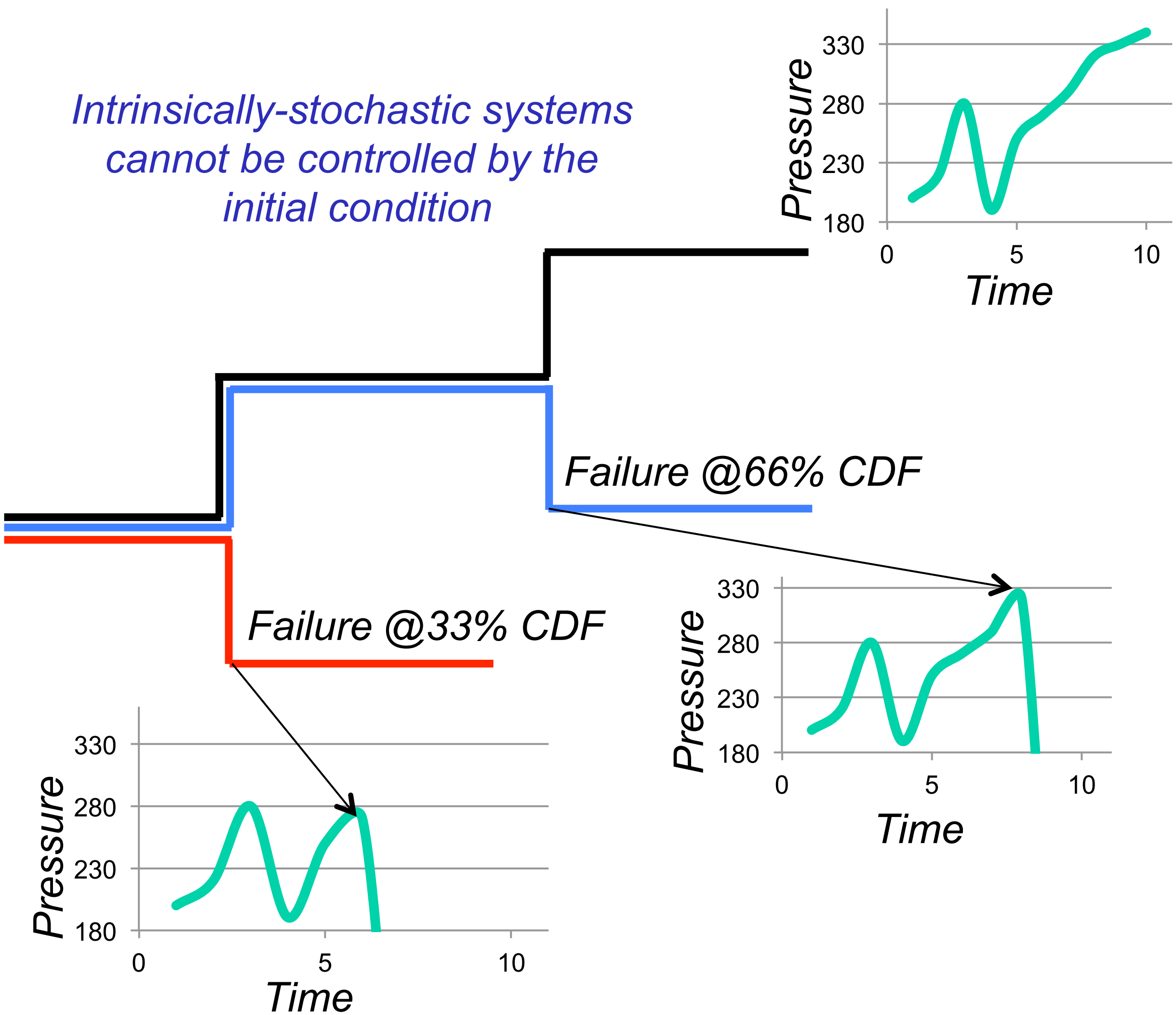
Grid (1764 points)

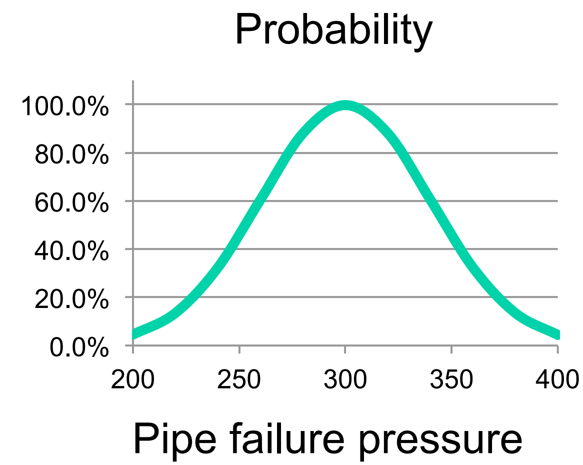


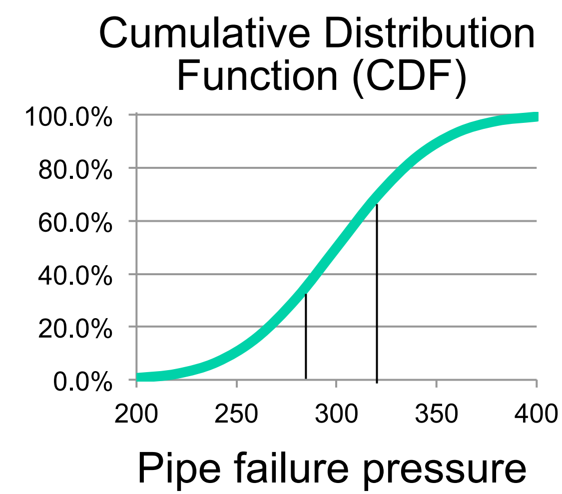
Monte Carlo (300 points)

Dynamic Event Tree

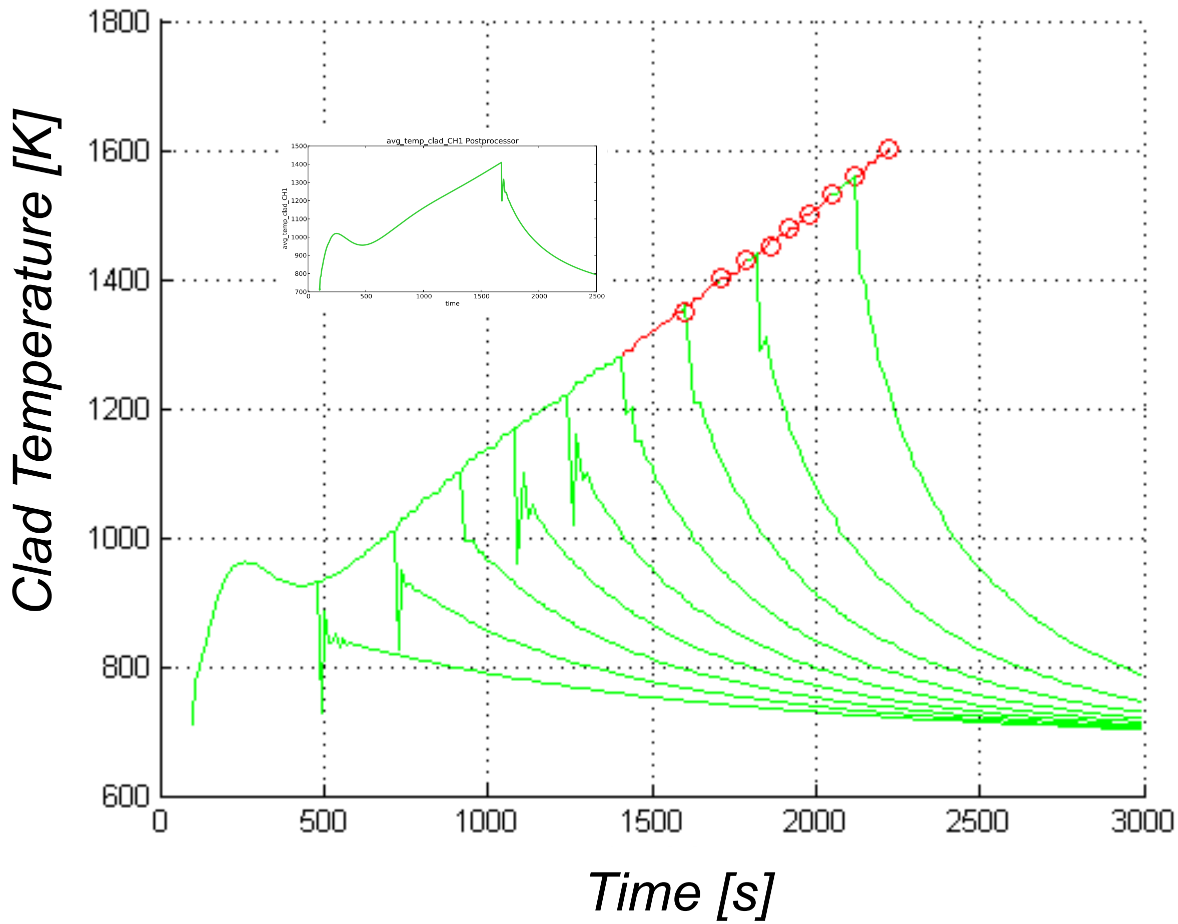
Intrinsically-stochastic systems cannot be controlled by the initial condition







