Workshop – Robust Design Optimization – Acoustic Box

AN UNCERTAINTY QUANTIFICATION AND OPTIMIZATION UNDER UNCERTAINTY TUTORIAL WITH SANDIA DAKOTA AND MSC NASTRAN

Before Starting

This example requires MSC Nastran 2023.3 or newer.

This example uses the case control command DRSPAN that points to a DRESP1 entry with ATTB=MAX. The use of ATTB=MAX creates an equivalent DRESP2 and is managed internally during MSC Nastran's execution. DRSPAN that references DRESP2 is not supported in older versions and produces this USER FATAL MESSAGE. Use MSC Nastran 2023.3 or newer to avoid this error.

```
 *** USER FATAL MESSAGE 7145 (DOPR3H)
     THE DRSPAN COMMOR IN SUBCASE          1 REFERENCES DRESP1 ENTRY ID =   6000001
     WHICH INVOKES MULTIPLE RESPONSES.
     USER INFORMATION: DRESP1 ENTRIES REFERENCED BY DRSPAN REQUEST MUST BE A SCALAR
QUANTITY.
```

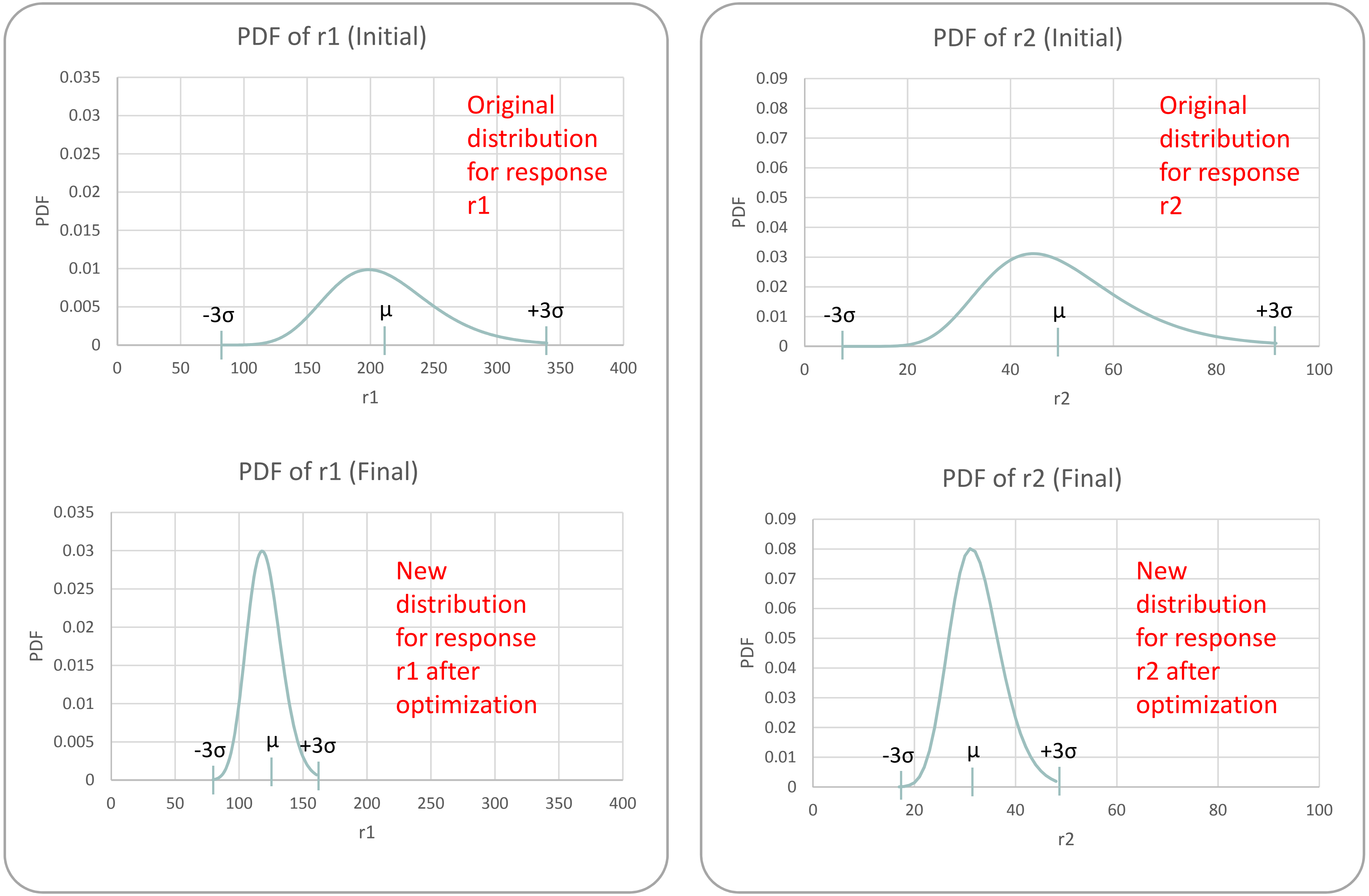
Goal: Use Robust Design Optimization to Minimize the Response Distribution's Spread

Robust design optimization is a type of optimization under uncertainty (OUU) where the goal is to reduce the spread of one or more response distributions. In robust design optimization, a typical objective function is to minimize

$$1 * r_{i,mean} + 3 * r_{i,standard\ deviation}.$$

A characteristic of a robust optimal design is that its responses do not significantly vary when the inputs are uncertain. For example, when manufacturing ground vehicles, small deviations in the final vehicle can lead to a variety of acoustics a passenger hears during vehicle operation. A robust optimal design should minimize the variation of responses.

This exercise uses robust design optimization to yield a robust design for 2 responses corresponding to peak acoustic pressures at 2 locations in a finite element model.



Goal: Use Robust Design Optimization to Minimize the Response Distribution's Spread

Initial Analysis Model Prior To Optimization

Optimal Solution

- Variables
  - x1: .02047
  - x2: .02596
- Max probability of failure:
  - ~0.00% (Actual after UQ with 50 run LHS)

	Mean	Standard Deviation
r1	2.1090780780e+02	4.2544486407e+01
r2	4.9869224760e+01	1.4092696547e+01

Optimization for Stochastic Responses (Sandia Dakota OUU)

Optimal Solution

- 40 MSC Nastran Runs
- Variables
  - x1_mean: 1.71058e-02
  - x2_mean: 2.23106e-02
- Max probability of failure:
  - 0.00% (Approximated probability after final OUU iteration)
  - 0.00% (Actual probability after UQ with LHS of size 50 (50 MSC Nastran runs))

	Mean	Standard Deviation
r1	1.2025857498e+02	1.3528718043e+01
r2	3.2386400460e+01	5.1314392890e+00

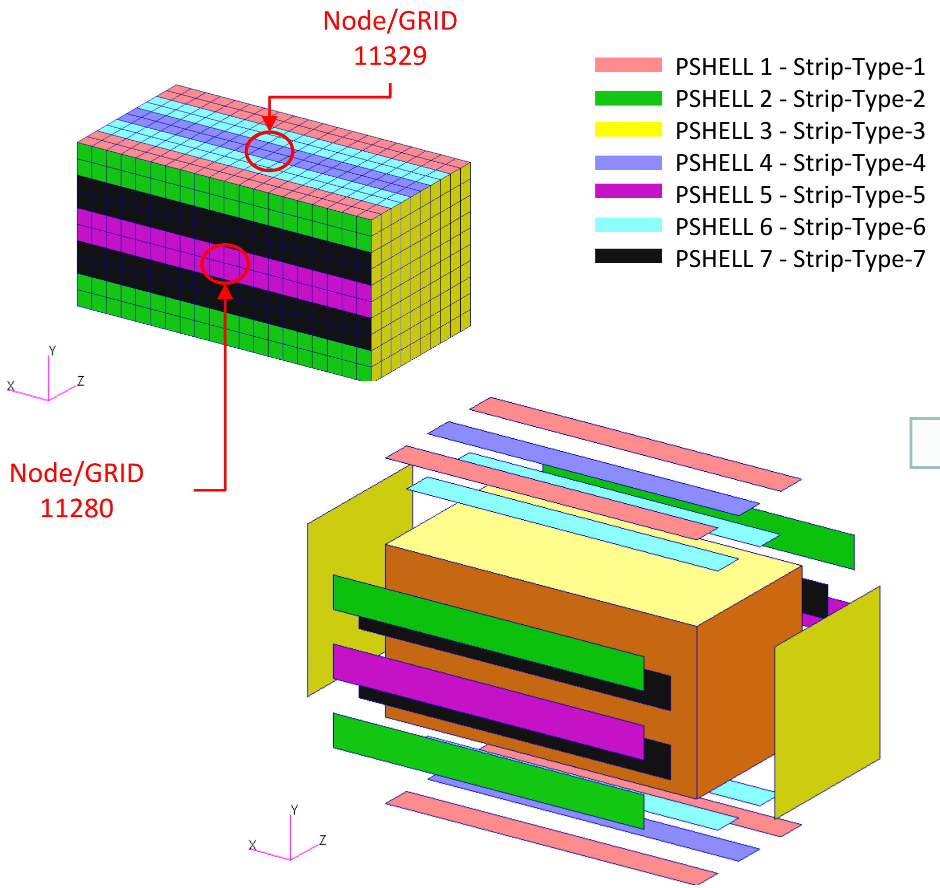
Mean and standard deviation have been reduced

Uncertainty Quantification Problem Statement

Design Variables

x1: T of PSHELL 4
x2: T of PSHELL 5

Variable	Mean	Standard Deviation	Distribution
x1	0.2047	0.001	Lognormal
x2	0.2596	0.001	Lognormal



Responses

r1: Peak acoustic pressure at node 11280, subcase 1

r2: Peak acoustic pressure at node 11329, subcase 2

r3: Weight

Quantities of interest

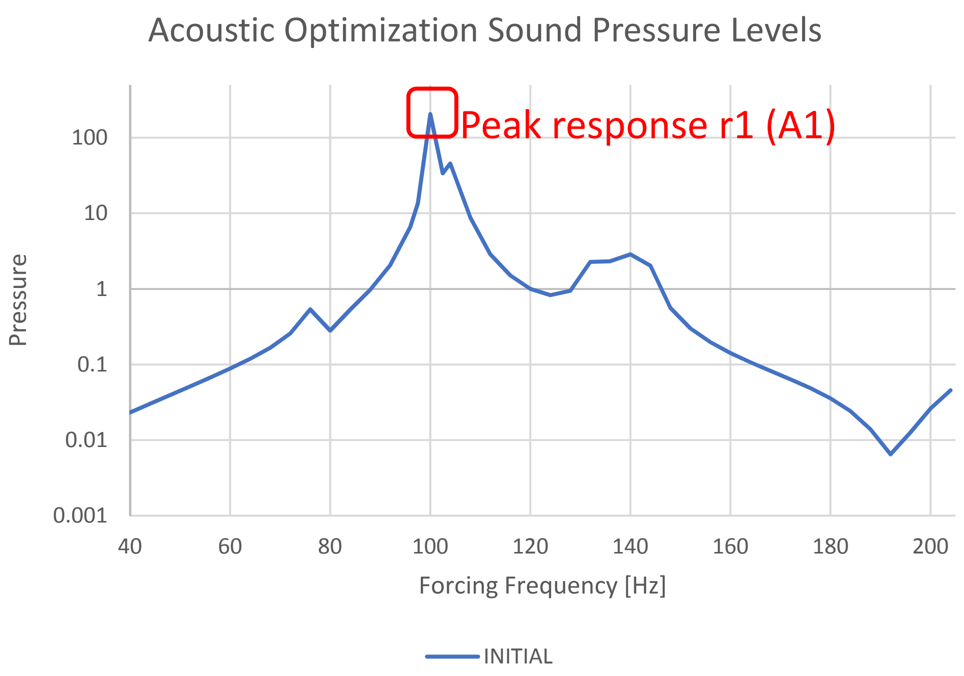
r1: Mean and standard deviation (2 quantities r1_mean, r1_standard_deviation)

r2: Mean and standard deviation (2 quantities r2_mean, r2_standard_deviation)

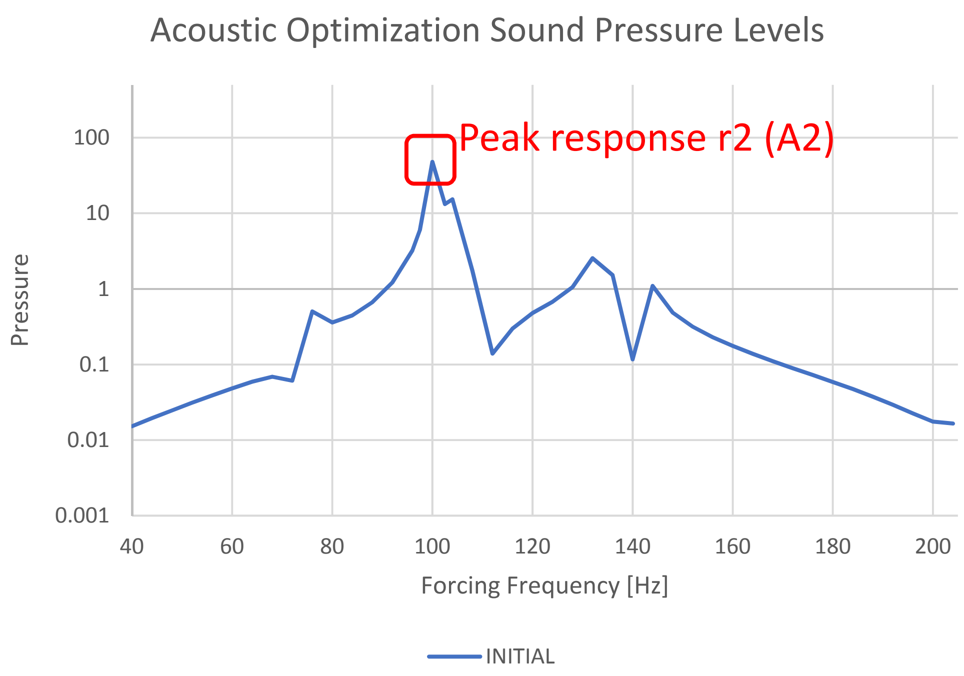
r3: Mean, standard deviation, and probabilities of exceeding the bounds of r3 < 2910 (3 quantities)

Examples of Peak Responses

ACOUSTIC PRESSURE AT NODE 11280 FOR SUBCASE 1



ACOUSTIC PRESSURE AT NODE 11329 FOR SUBCASE 2



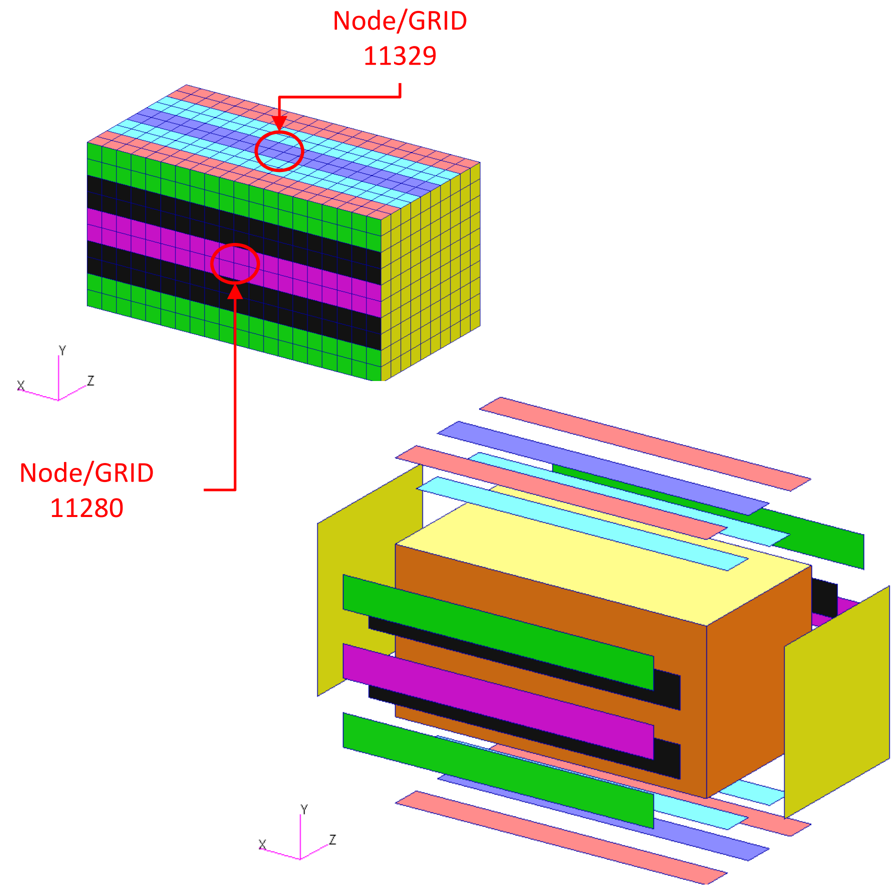
Robust Design Optimization Problem Statement

Design Variables

x1_mean: Mean of x1 (T of PSHELL 4)
x2_mean: Mean of x2 (T of PSHELL 5)

.001 < x1_mean < 1.0
.001 < x2_mean < 1.0

Variable	Initial Value	Lower Bound	Upper Bound
x1_mean	0.2047	.001	1.0
x2_mean	0.2596	.001	1.0



Objective

Minimize

$$1 * r_{1,mean} + 3 * r_{1,standard\ deviation} + 1 * r_{2,mean} + 3 * r_{2,standard\ deviation}$$

Design Constraints

Constraints on probability of failure

r3_pu: P(2910 < r3)
Probability that 2910 < r3

r3_pu < 0.03 (3% failure) ~~0.05 (5% failure)~~

Why a max of 3% probability of failure?

When this exercise was performed with a maximum probability of failure of 5%, the actual final probabilities exceeded the max of 5%.

This is due to the following reasons.

- To reduce computational cost, the probabilities were approximated during OUU. There is an error between the approximate and actual probabilities.
- Optimizers often converge to solutions that are slightly infeasible, i.e. 0.01% violation of constraints.

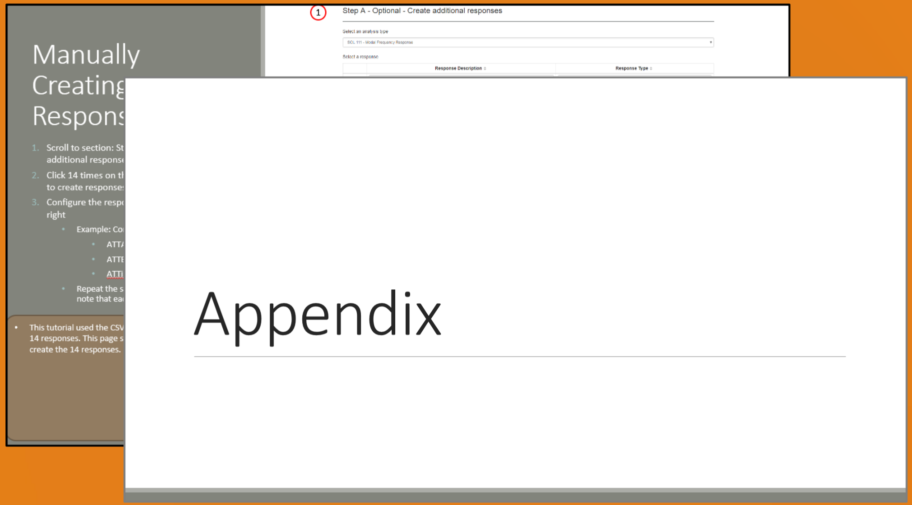
To ensure the final solution is feasible and the maximum probabilities of failure are well below 5%, bounds of 3% are used. When a 3% constraint is used, the final and actual probabilities of failure are 3.75% and are well under the limit of 5%.

r3_pu < 0.03 (3% failure) ~~0.05 (5% failure)~~

More Information Available in the Appendix

The Appendix includes information regarding the following:

- Interpreting the Dakota Input File
- Cumulative and Complementary Probabilities
- Probabilities, Reliability Index and Generalized Reliability Index
- Configuring bounds for probabilities of failure in Sandia Dakota
- Configuring bounds for both UQ and OUU variables in Sandia Dakota



Contact me

- Nastran SOL 200 training
- Nastran SOL 200 questions
- Structural or mechanical optimization questions
- Access to the SOL 200 Web App

christian@ the-engineering-lab.com

Tutorial

Tutorial Overview

1. Start with a .bdf and .h5 file
2. Use the SOL 200 Web App to:
   - Configure an Optimization Under Uncertainty
     - Design Variables
     - Design Objective
     - Design Constraints
   - Perform optimization
3. Plot the Optimization Results

Special Topics Covered

Robust Design Optimization - Small uncertainties in inputs can propagate and yield responses that are highly variable. In robust design optimization, optimal mean values of the inputs are desired such that the variability of responses is minimal. This exercise discusses how to configure a robust design optimization with Sandia Dakota. The responses of interest are generated by the FEA solver MSC Nastran.

SOL 200 Web App Capabilities

The Post-processor Web App and HDF5 Explorer are free to MSC Nastran users.

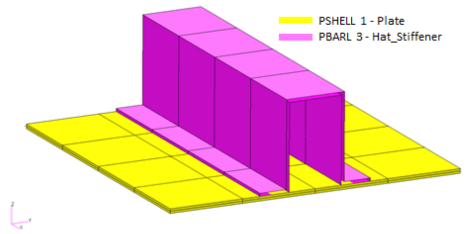
Compatibility

- Google Chrome, Mozilla Firefox or Microsoft Edge
- Windows and Red Hat Linux
- Installable on a company laptop, workstation or server. All data remains within your company.

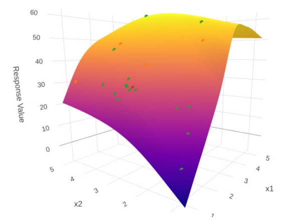
Benefits

- REAL TIME error detection. 200+ error validations.
- REALT TIME creation of bulk data entries.
- Web browser accessible
- Free Post-processor web apps
- +80 tutorials

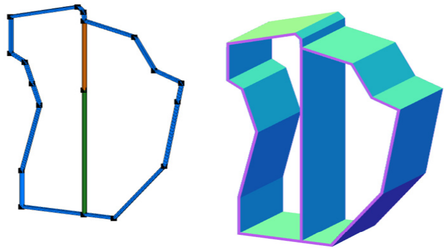
Web Apps



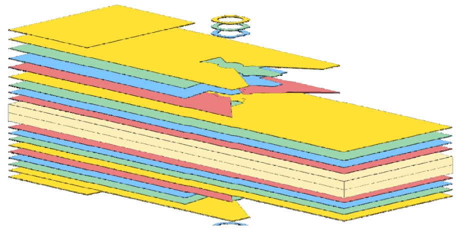
Web Apps for MSC Nastran SOL 200
Pre/post for MSC Nastran SOL 200. Support for size, topology, topometry, topography, multi-model optimization.



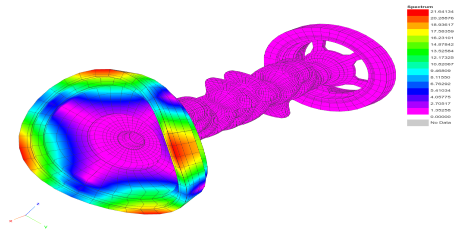
Machine Learning Web App
Bayesian Optimization for nonlinear response optimization (SOL 400)



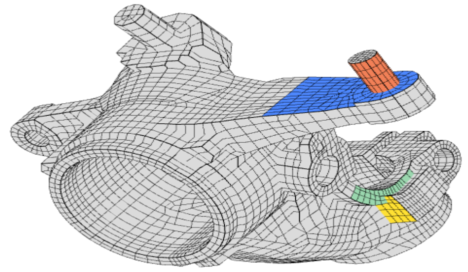
PBMSECT Web App
Generate PBMSECT and PBRSECT entries graphically



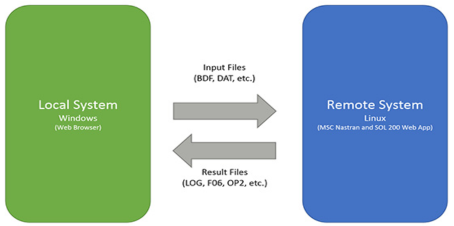
Ply Shape Optimization Web App
Optimize composite ply drop-off locations, and generate new PCOMPG entries



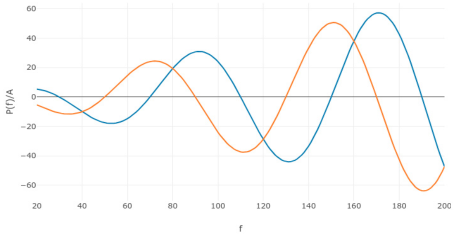
Post-processor Web App
View MSC Nastran results in a web browser on Windows and Linux



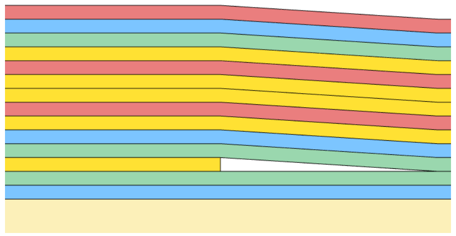
Shape Optimization Web App
Use a web application to configure and perform shape optimization.



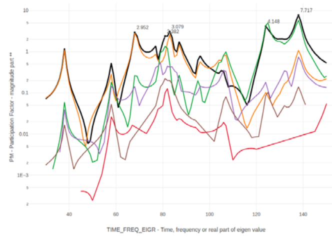
Remote Execution Web App
Run MSC Nastran jobs on remote Linux or Windows systems available on the local network



Dynamic Loads Web App
Generate RLOAD1, RLOAD2 and DLOAD entries graphically



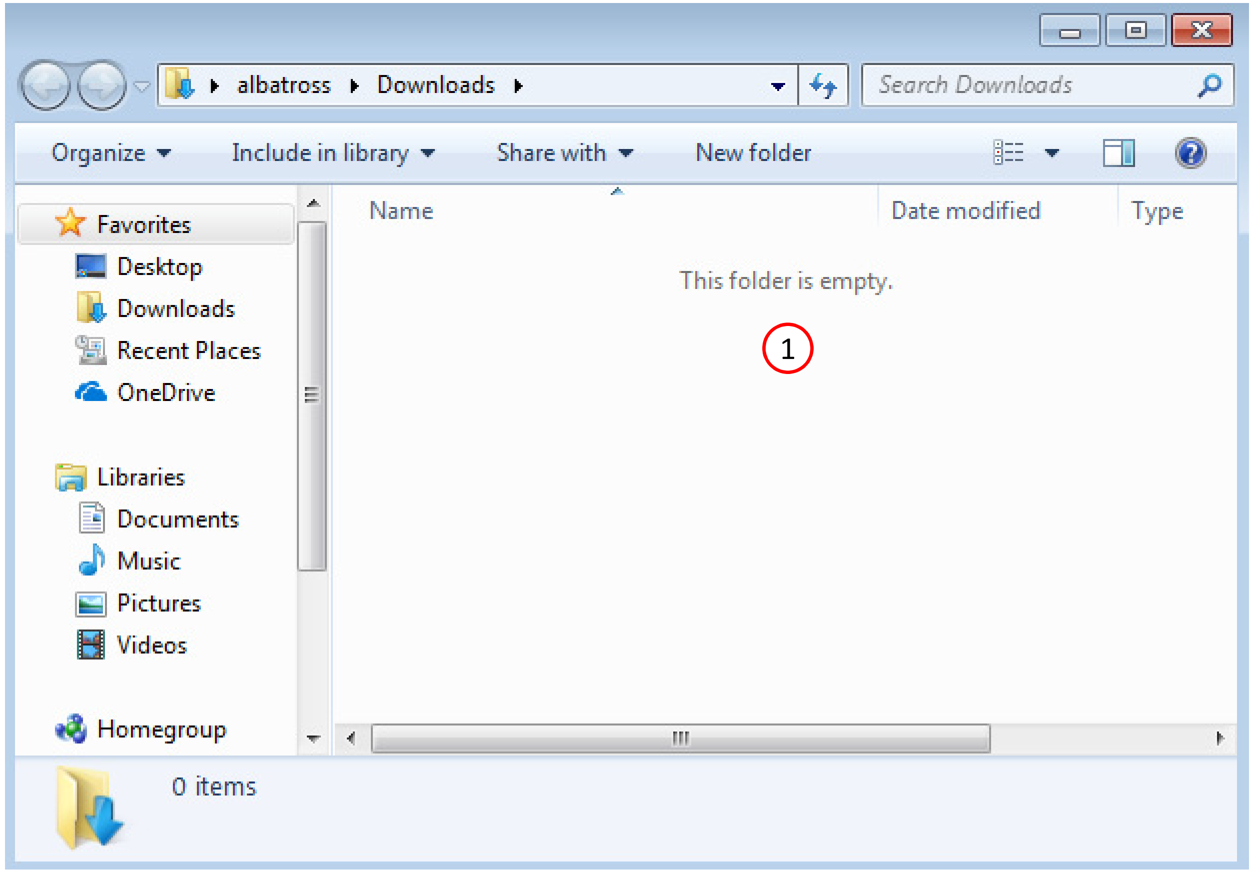
Stacking Sequence Web App
Optimize the stacking sequence of composite laminate plies