Dynamic Event Tree (DET) Example

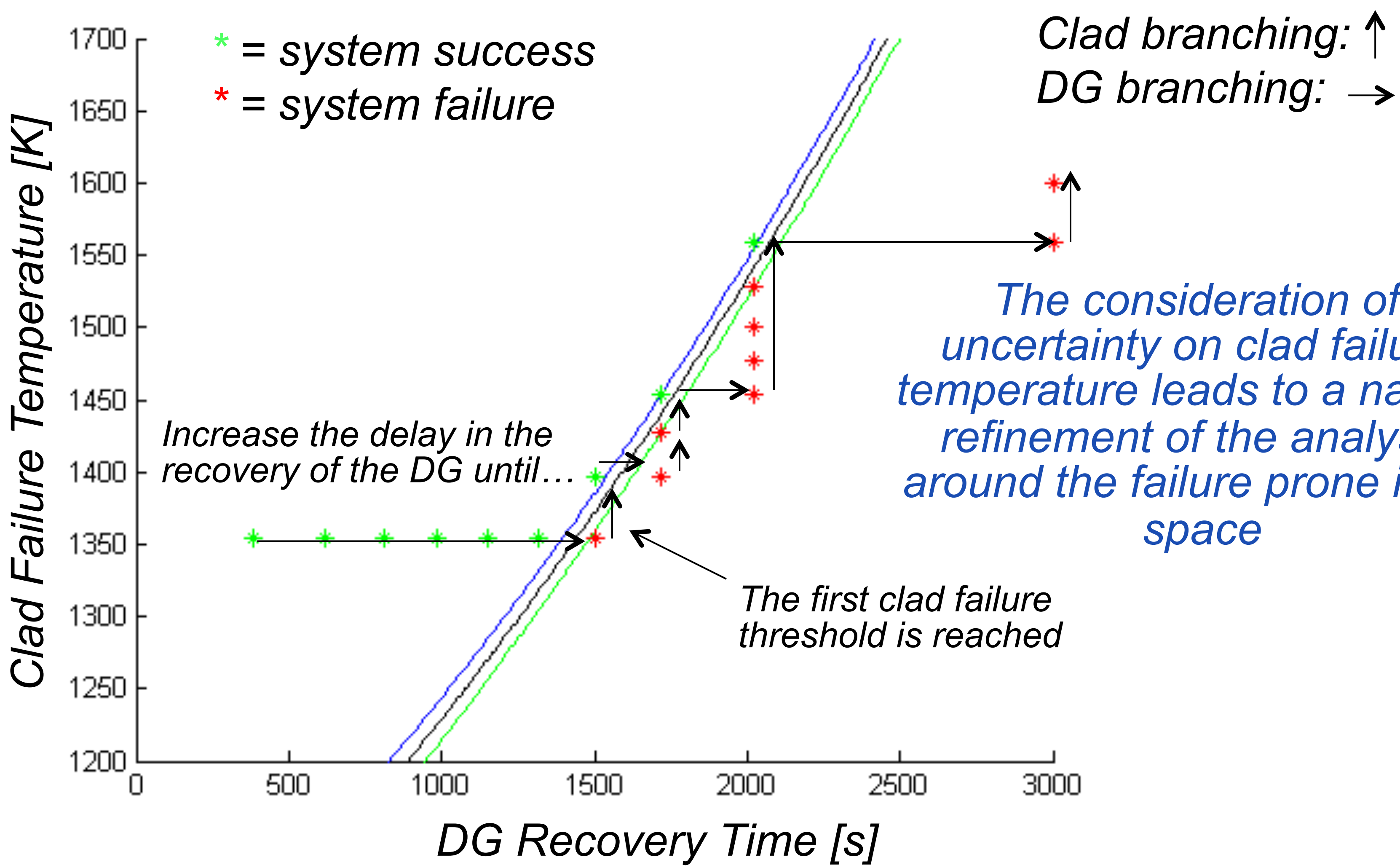


* = system success

* = system failure

All the tree branches are assembled to reproduce the full evolution

DET Follows Failure Patterns



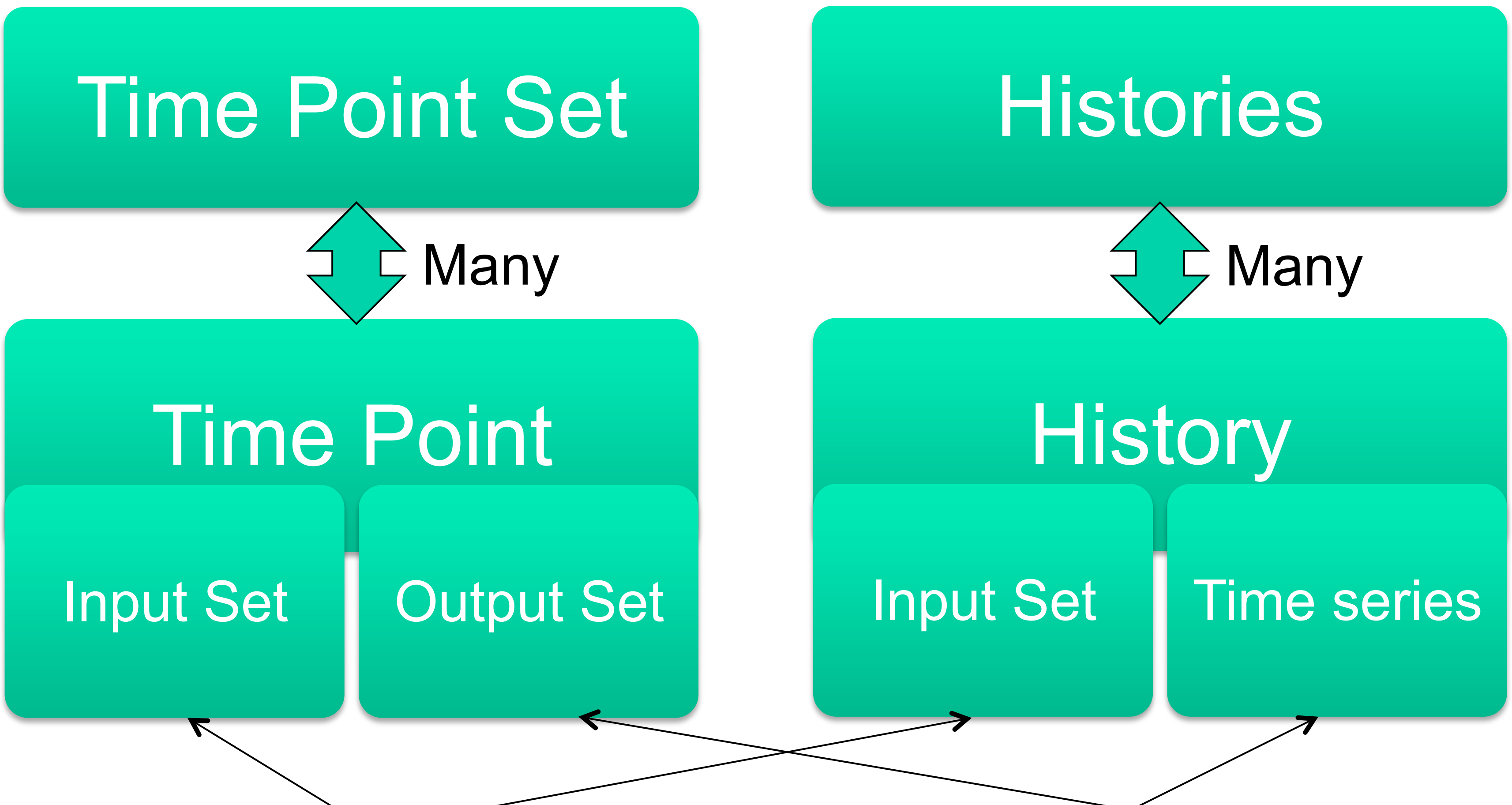
The consideration of uncertainty on clad failure temperature leads to a natural refinement of the analysis around the failure prone input space