HDF5 Explorer Web App
Create graphs (XY plots) using data from the H5 file

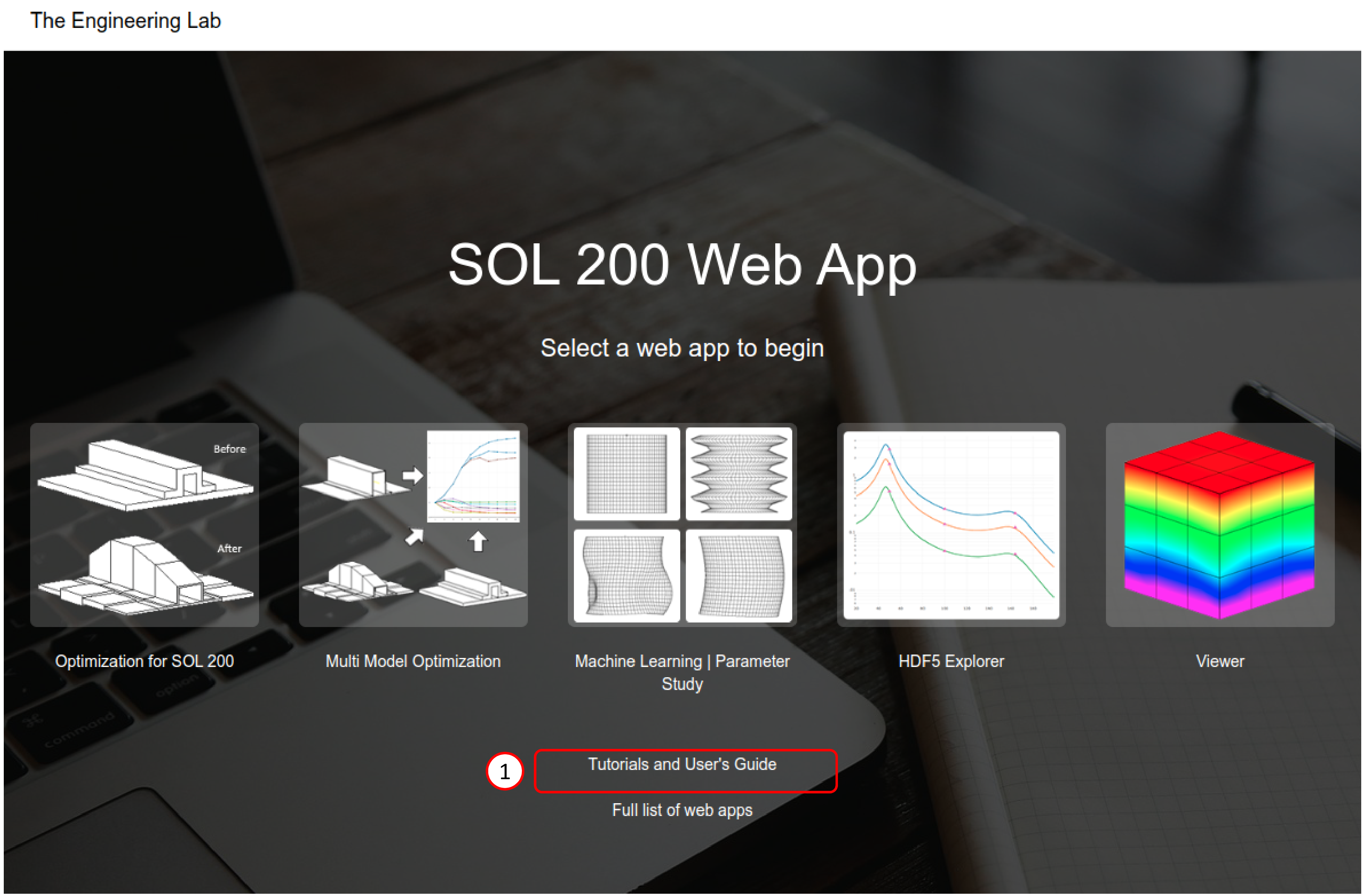
Before Starting

1. Ensure the Downloads directory is empty in order to prevent confusion with other files



Go to the User’s Guide

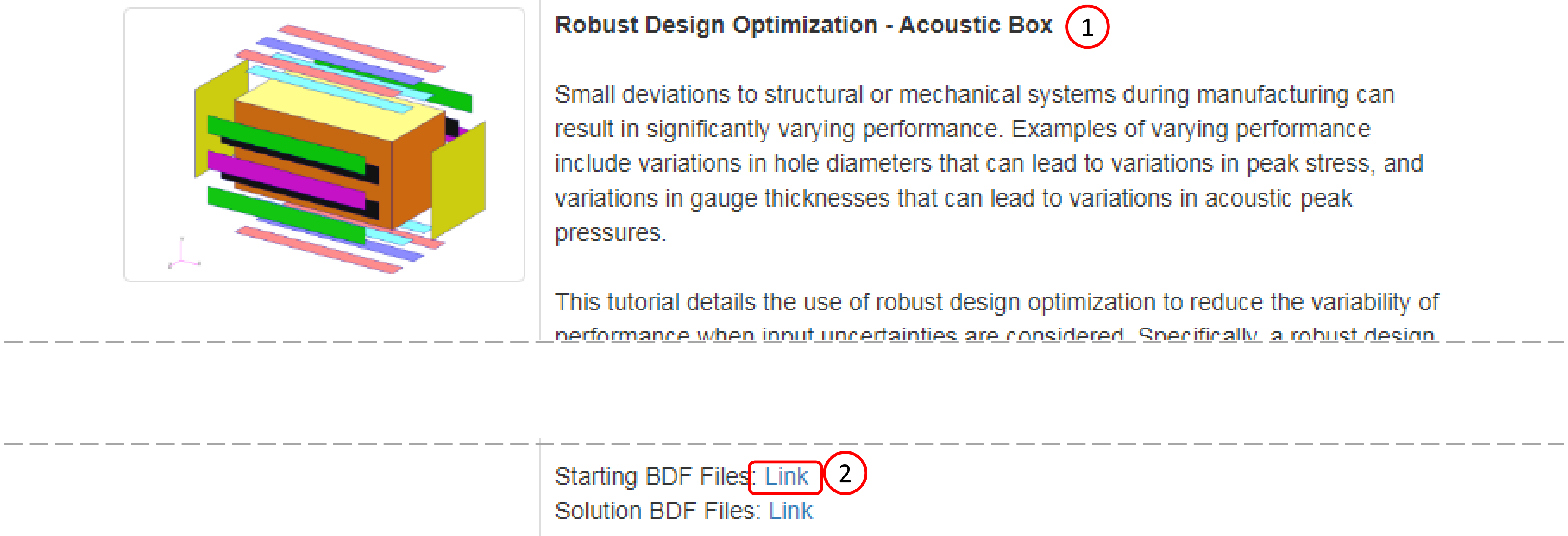
1. Click on the indicated link

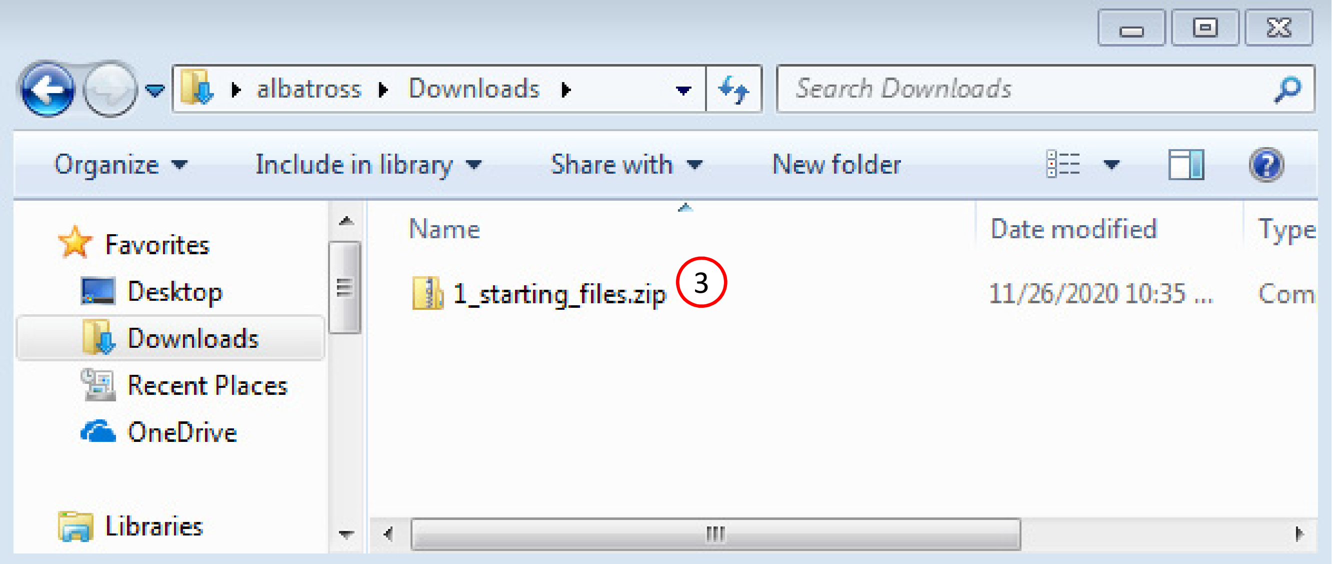


- The necessary BDF files for this tutorial are available in the Tutorials section of the User’s Guide.

Obtain Starting Files

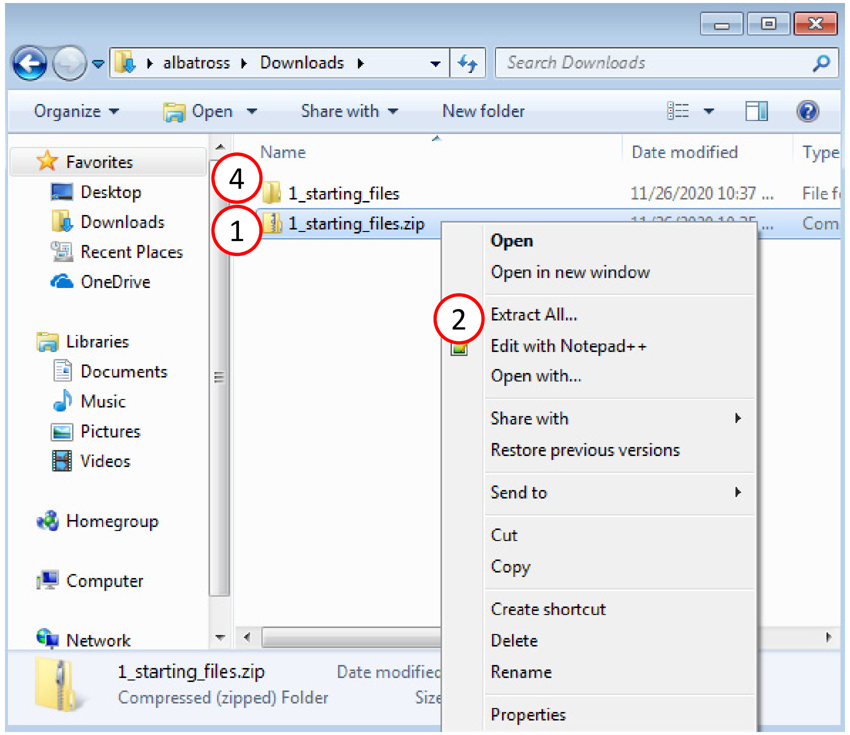
1. Find the indicated example
2. Click Link
3. The starting file has been downloaded

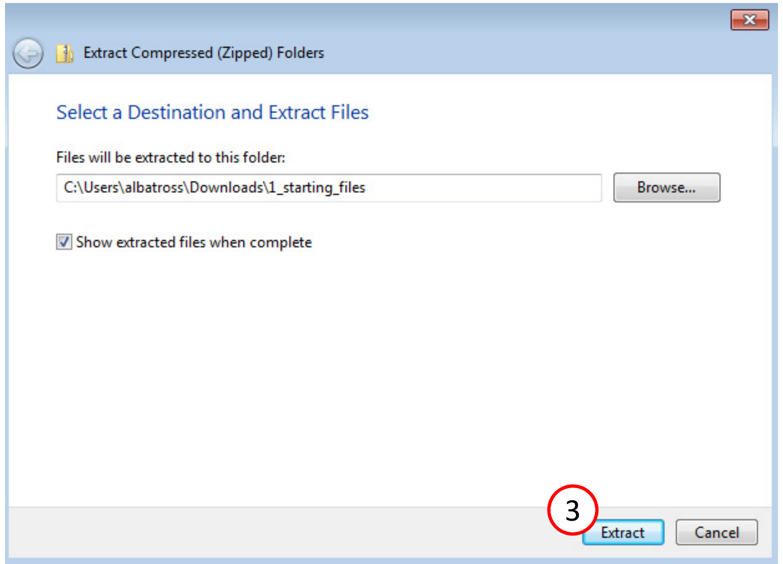


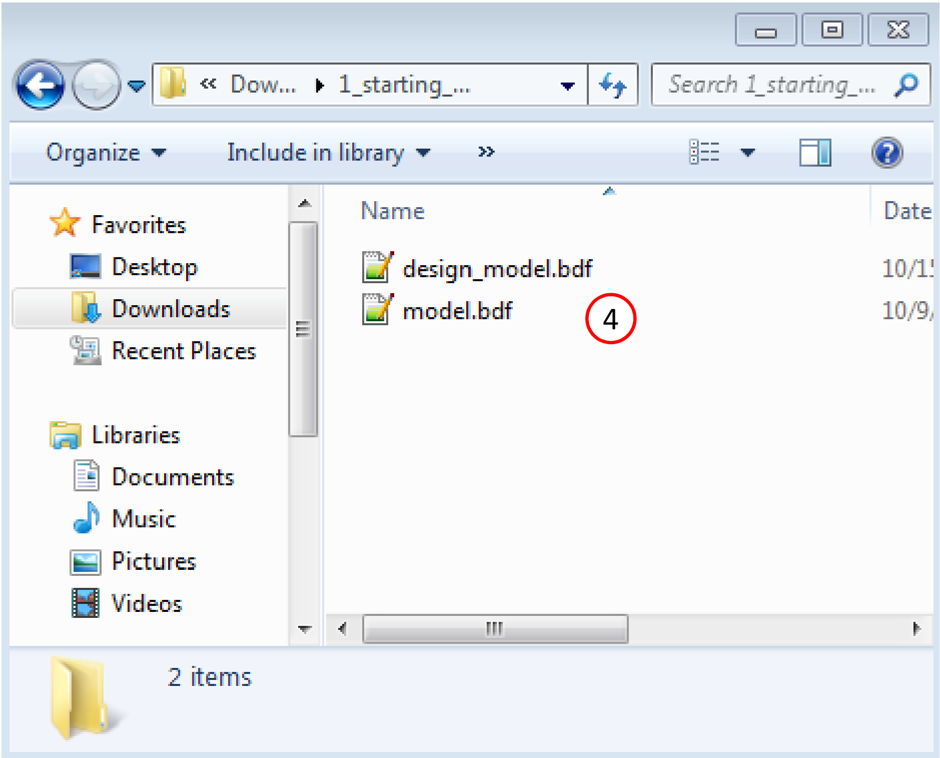


Obtain Starting Files

1. Right click on the zip file
2. Select Extract All…
3. Click Extract
4. The starting files are now available in a folder



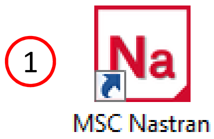


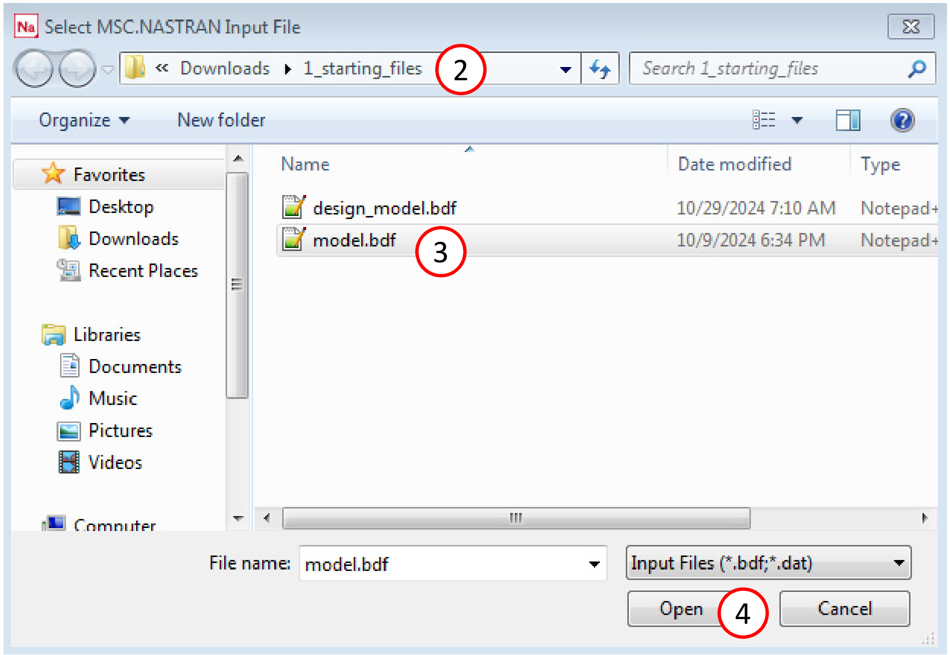


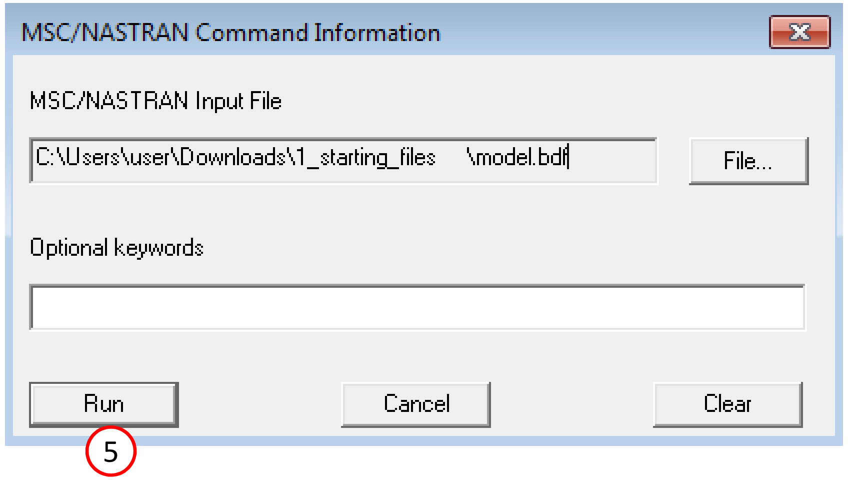
Create the Starting H5 File

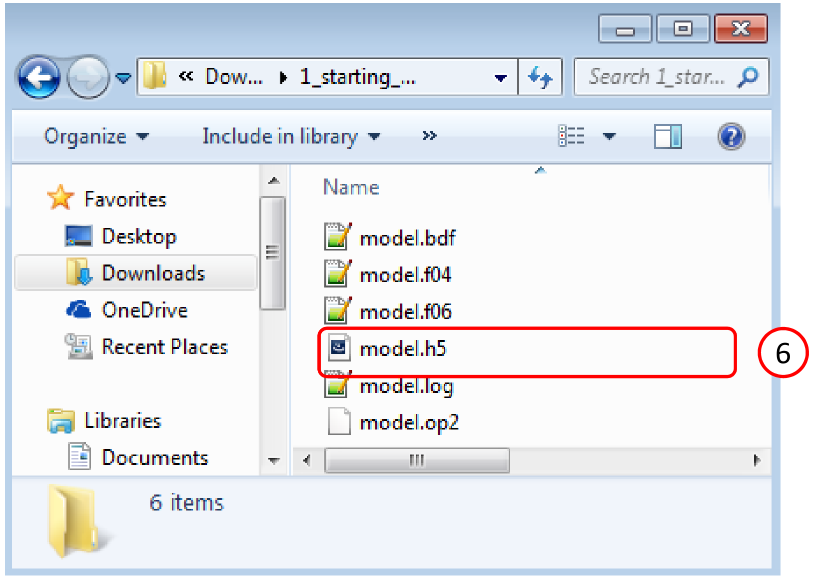
A starting H5 file must be created. This H5 file will be used to configure the responses later on.