Models

- At the software level, models relate input to output
- It might be more convenient to think about a model as software simulating a physical system
- Within this context the following models are available:
  - **Code**: an external code that RAVEN can inquire via properly-built interfaces to manage input and output
  - **External model**: a numerical implementation of a physical model that is interfaced to RAVEN directly (no file exchange) using RAVEN's API
  - An internally generated surrogate models (**ROM**)

Data

- RAVEN posses four main data type



For a set of the input parameters ... the corresponding figures of merit

Statistical Post Processing

Statistical characterization of the output (***uncertainty propagation***)

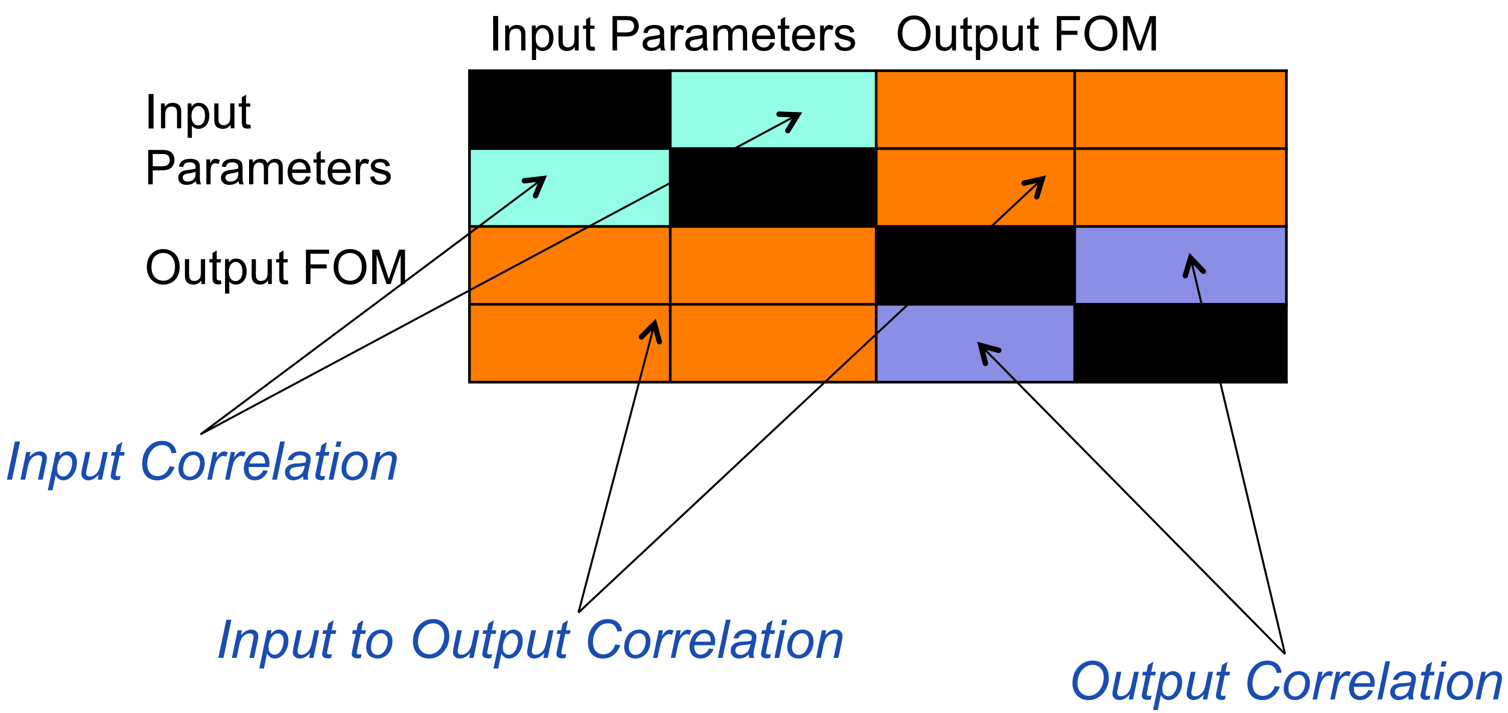
- Mean
- Sigma
- Skewness (asymmetry)
- Kurtosis (more/less peaked than a standard normal)

Input/output relationship (***ranking/sensitivity***)

- Correlation matrix
- Covariance matrix
- Sensitivity matrix (multidimensional linear regression)
- Normalized sensitivity matrix (% change of the response / % change in the answer)

Input/Output Relationship

- Correlation matrix
- Covariance matrix
- Sensitivity matrix
- Normalized sensitivity matrix



Advanced Post Processing

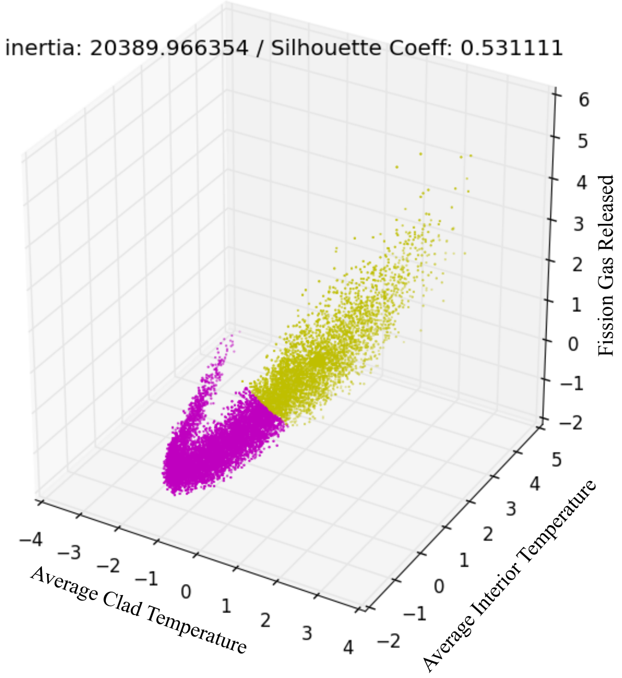
- RAVEN gives access to several data mining algorithms for data post-processing via implementation of APIs between RAVEN and Scikit-learn*
- Cardinality Reduction algorithms aim to recognize commonalities among the data - Clustering, Gaussian Mixture Models, etc.
- Dimensionality Reduction algorithms are used to reduce the number of degrees of freedom in the data under consideration – Manifold Learning, Decomposing Signals in Components, etc.
- In principle data mining makes no distinction between input and output