1. Double click the MSC Nastran desktop shortcut
2. Navigate to the directory named 1_starting_files
3. Select the indicated file
4. Click Open
5. Click Run
6. The starting H5 file is created









Use the same MSC Nastran version throughout this exercise

The following applies if you have multiple versions of MSC Nastran installed.

To ensure compatibility, use the same MSC Nastran version throughout this exercise.
For example, scenario 1 is OK but scenario 2 is NOT OK.

- Scenario 1 - OK
  - MSC Nastran 2021 is used to create the starting H5 file.
  - MSC Nastran 2021 is used for each run during Machine Learning or Parameter study.
- Scenario 2 – NOT OK
  - MSC Nastran 2018.2 is used to create the starting H5 file.
  - MSC Nastran 2021 is used for each run during Machine Learning or Parameter study.

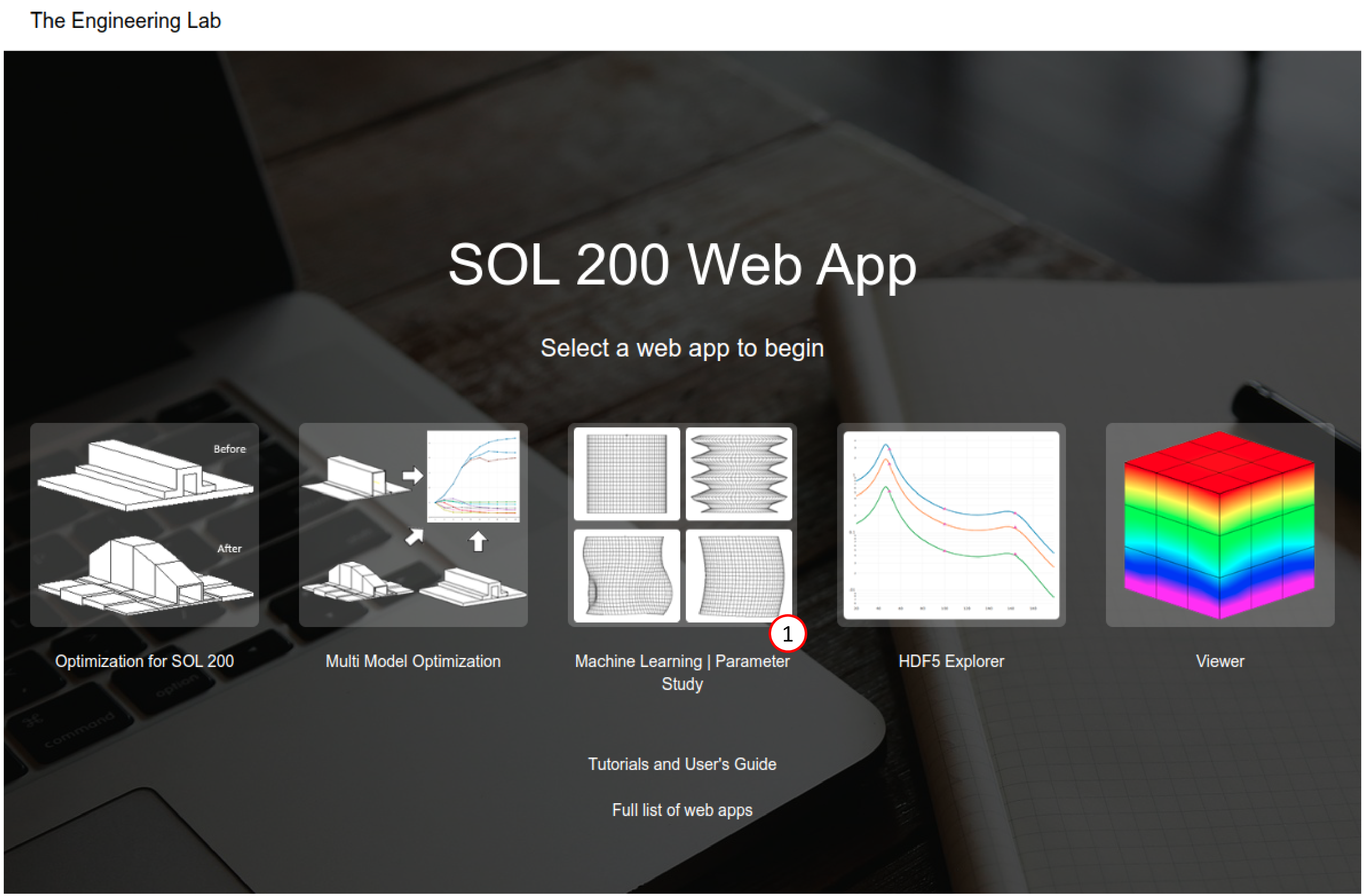
Using the same MSC Nastran version is critical for consistent response extraction from the H5 file. A response configured for Nastran version X may not match in Nastran version Y, which leads to unsuccessful response extraction from the H5 files. The goal is to make sure all H5 files generated are from the same MSC Nastran version.

Part A – Robust Design Optimization

Open the Correct Page

1. Click on the indicated link

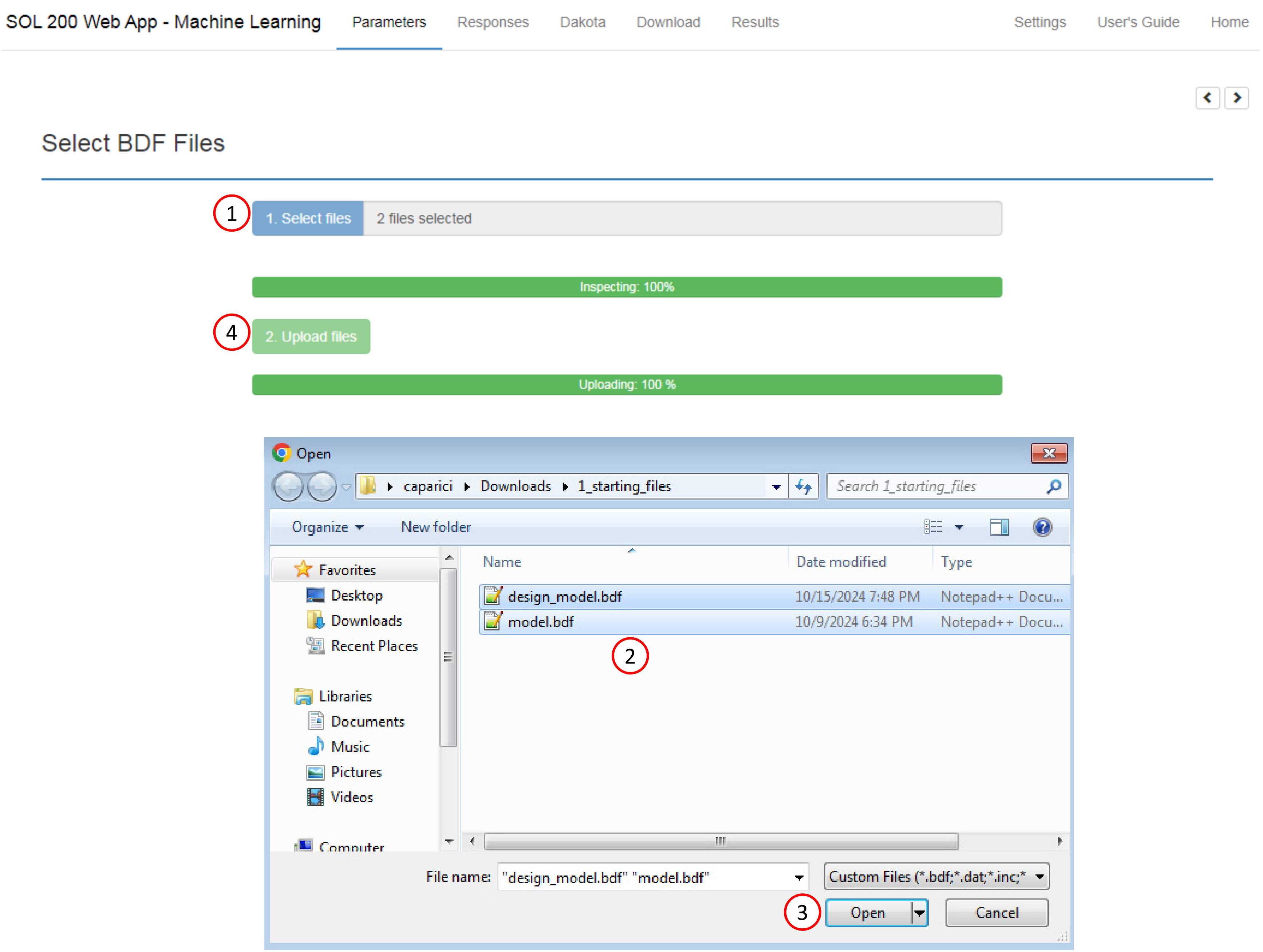
- MSC Nastran can perform many optimization types. The SOL 200 Web App includes dedicated web apps for the following:
    - Optimization for SOL 200 (Size, Topology, Topometry, Topography, Local Optimization, Sensitivity Analysis and Global Optimization)
    - Multi Model Optimization
    - Machine Learning
- The web app also features the HDF5 Explorer, a web application to extract results from the H5 file type.



Select BDF Files

1. Click Select files
2. Select the indicated file
3. Click Open
4. Click Upload files

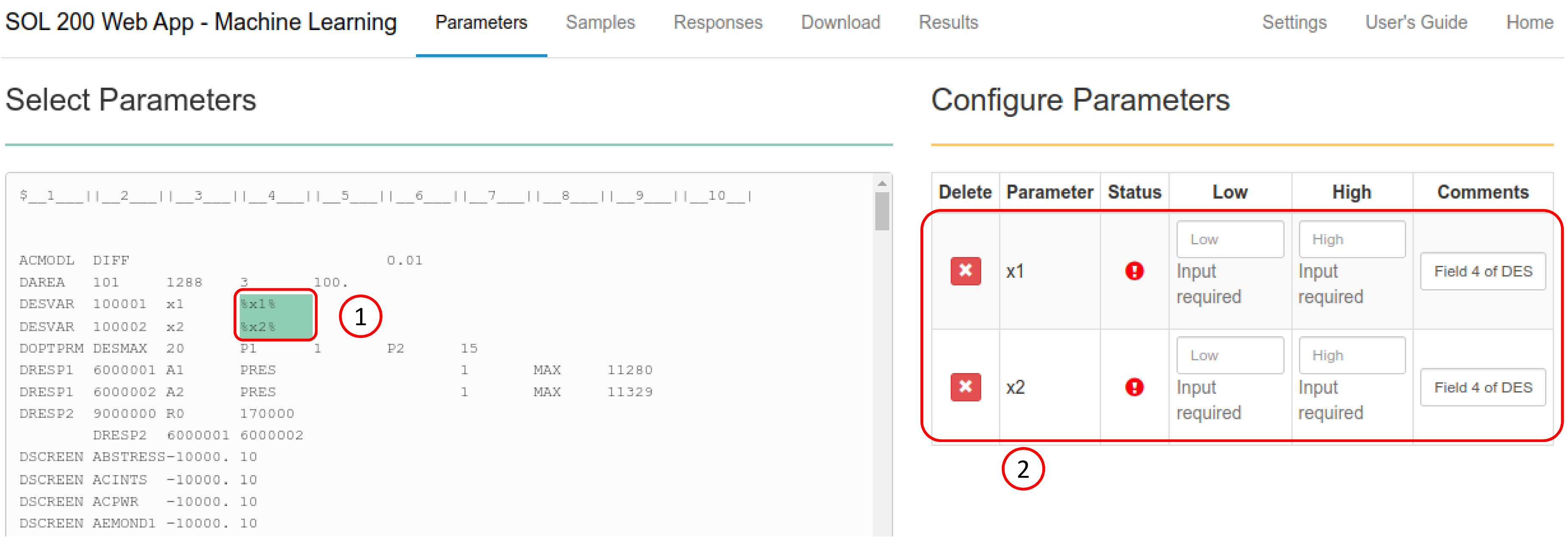
- When starting the procedure, all the necessary BDF, or DAT, files must be collected and uploaded together. Relevant INCLUDE files must also be collected and uploaded.



Parameters

1. Set the following fields as parameters
   - x1: Initial value, field 4, of DESVAR 100001
   - x2: Initial value, field 4, of DESVAR 100002
2. Two new variables should be listed

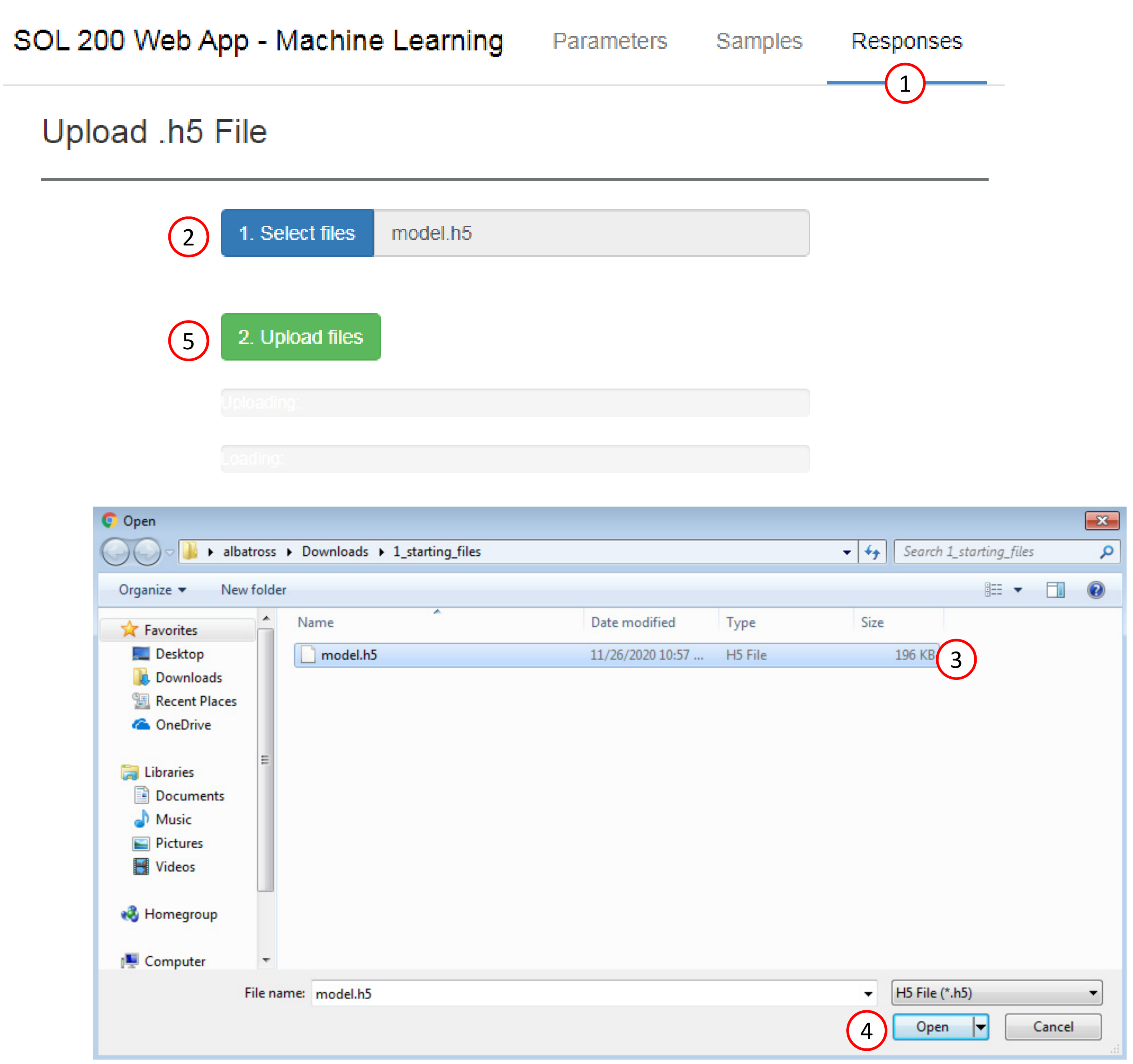
- Since DESVAR and DVPREL1 entries are configured for the thickness of PSHELL entries, the initial value field of DESVAR entries must be set as a parameter.



Responses

1. Click Responses
2. Click Select files
3. Select the indicated file
4. Click Open
5. Click Upload files

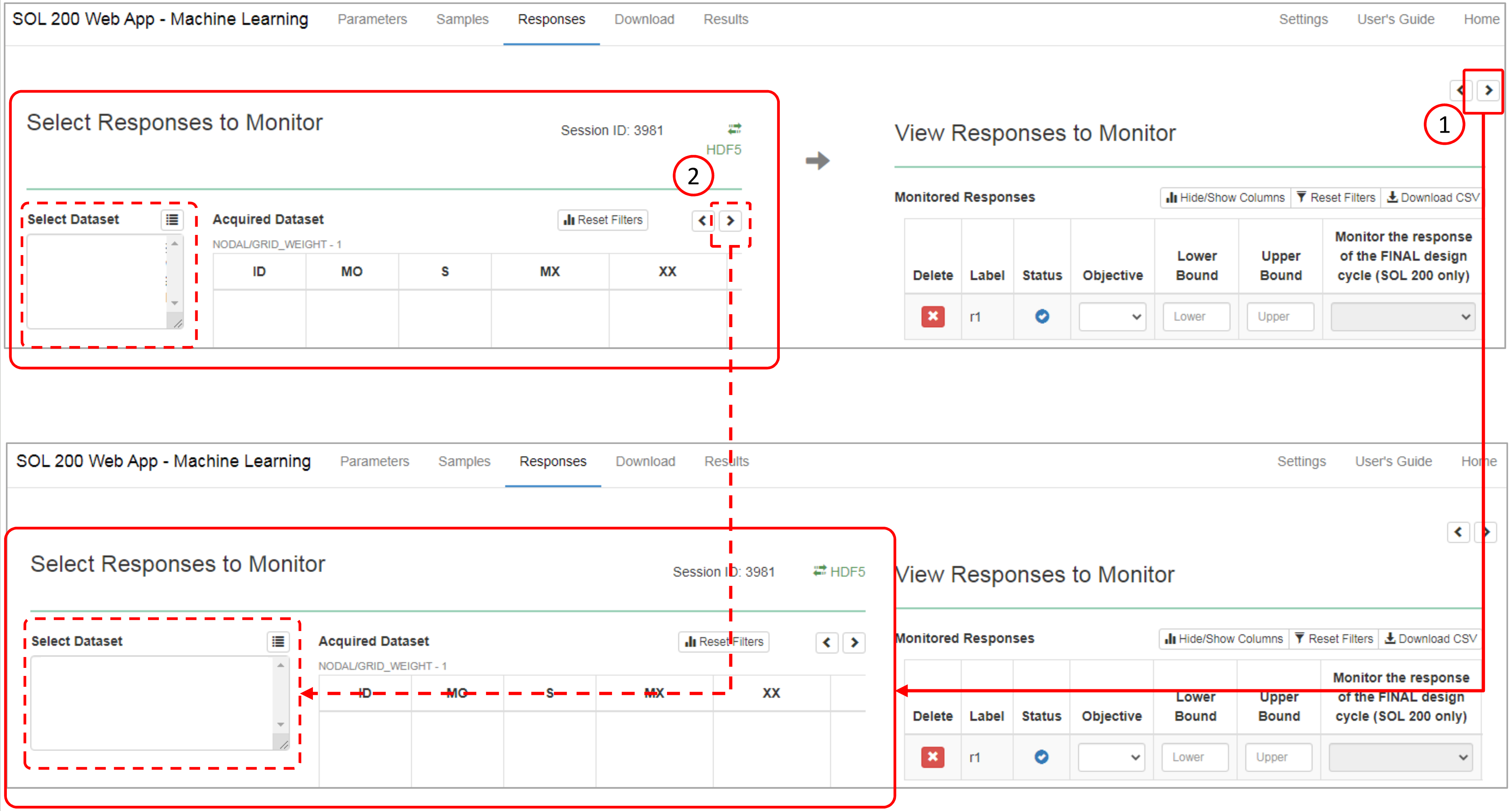
- On this page, the H5 file is uploaded to the web app.



Adjust the Column Width

1. Optional - Use at your liking the buttons at the top right hand corner to adjust the width of the left and right columns
2. Optional – Use the indicated buttons to adjust the width of the column Select Dataset

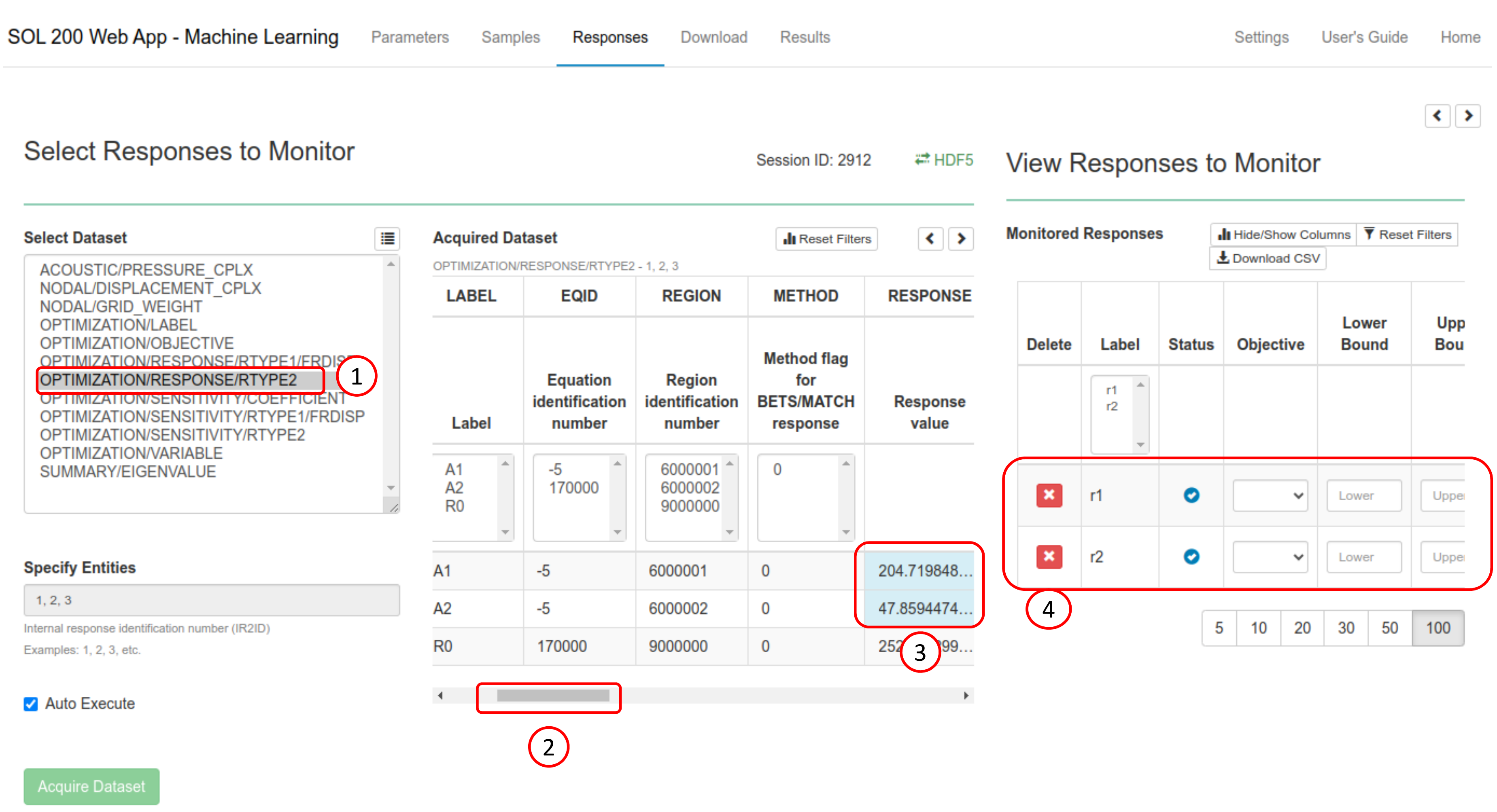
- IMPORTANT! This image is not meant to match exactly what you see in your view. The text in this image is expected to be different from your view. The purpose of this page and image is to demonstrate how to increase the width of the indicated sections.



Select Responses

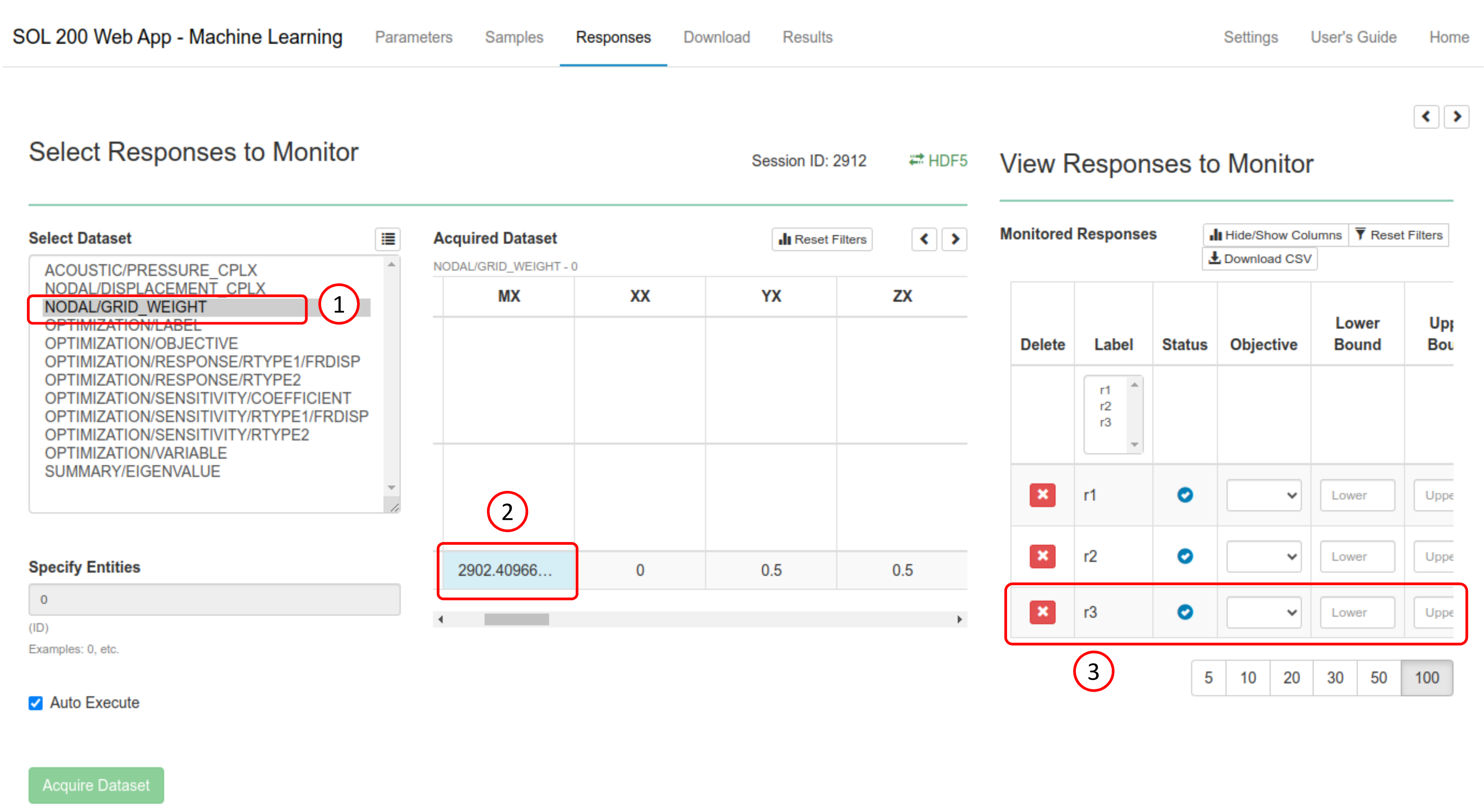
1. Select the following dataset: OPTIMIZATION/RESPONSE/RTYPE2
2. If needed, use the horizontal scroll bar to find the RESPONSE column
2. Select the 2 indicated cells
1. New response r1 and r2 have been created

- Recall that response A1 is the peak acoustic pressure at node 11280 from subcase 1. This is set as response r1 for an upcoming robust design optimization.
- Response A2 is the peak acoustic pressure at node 11329 from subcase 2. This is set as response r2 for an upcoming robust design optimization.