*http://scikit-learn.org

Application to BISON Fuel Analysis

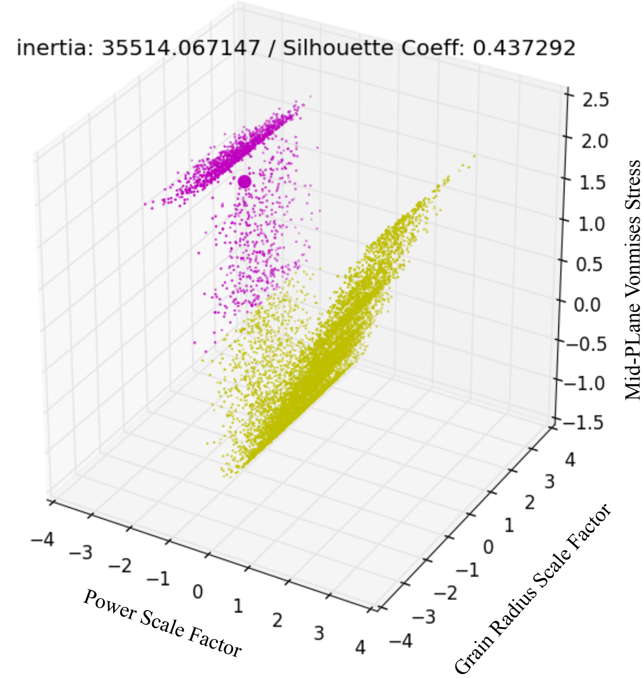
- ~17,000 simulations, 3 input parameters

power_scalef	Power scaling factor of the core
grainradius_rscalef	Scaling factor of the grain size in the fuel microstructure
thermal_expansion	Thermal expansion coefficient of the fuel



Output Space

- K-Means algorithm (Lloyd's algorithm)
- No. Clusters: 2



Input - Output Space

Simple Workflow

- Definition of the distributions to be used
- Definition of the sampler
  - Association of variable name(s) with distribution(s)
  - Choice of a sampling strategy
- Definition of the data structure(s) for internal storage
- Evaluation step
  - The distributions are sampled according to the sampler
  - The variables are associated to the values
  - The model is executed as many times as the sampler requires
  - The inputs and corresponding outputs are stored in the data structure(s)
- Post-processing
  - The data is analyzed according to requested type of post-processing
- I/O
  - Data is saved for further analysis

Reduced Order Models (ROM)/Surrogate Models

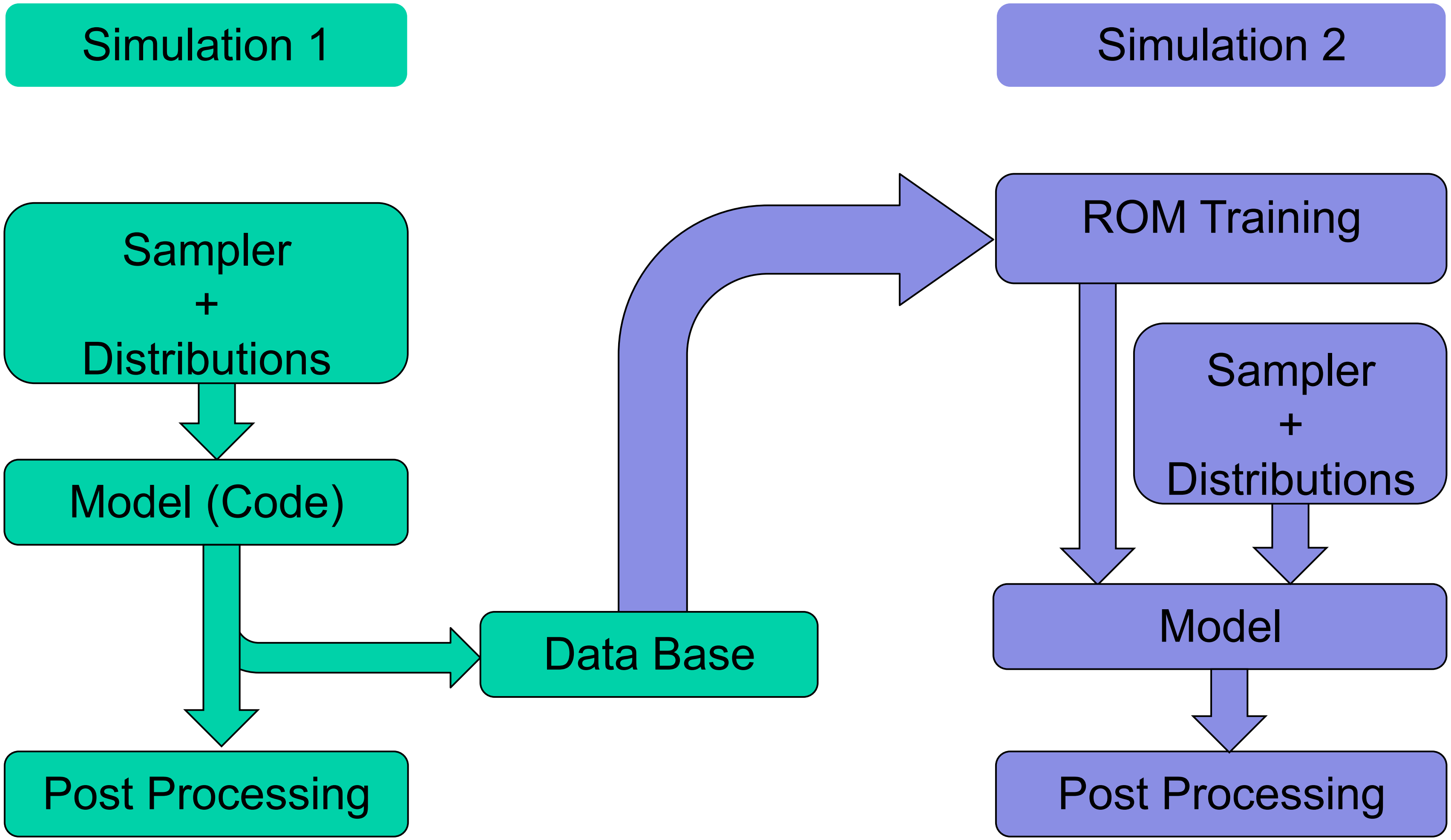
- Not for all possible inputs, but for a range of possibilities…
- Not for all possible geometries, but for a parametric range…
- Not to represent all outputs, but for some of the outputs…

It is possible to:

-Build a set of equations of faster solution with respect to the original set

-Increase the prediction capabilities by further training

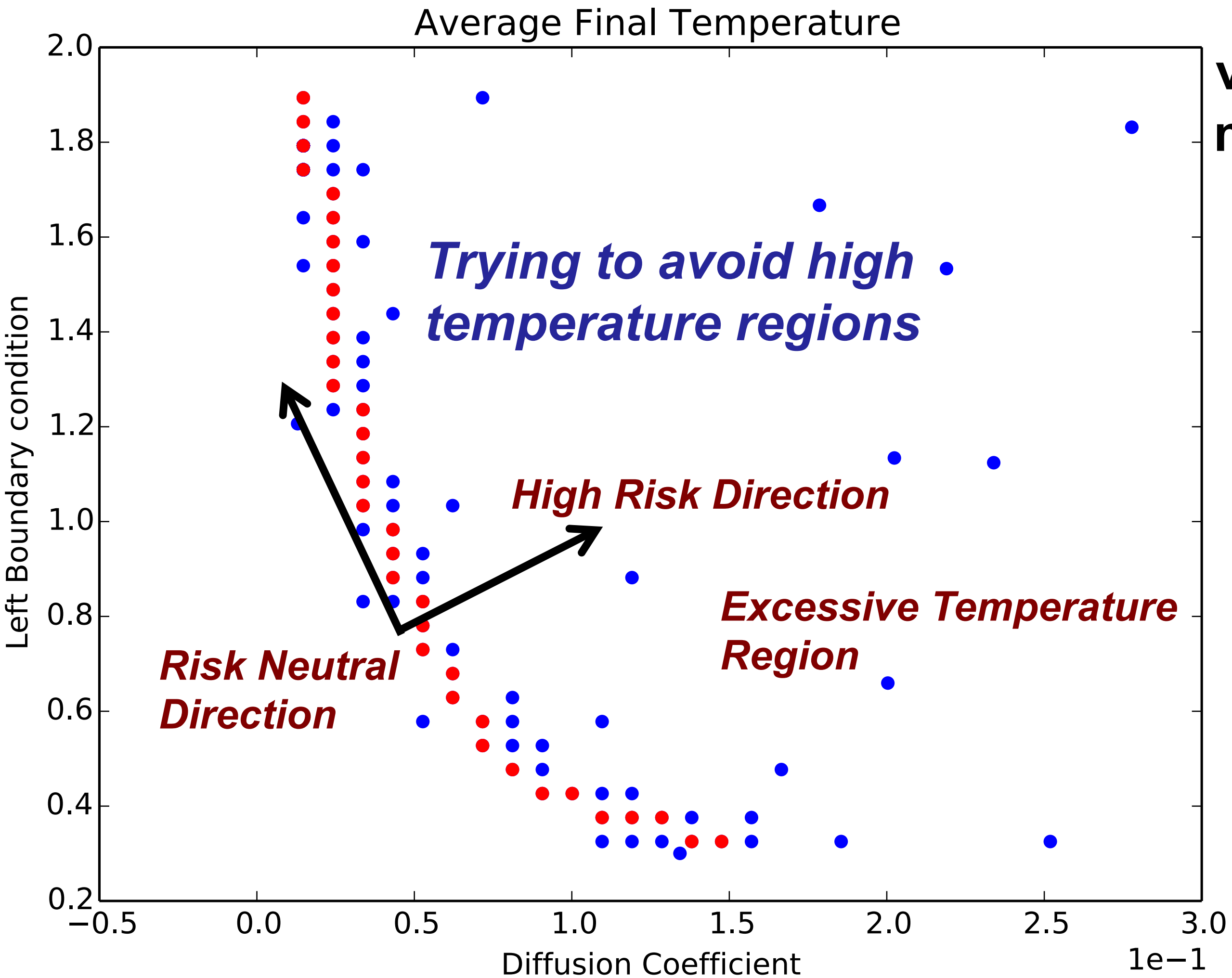
Using Surrogate Models



Available Surrogate Models

- The surrogate models available are:
  - Supervised algorithms available from scikit-learn
    - SVMs
    - Nearest neighbor
    - Gaussian Process
    - Etc…
  - Inverse weight
  - ND spline (only regular Cartesian grids)
  - Generalized Polynomial Chaos Expansion
- RAVEN has an API to implement your own surrogate model

Reliability Analysis (Limit Surface)



System response depends on many variables but often what matters are a few figures of merit

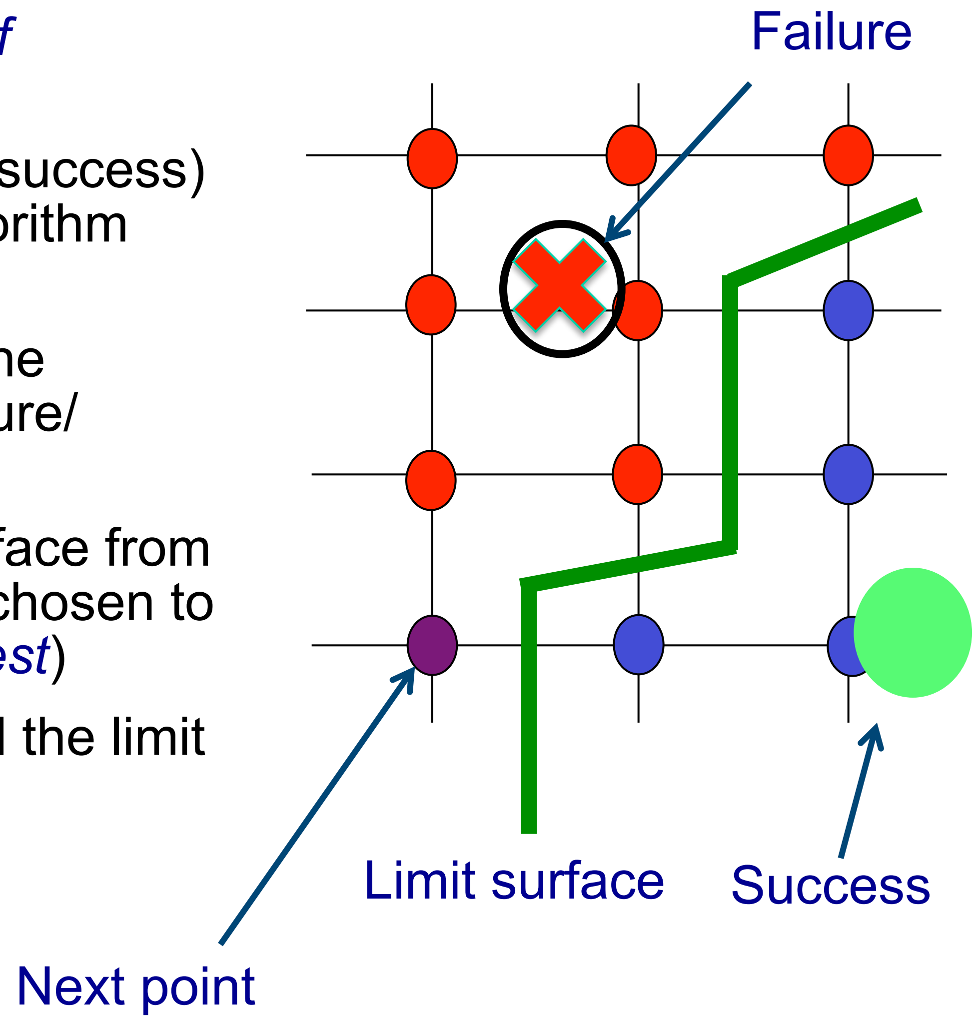
What is the probability of exceeding the threshold?

Which parameter is the most influential?

Which uncertainties affect my margins the most?