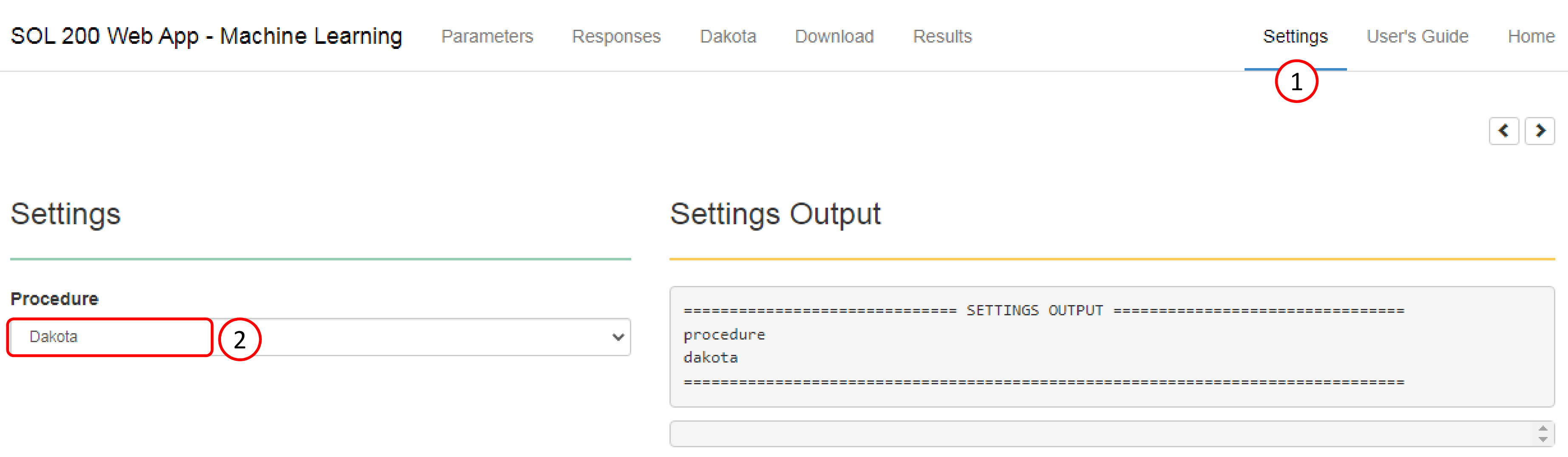
Select Responses

1. Select the following dataset: NODAL/GRID_WEIGHT
2. Select the indicated cell
3. New response r3 has been created



Settings

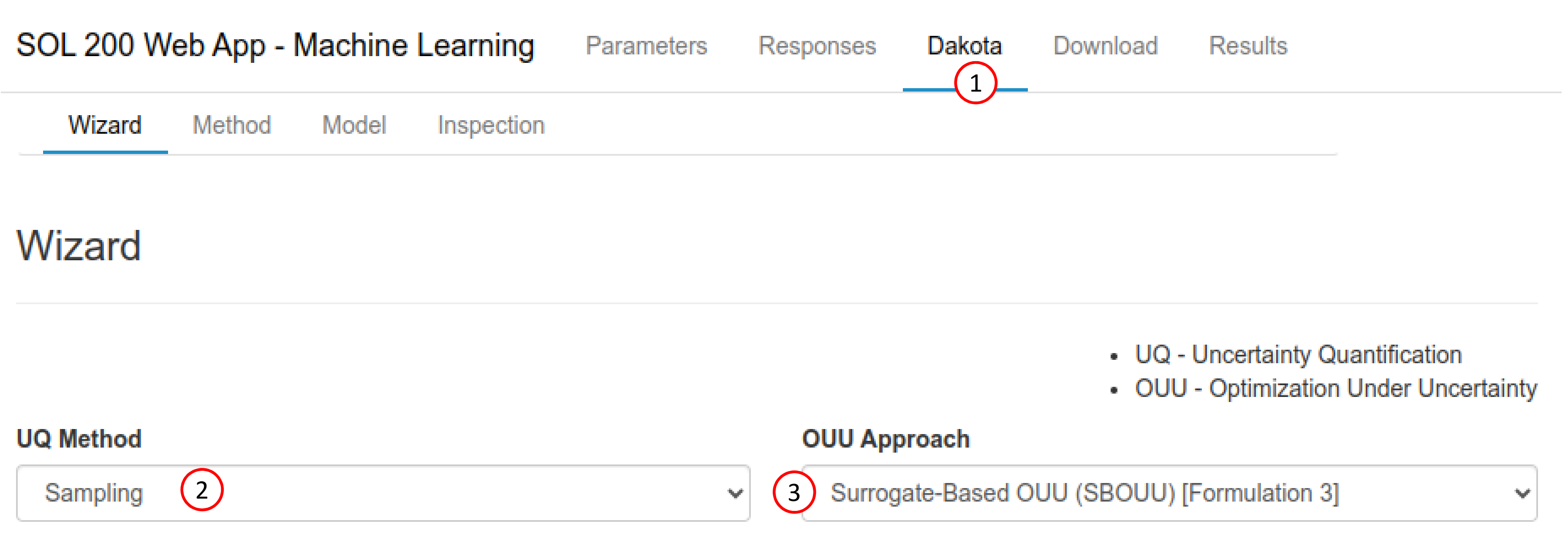
1. Click Settings
2. Set Procedure to Dakota



Dakota

1. Click Dakota
2. Set UQ Method to Sampling
3. Set OUU Approach to Surrogate-Based OUU (SBOUU) [Formulation 3]

- Let's assume that reliability methods, such as the MVFOSM method, is unsuitable for uncertainty quantification of responses r1 and r2. For OUU, we will rely on a surrogate model for responses r1, r2 and r3. For UQ, the surrogate models will be sampled instead of directly running the FEA solver MSC Nastran.



Dakota - Uncertainty Quantification (UQ)

1. Scroll to section Uncertainty Quantification
2. Set both distributions to Lognormal Uncertain
3. Set both standard deviations to 0.001
4. For this example, bounds are not used. Ensure the bounds are blank.

- Variables that are normally distributed allow for negative values. This is problematic if the variable should always be positive. In this example, the cross sectional area is varied and should always be positive, else if the area is negative, the FEA solver will fail. A lognormal distribution allows for only positive values. The variables in this exercise are configured as having a lognormal distribution.
- The standard deviation is often determined via testing or provided by the supplier or manufacturer.
- In this exercise, bounds are not provided for the uncertain variables. Bounds are provided for the optimization variables later on in this exercise. If there is a desire to provide bounds for the uncertain variables, refer to the information in the Appendix, section *Configuring bounds for both UQ and OUU variables in Sandia Dakota*.

Uncertainty Quantification (1)

Configure UQ Variables

Reset Table

Delete	Descriptor	Status	Distribution	Mean	Standard Deviation	Initial Value	Lower Bound	Upper Bound	Description
✖	x1	✔	Lognormal	.02047	0.001		UQ Lower l	UQ Upper l	
✖	x2	✔	Lognormal	.02596	0.001		UQ Lower l	UQ Upper l	

(2) (3) (4)

Dakota - Optimization Under Uncertainty (OUU)

1. Scroll to section Optimization Under Uncertainty
2. Set the means of x1 and x2 as variables during OUU
3. For x1_mean, set the following:
   - Initial Value: .02047
   - Lower Bound: 0.001
   - Upper Bound: 1.0
4. For x2_mean, set the following:
   - Initial Value: .02596
   - Lower Bound: 0.001
   - Upper Bound: 1.0

Optimization Under Uncertainty (1)

Select OUU Variables

Reset Table

Descriptor	Initial Value	Mean	Description
x1		+ Mean	Field 4 of DESVAR
x2		+ Mean	Field 4 of DESVAR

(2)

Configure OUU Variables

Reset Table

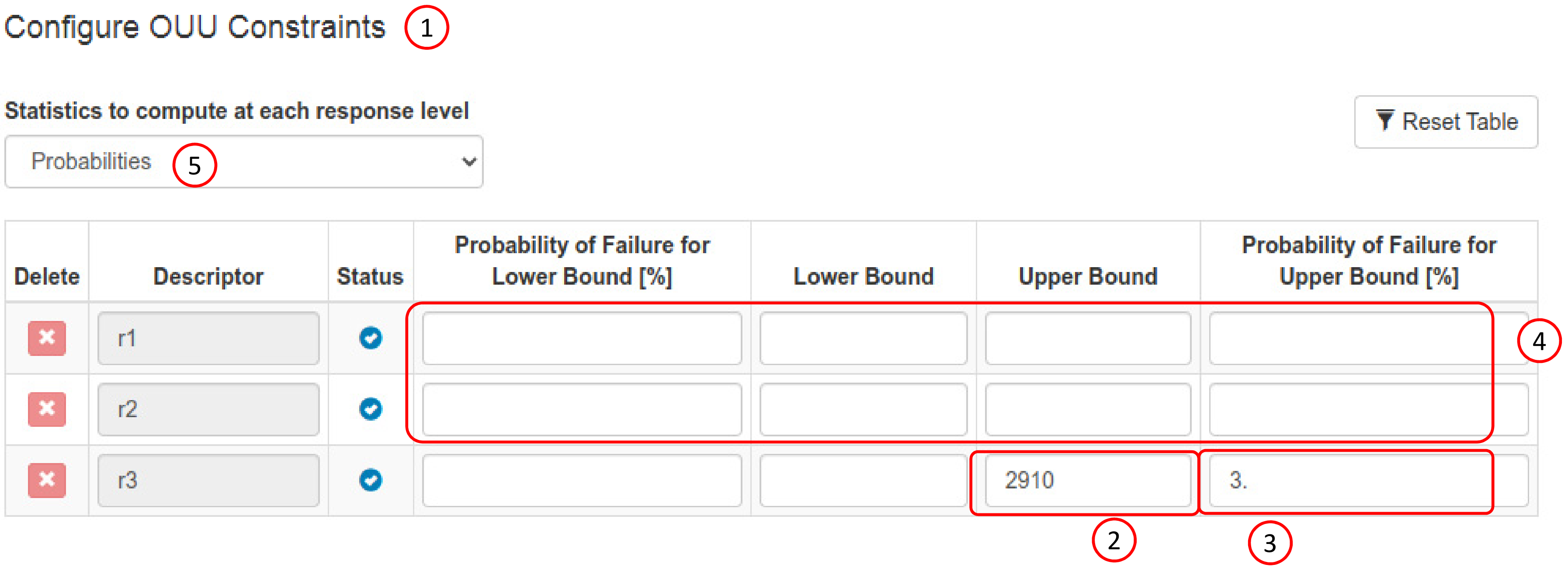
Delete	Descriptor	Status	Initial Value	Lower Bound	Upper Bound	Description
✖	x1_mean	✔	.02047	0.001	1.0	Mean - Field 4 of DESVAR
✖	x2_mean	✔	.02596	0.001	1.0	Mean - Field 4 of DESVAR

(3) (4)

Dakota - Uncertainty Quantification (UQ)

1. Scroll to section Configure OUU Constraints
2. Set the following bound on the weight response r3
   - Upper Bound: 2910
3. Set the following bounds on the probabilities of failure
   - Probability of Failure Upper Bound: 3
4. Do NOT provide any constraint information for response r1 and r2, i.e. do NOT constrain the peak acoustic pressure responses.
5. Set the Statistics to compute at each response level to Probabilities

- The probability of exceeding each bound is set to 3%. Why 3% and not 5%? Optimizers often yield final solutions where their constraints have a slight violation, e.g. 0.01% violation. 3% is used to ensure the final probabilities are well below 5%.



Dakota - Optimization Under Uncertainty (OUU)

Since a robust design is desired for responses r1 and r2, the peak acoustic responses, the following objective response must be defined:

$$1 * r_{1,mean} + 3 * r_{1,standard\ deviation} + 1 * r_{2,mean} + 3 * r_{2,standard\ deviation}$$

1. Scroll to section Configure OUU Objective and Additional Constraints
2. Click the indicated buttons to include the mean and standard deviation of response r1 and r2 in the objective.
3. Ensure the scale factor is 1.0 and 3.0 are used for the mean and standard deviation, respectively.
4. Notice that the constraints on probability of failure have been automatically created.

A. As an option, when Create Constraint is clicked, a new constraint is added and may be configured to constrain additional quantities, e.g. μ + 3σ.

Configure OUU Objective and Additional Constraints (1)

(A) + Create Constraint

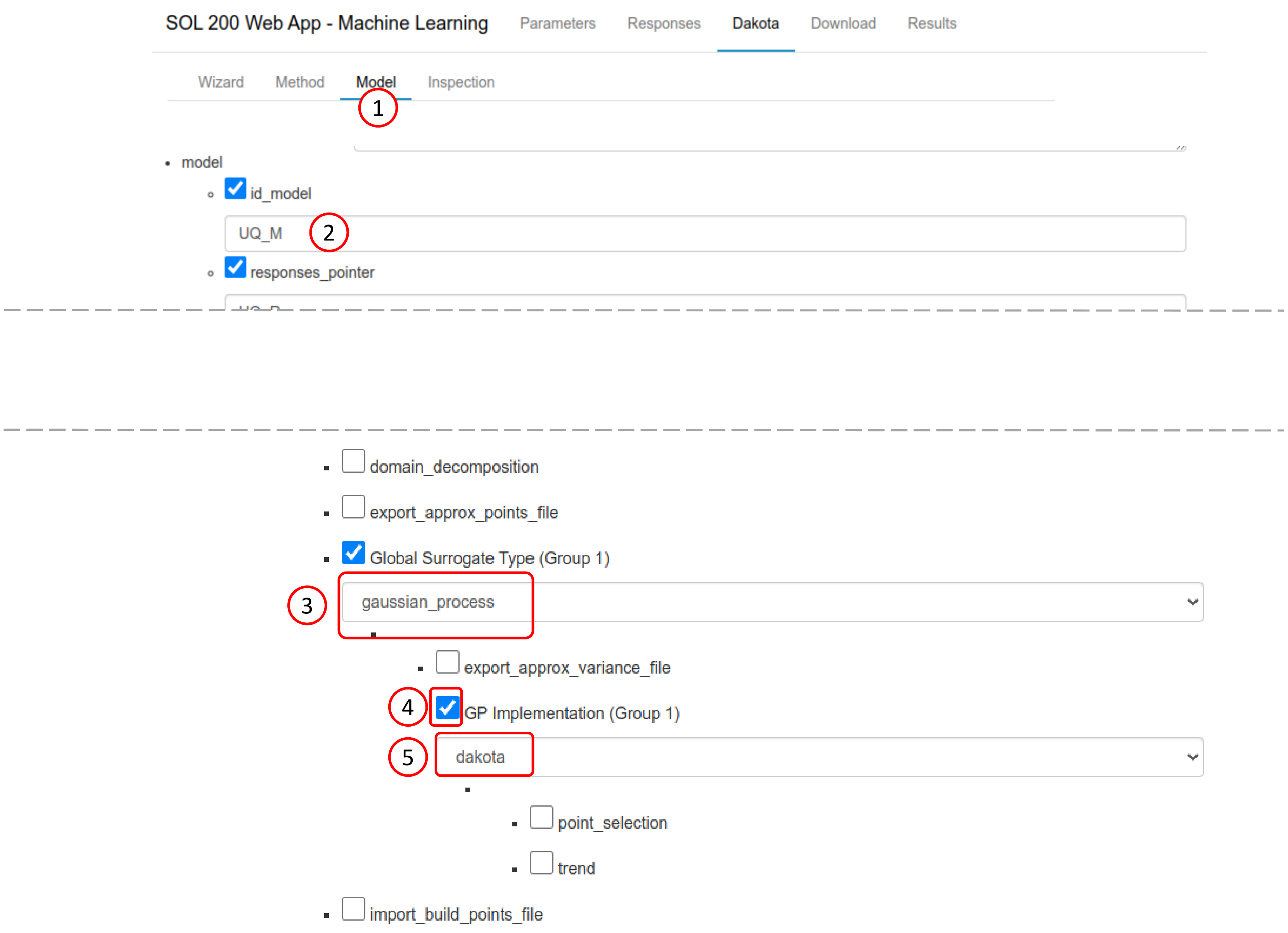
	Objective (f_obj, g1)		Constraint 1 (r3_pu)	
Label	Include	Scale Factor	Include	Scale Factor
r1_mean	☑	1.	☐	
r1_standard_deviation	☑	3.	☐	
r2_mean	☑	1.	☐	
r2_standard_deviation	☑	3.	☐	
r3_mean	☐		☐	
r3_standard_deviation	☐		☐	
r3_p1	☐		☑	1.
Lower Bound				
Upper Bound			0.030000	

(2) (3) (4)

Dakota - Optimization Under Uncertainty (OUU)