The Limit Surface: Simulation Flow

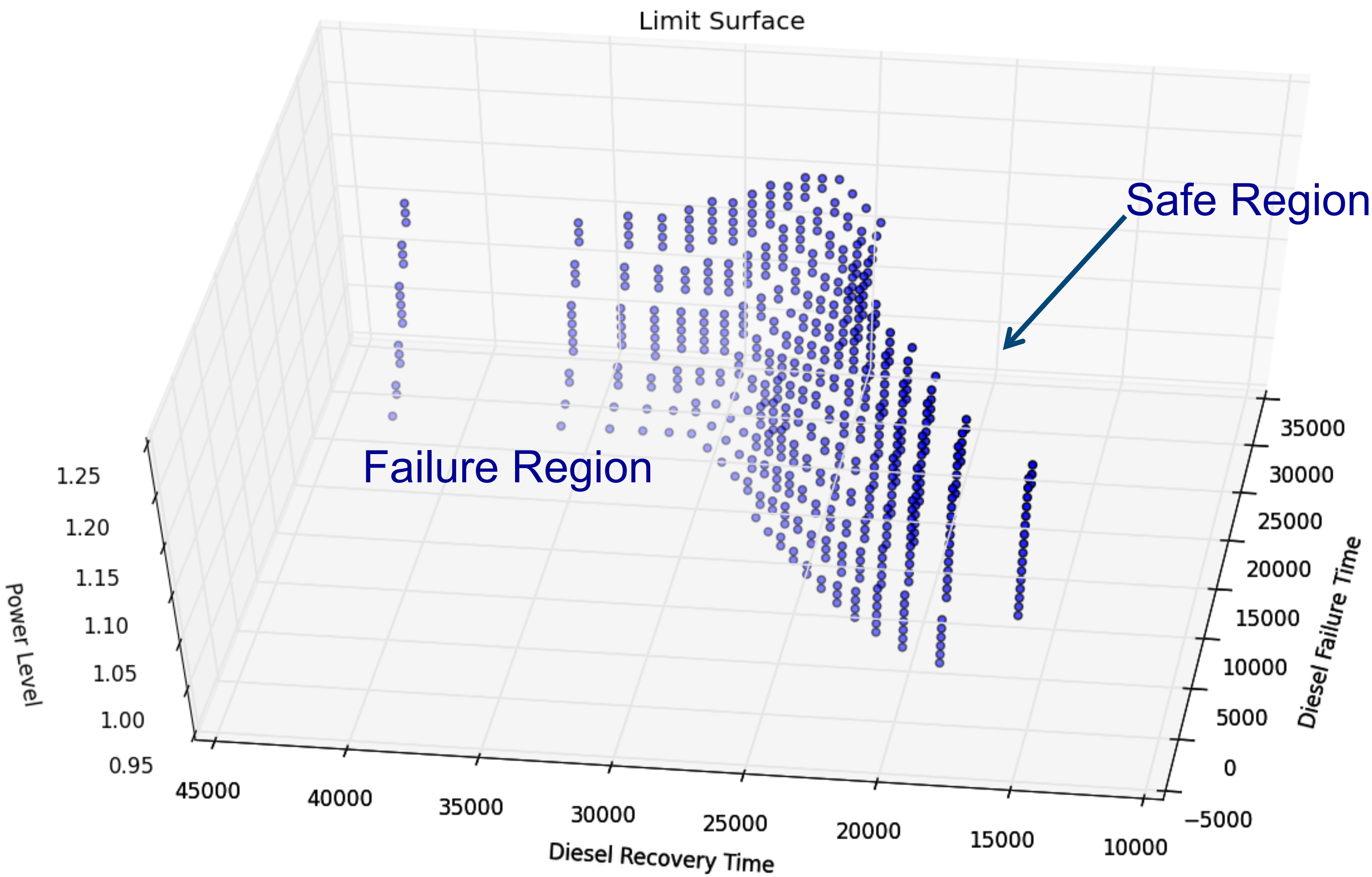
1. Training on initial *small number of sampled points*
2. A fine grid is classified (failure or success) based on a nearest-neighbor algorithm (*sampling of the surrogate*)
3. The limit surface is identified by the location of transition between failure/ success
4. The furthest point on the limit surface from any other already tested point is chosen to test the classifier (*convergence test*)
5. Process repeats from point 2 until the limit surface converges



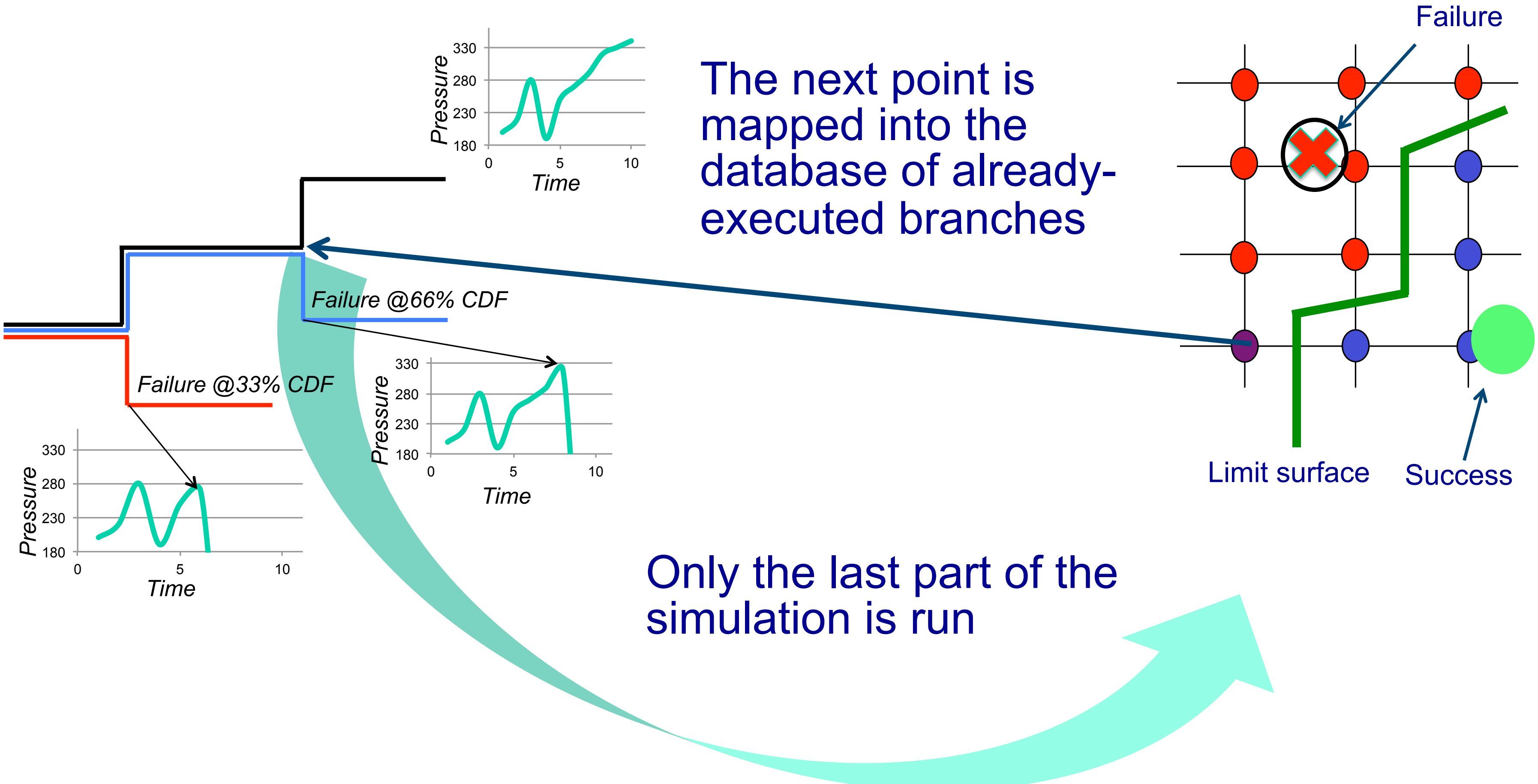
Limit Surface Workflow

- Define the distributions. Distributions are used to:
  - Weigh the error estimation
  - Generate a probability-weighted metric for grid construction
  - Determine the probability of being inside one of the regions identified by the limit surface
- Define the surrogate model to be used as the accelerator (of type Boolean or discrete)
- Define the adaptive sampler:
  - Convergence control and grid
  - Import the acceleration ROM
  - Associate the distribution with the variables
- Define a step
- Post-process the data to compute the failure probability

Limit Surface Example



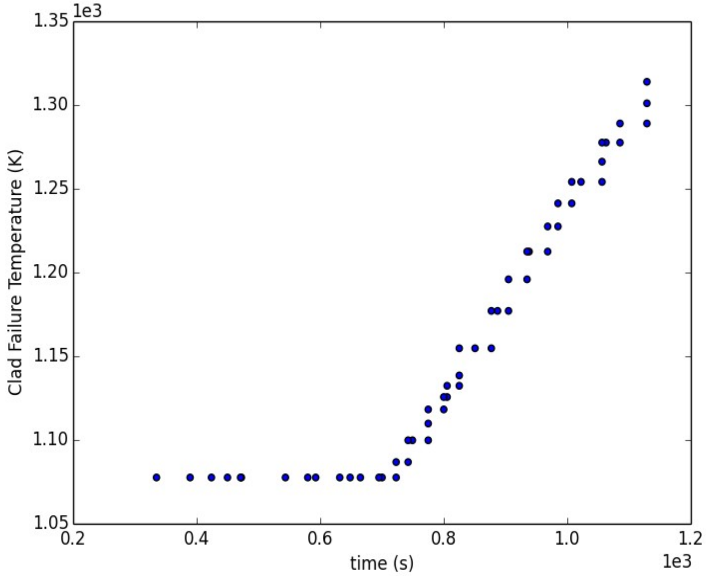
Adaptive Dynamic Event Tree (A-DET)