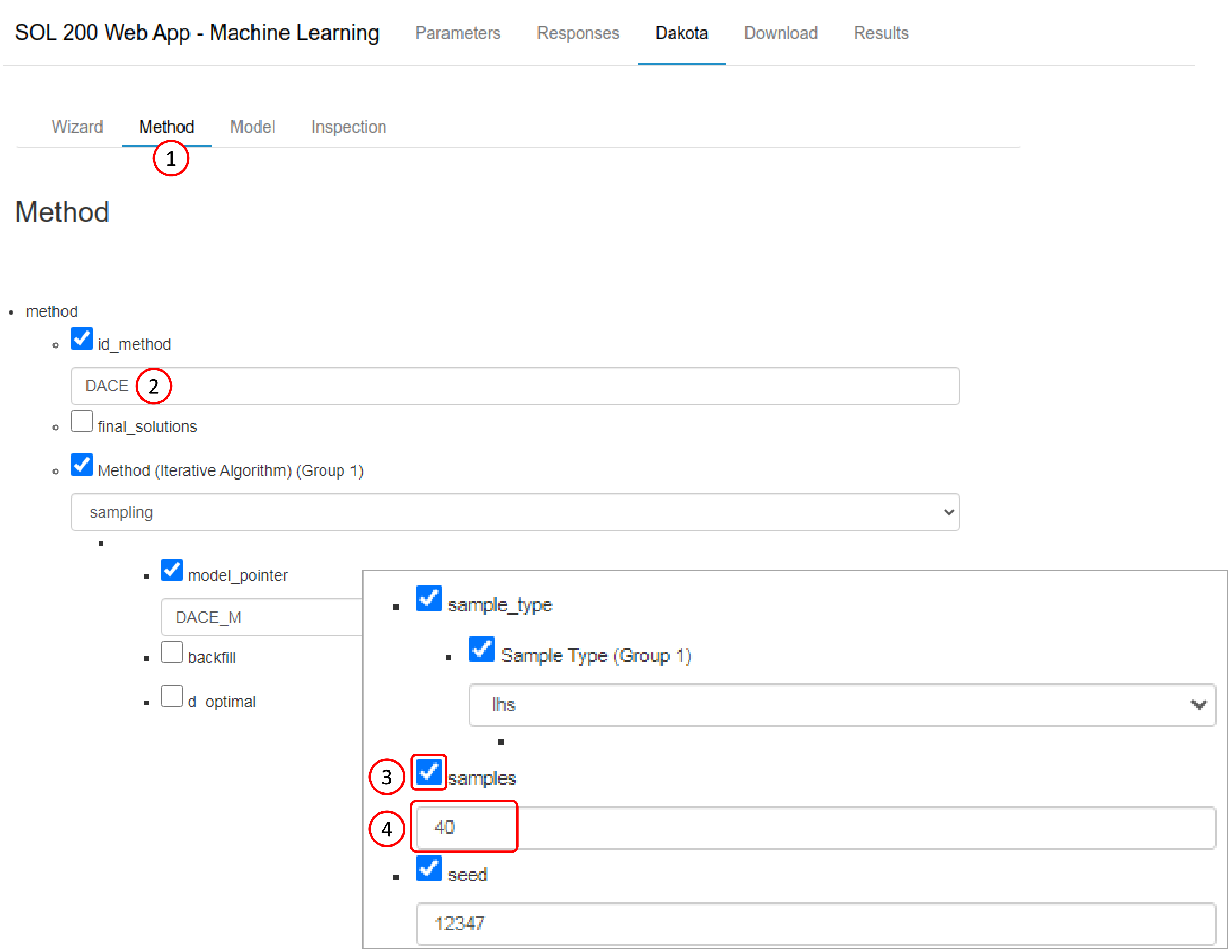
1. Click Model
2. Find the input box where id_model='UQ_M'
3. Set the Global Surrogate Type to gaussian_process
4. Mark the indicated checkbox for GP Implementation
5. Set the option to dakota

- The Gaussian process model is configured as the surrogate model



Dakota - Optimization Under Uncertainty (OUU)

1. Click Method
2. Find the input box where id_method='DACE'
3. Mark the checkbox for samples
4. Set samples to 40



- The Gaussian process model must be created by using training data. The training data is acquired by running MSC Nastran 40 times at various values for the variables and collecting the corresponding responses. An LHS of size 40 is used.

What is formulation 3?

No steps are required on this page. The information is meant to elaborate on the configurations made to the surrogate mode.

A. Previously, the OUU Approach selected was Surrogate-Based OUU (SBOUU) [Formulation 3]. This approach involves the creation of a surrogate model for the responses. Whenever the optimizer needs response values, the optimizer queries the surrogate model instead of the black box function, which in this case is the FEA solver MSC Nastran.

B. The surrogate model is a Gaussian process (kriging) model. An LHS of size 40, or 40 MSC Nastran runs, are used to construct the surrogate models for r1, r2, and r3.

Source: Dakota User's Manual

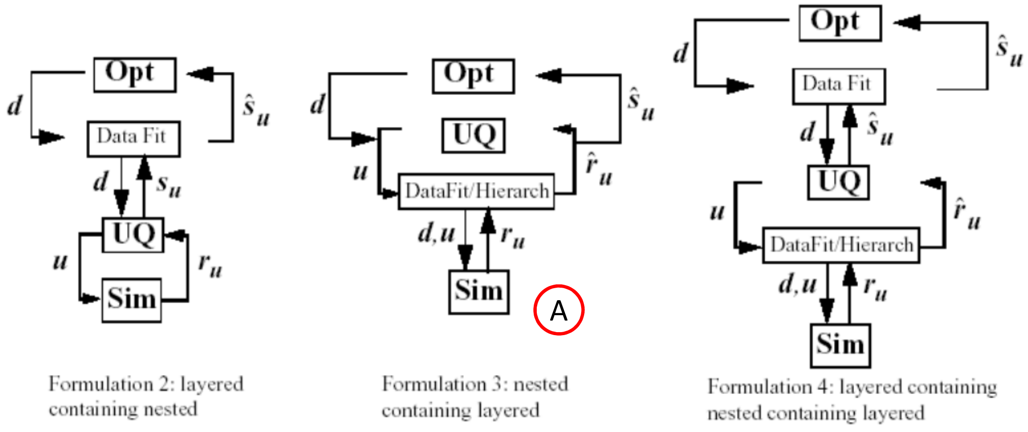
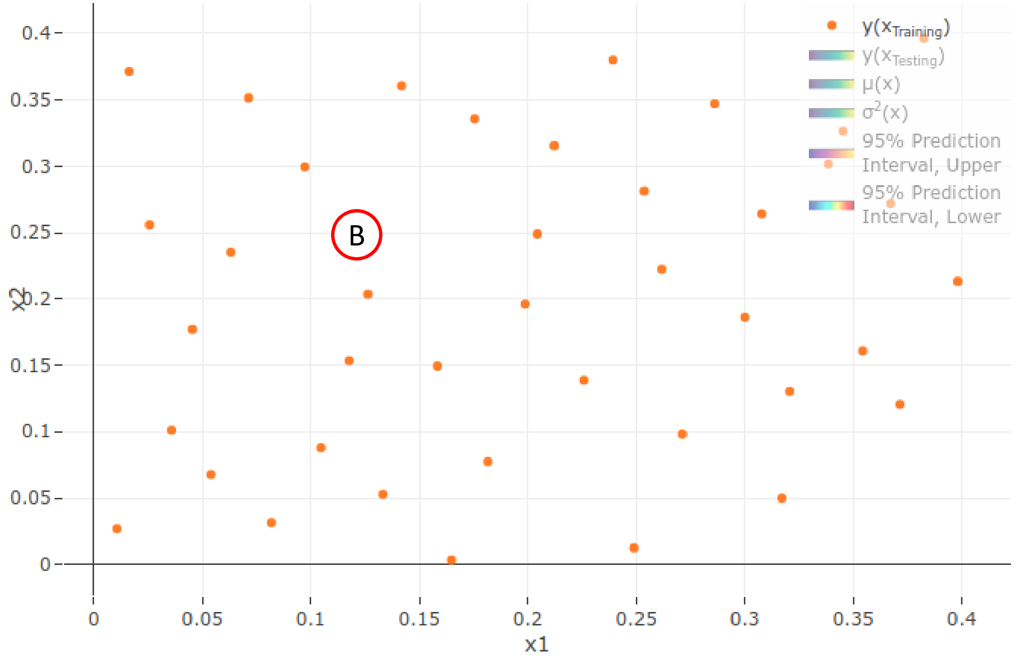


Figure 15.6: Formulations 2, 3, and 4 for Surrogate-based OUU.



Surrogate Models

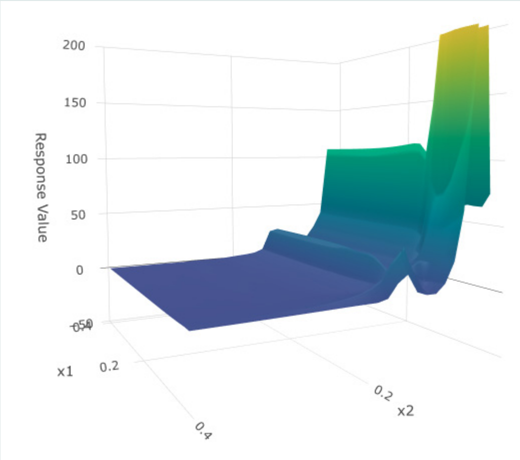
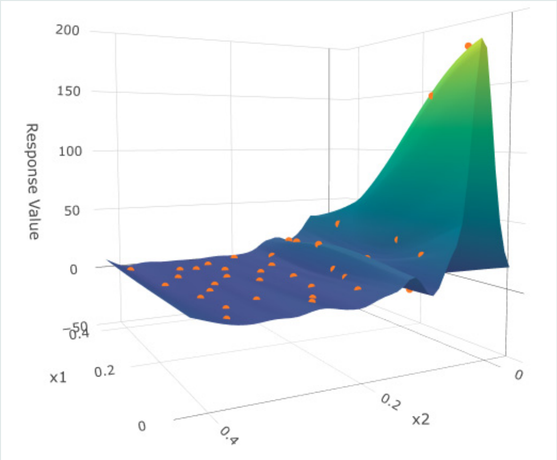
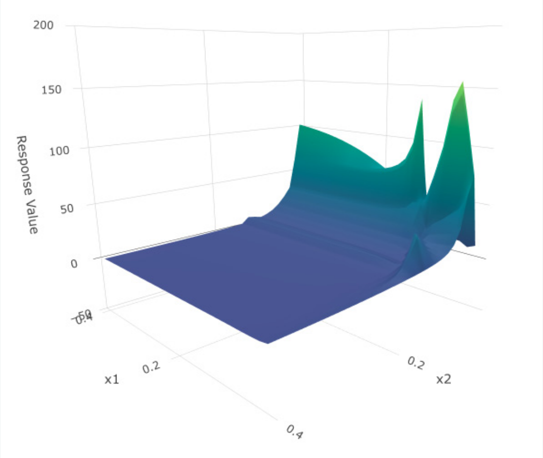
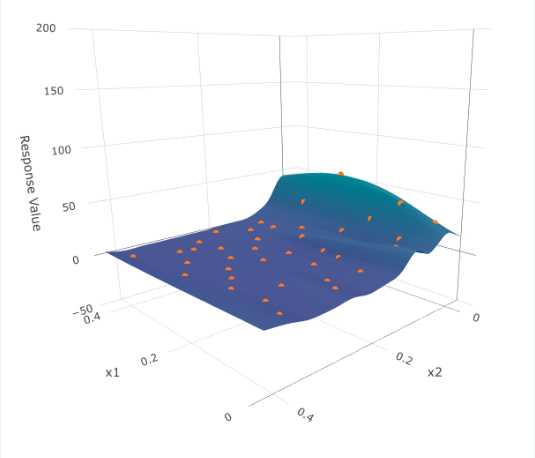
No steps are required on this page. The information is meant to elaborate on the configurations made to the surrogate mode.

The Gaussian process models are typically not visible. Since this example involves 2 variables, a 3D plot of the Gaussian process models may be compared with the true response surfaces for the peak acoustic pressures. The plots of the response surfaces were created with the Prediction Analysis web app.

A. True Response Surfaces
    A. 900 equally spaced points and MSC Nastran evaluations to acquire the resposnes were used to generate response surfaces

B. Surrogate Model
    A. An LHS of size 40, or 40 MSC Nastran runs, are used to train Gaussian process (kriging) models

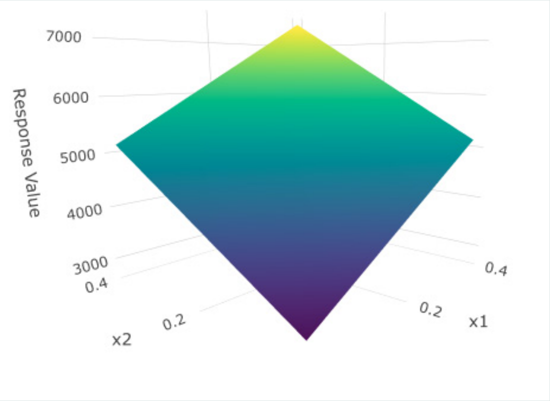
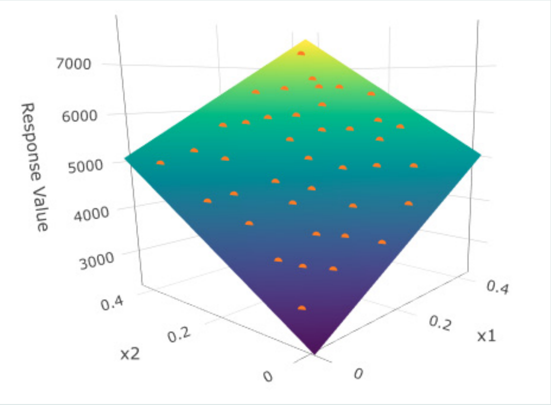
For most responses, no surrogate model will match the true response surface 100%. The goal is to construct a surrogate model with sufficient accuracy such that an optimizer can rely on surrogate models alone to perform the optimization.

Response	True Response Surface (A)	Surrogate Model (B) Approximation of True Response Surface
r1		
r2		

Surrogate Models

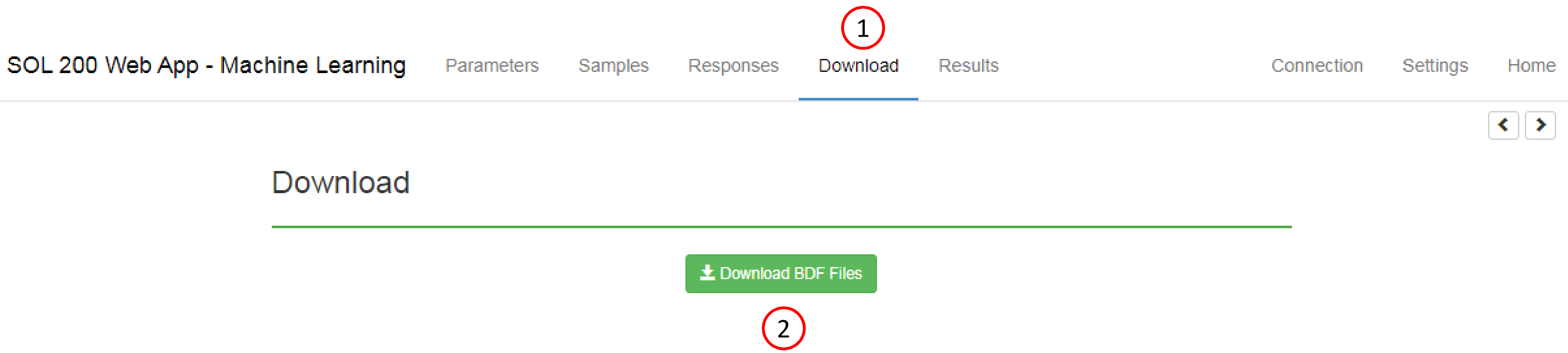
No steps are required on this page. The information is meant to elaborate on the configurations made to the surrogate mode.

The true response surface of the weight response is linear and is accurately modeled by the surrogate model.