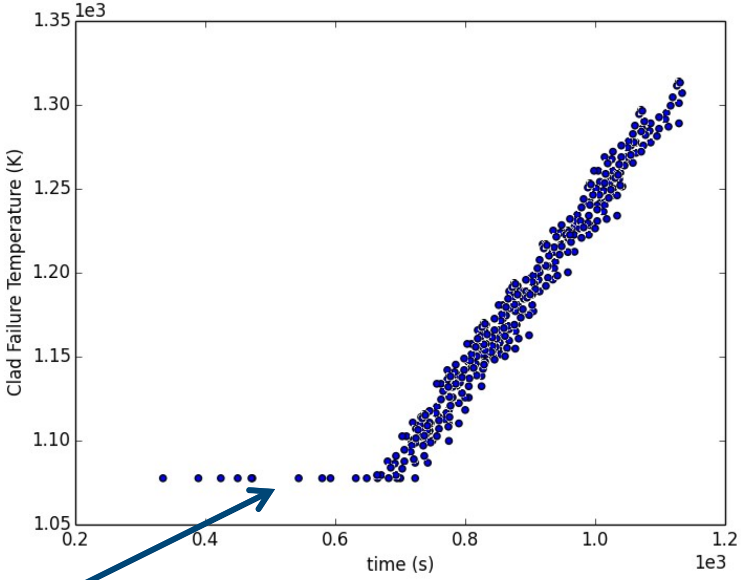
The DE's intrinsic characteristic of following failure patterns provides a more effective training set than other random sampling approaches

Dynamic Event Tree Demo

Training



Convergence



No density increase

Databases and IO

- RAVEN posses the capability to dump the internal data in:
  - HDF5 databases
  - CSV files (raw output plus metadata)

- Both HDF5 and CSV could be used to recreate the data set during sub sequential simulations

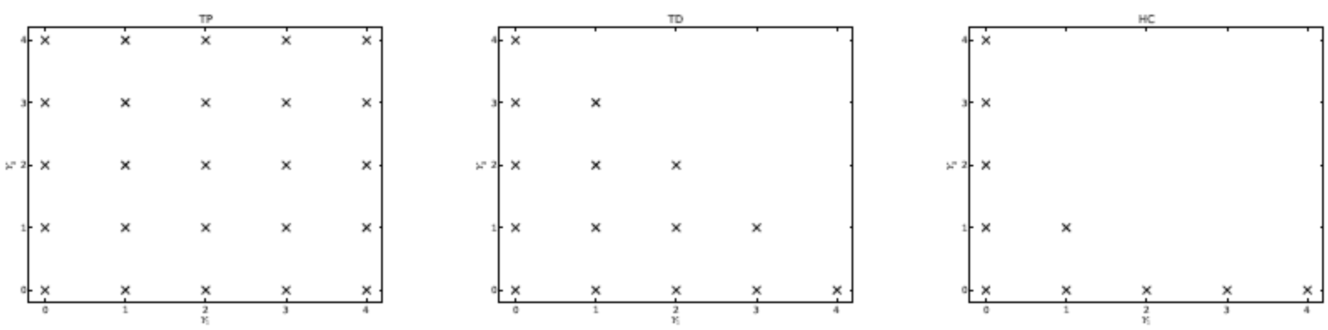
Stochastic Polynomials

- It is a special type of surrogate model
- It requires a specific sampling strategy
- Some of the information of statistical relevance are analytical from the coefficients of the surrogate model
- Polynomial Representation
  - Given a quantity of interest: $u(Y)$
  - Represent as combination of polynomials: $\phi_k(Y)$
    - Simpler to evaluate
    - Easy to get statistical moments
    - Less effort and more accurate than Monte Carlo

$$u(Y) \approx \sum_{k=0}^{N} u\left(Y^{(k)}\right) \phi_k(Y)$$

Stochastic Collocation

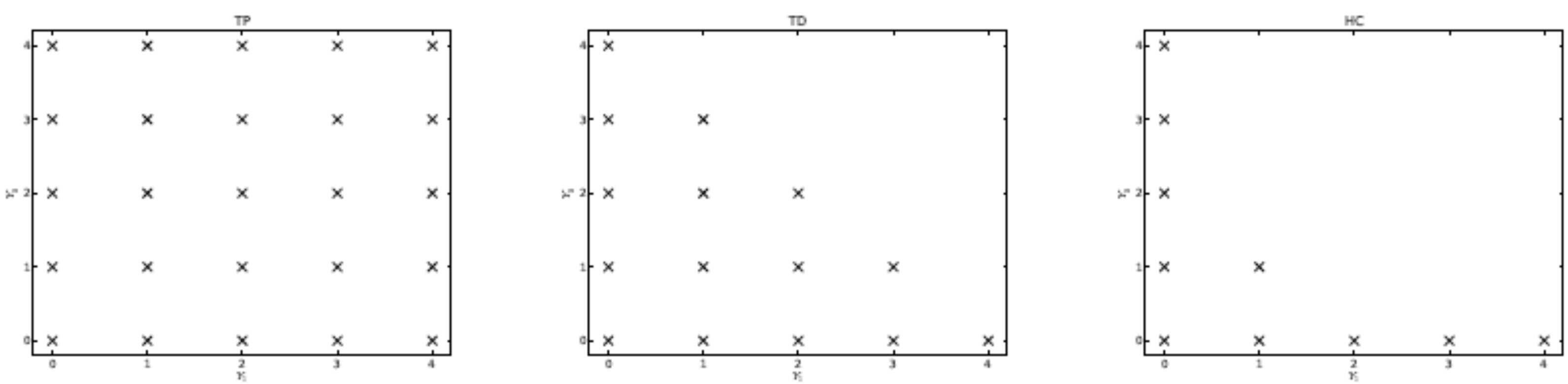
- Effectiveness depends on:
  - Regularity of quantity of interest $u(Y)$
  - Polynomial expansion order L
  - Polynomial combination indices $\Lambda(L)$
  - Sparse Grid quadrature types (Gauss, Clenshaw)
  - Number of uncertain inputs $N = |Y|$
- Tensor Product: Overkill
- Total Degree: For Analytic Uncertainty
- Hyperbolic Cross: For Irregular Uncertainty



(a) Tensor Product (b) Total Degree (c) Hyperbolic Cross

Reduced Polynomial Sets

- Tensor Product
  - Overkill
- Total Degree
  - For Analytic Uncertainty
- Hyperbolic Cross
  - For Irregular Uncertainty



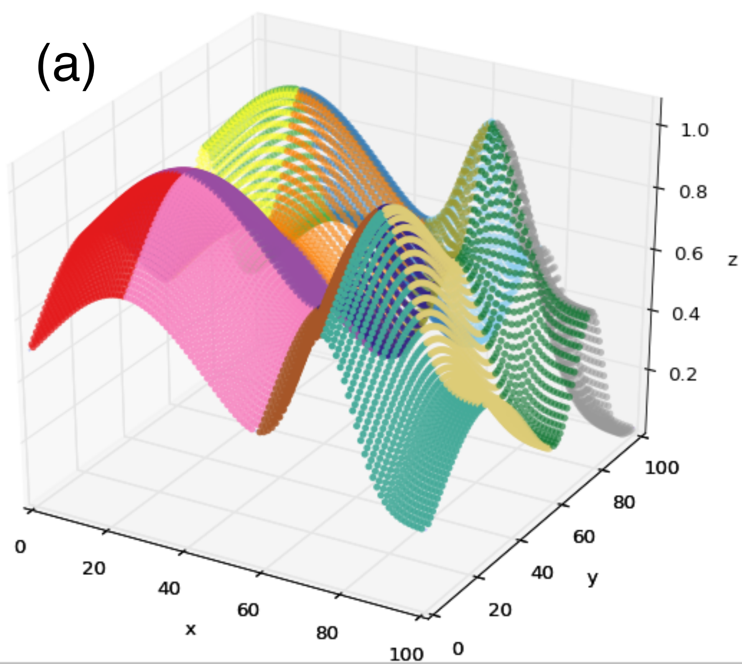
(a) Tensor Product (b) Total Degree (c) Hyperbolic Cross

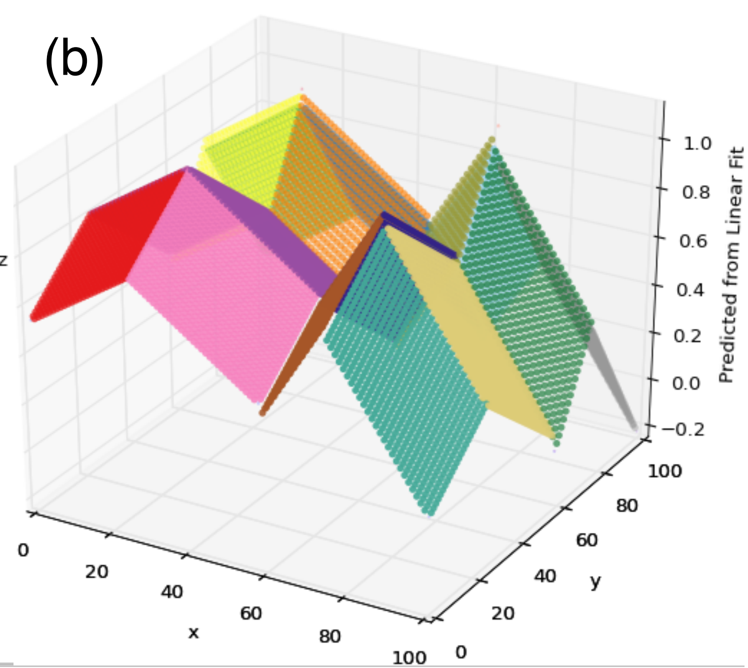
Application of Stochastic Polynomials

Statistic	Mean (x10^8)	Std. Dev.	Failure Pb	Solves	Err Stdev *Runs
BISON MC	5.84360195	4.75237678	0.0986	10000	
J_HC2	5.13E-06	8.91E-03	7.61E-02	7	0.0624
J_HC4	-3.48E-05	9.95E-03	7.10E-02	31	0.3084
J_HC8	-2.35E-05	9.63E-03	7.91E-02	153	1.4736
J_TD2	-2.60E-05	6.41E-01	8.01E-02	25	16.0187
J_TD4	-2.93E-05	9.64E-03	7.61E-02	165	1.5899
J_TD8	-1.92E-05	9.19E-03	7.71E-02	2097	19.2691
L_HC2	1.70E-04	-6.88E-02	-1.38E-01	7	0.4818
L_HC4	3.38E-05	-1.91E-02	1.38E-01	31	0.593
L_HC8	1.71E-05	3.32E-03	8.72E-02	153	0.5087
L_TD2	1.52E-04	-6.90E-02	8.82E-02	25	1.7261
L_TD4	1.53E-05	-1.87E-02	1.20E-01	165	3.0929
L_TD8	-1.28E-05	1.35E-03	7.10E-02	2097	2.8219

Structured Sensitivity Analysis via Topological Decomposition

- (a) Decompose the domain based on gradient flow from/to local minima/maxima
- (b) Fit a linear regressor to each local segment of the data

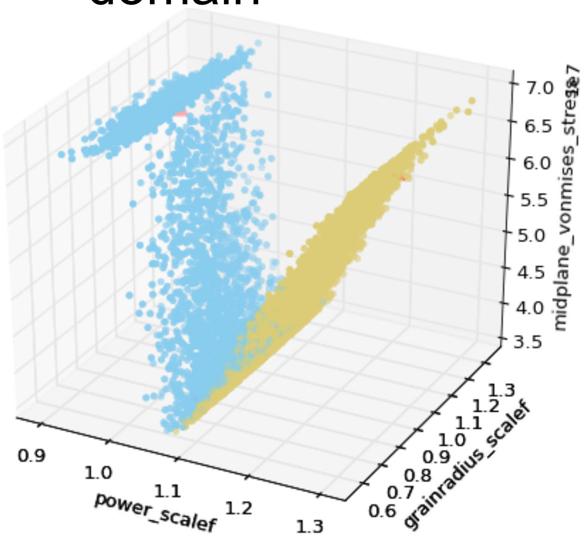


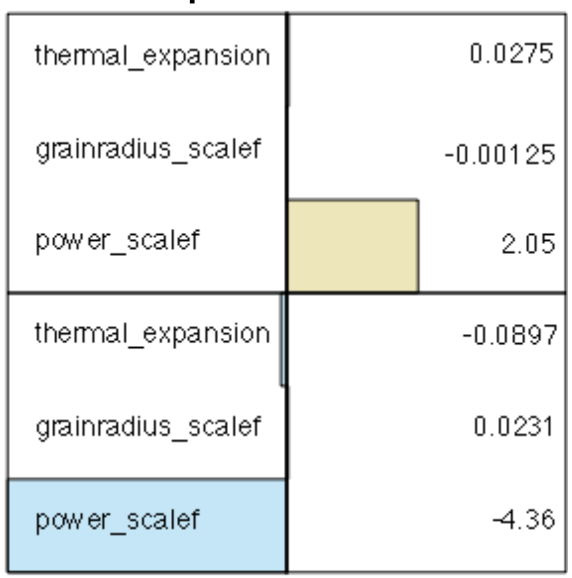


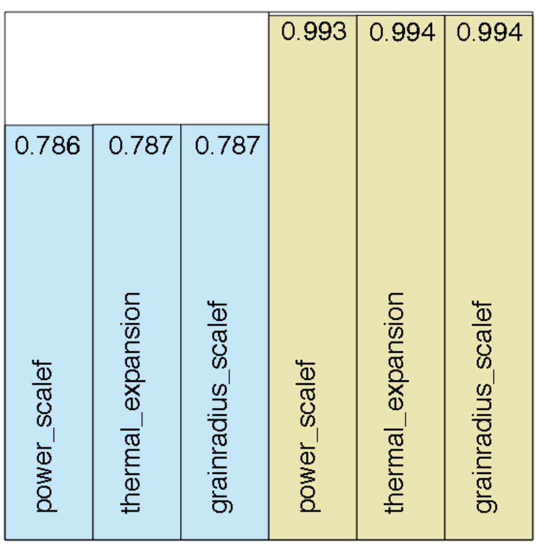
Application to BISON Fuel Analysis

- ~17,000 simulations
- 3 input parameters:
  - power_scalef = Power scaling factor of the core
  - grainradius_rscalef = Scaling factor of the grain size in the fuel microstructure
  - thermal_expansion = Thermal expansion coefficient of the fuel
- Response value: Midplane Von Mises Stress

Goal: Determine most sensitive parameters within each subset of the domain







How to get the Code

- RAVEN is available for free to:
  - University (no commercial use allowed)
  - DOE

- RAVEN license is worded such that INL retains the intellectual property of any new developments from third parties

- We are currently talking with commercial entities and we would be interested in supporting commercial applications

- Contact: cristian.rabiti@inl.gov, beth.jagger@inl.gov