Response	True Response Surface	Surrogate Model Approximation of True Response Surface
r3		

Download

1. Click Download
2. Click Download BDF Files

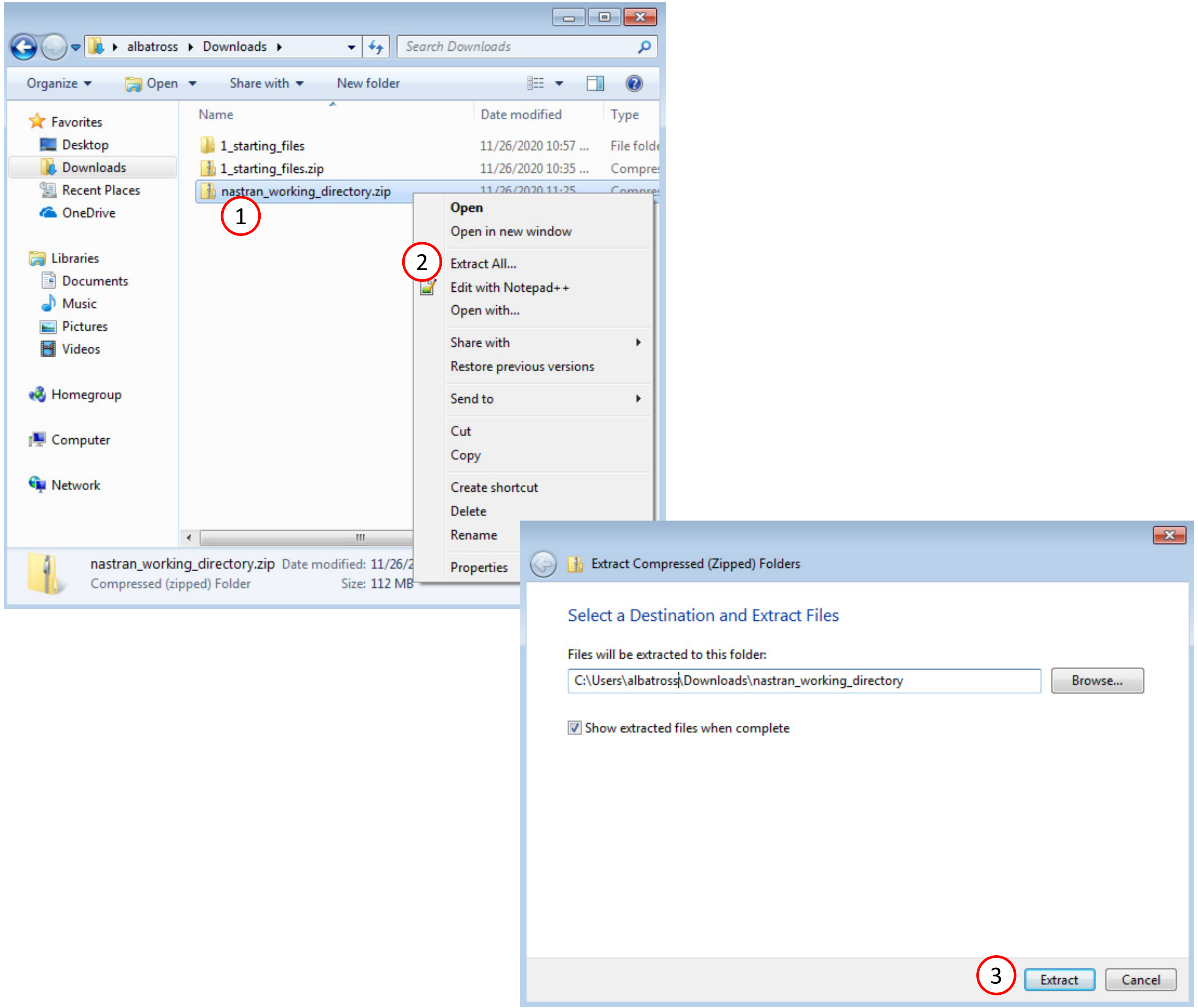


Start MSC Nastran

A new .zip file has been downloaded

1. Right click on the file
2. Click Extract All
3. Click Extract on the following window

- Always extract the contents of the ZIP file to a new, empty folder.



Start Desktop App

1. Inside of the new folder, double click on Start Desktop App
2. Click Open, Run or Allow Access on any subsequent windows
3. The Desktop App will now start

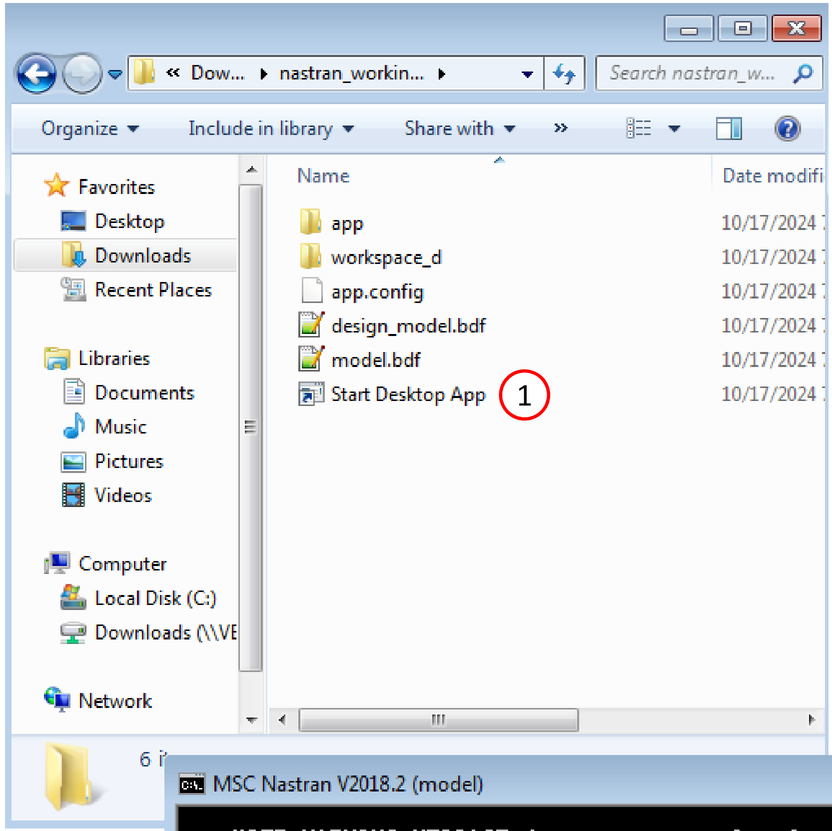
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click "Start Desktop App" to display the results.

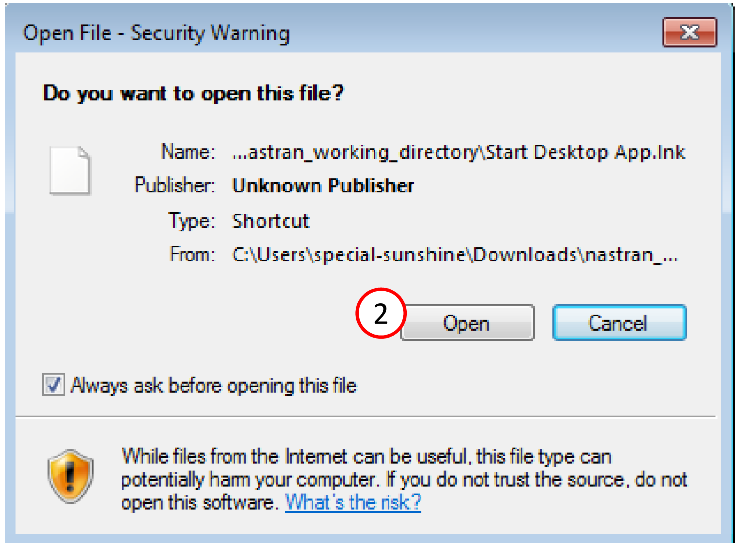
Using Linux?

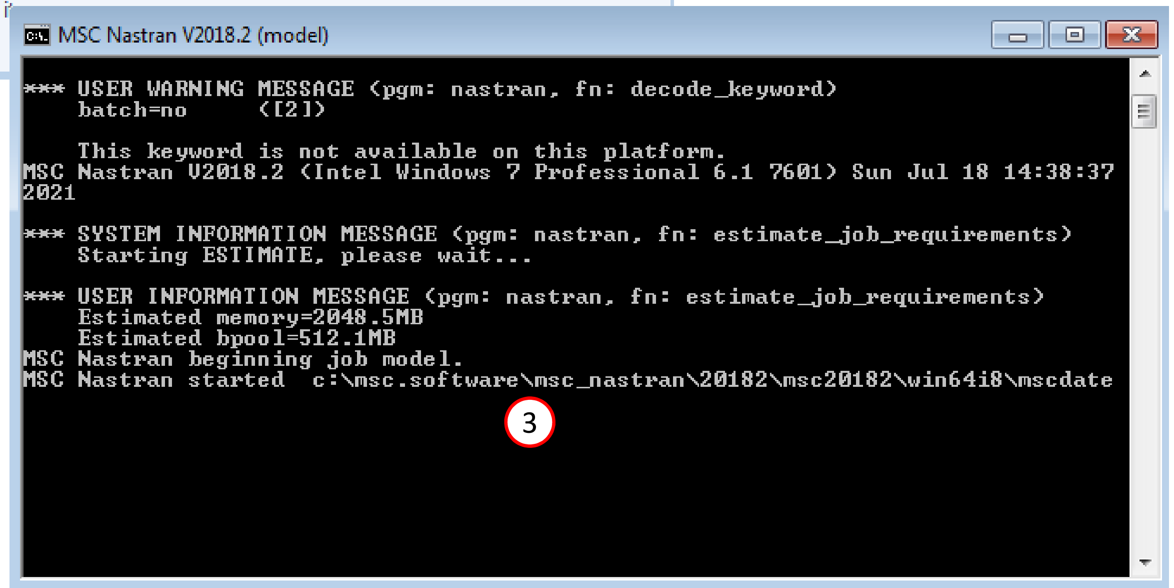
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

In some instances, execute permission must be granted to the directory. Use this command. This command assumes you are one folder level up.

sudo chmod -R u+x ./nastran_working_directory







Status

- While MSC Nastran is running, a status page will show the current state of MSC Nastran

SOL 200 Web App - Status

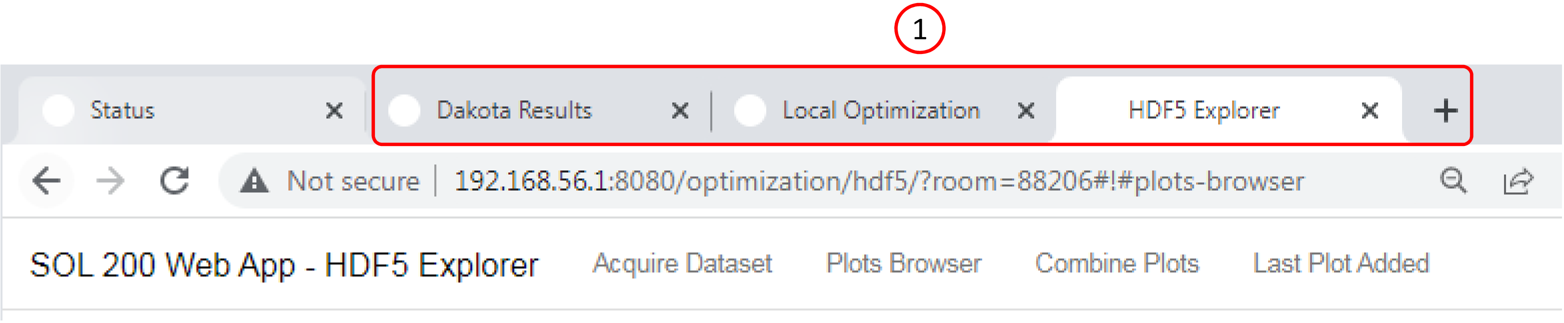
Python MSC Nastran

Status

Name	Status of Job	Design Cycle	RUN TERMINATED DUE TO
model.bdf	Running	None	

OUU Completion

1. The OUU is complete when the indicated web apps are opened.



OUU Results

1. Select the window or tab that displays the Local Optimization Results web app. This web app displays the OUU history for the objective, constraints and variables.
2. Note that the start of the optimization, the normalized constraint is very high and positive, indicating the initial design was infeasible. The constraint on the weight is initially violated.
3. At the end of the optimization, the normalized constraint is negative. Negative or near zero constraint values indicate a feasible design. This optimization has converged to a feasible design.
4. Throughout this optimization under uncertainty, the objective was continuously minimized. Recall the objective was to improve the robustness of the design, i.e. minimize the total sum of the mean and 3 standard deviations of responses r1 and r2.

OUU Results

1. Navigate to section Design Variables
2. The mean thickness of PSHELL 4 and 5 has been adjusted to satisfy the constraints and minimize the objective response.

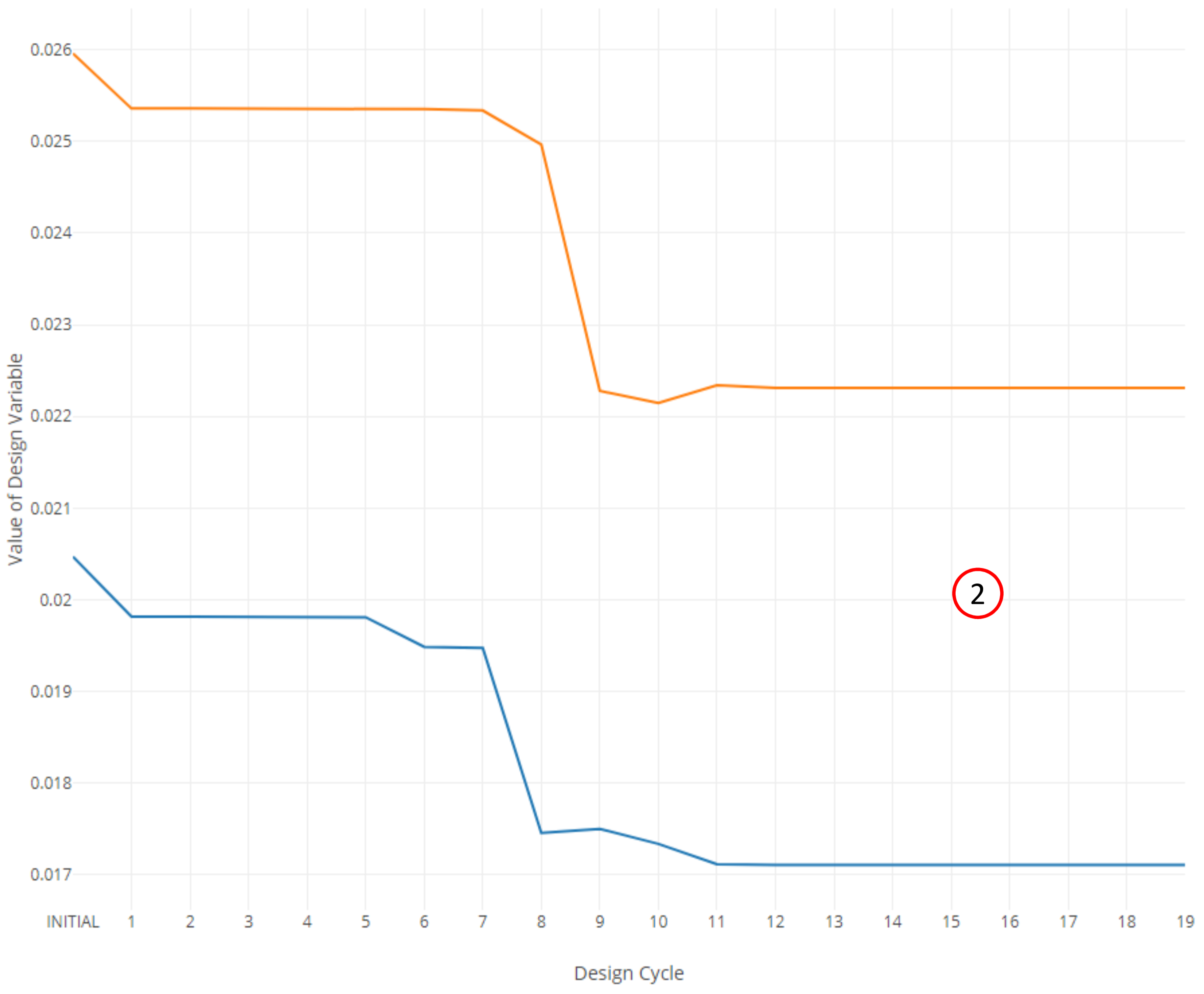
Design Variables (1)

+ Options

Reset Table | Display None | Display All

Display	Color	Label
		Search
☑	blue	x1_mean
☑	orange	x2_mean

5 | 10 | 20 | 50 | 100 | 200

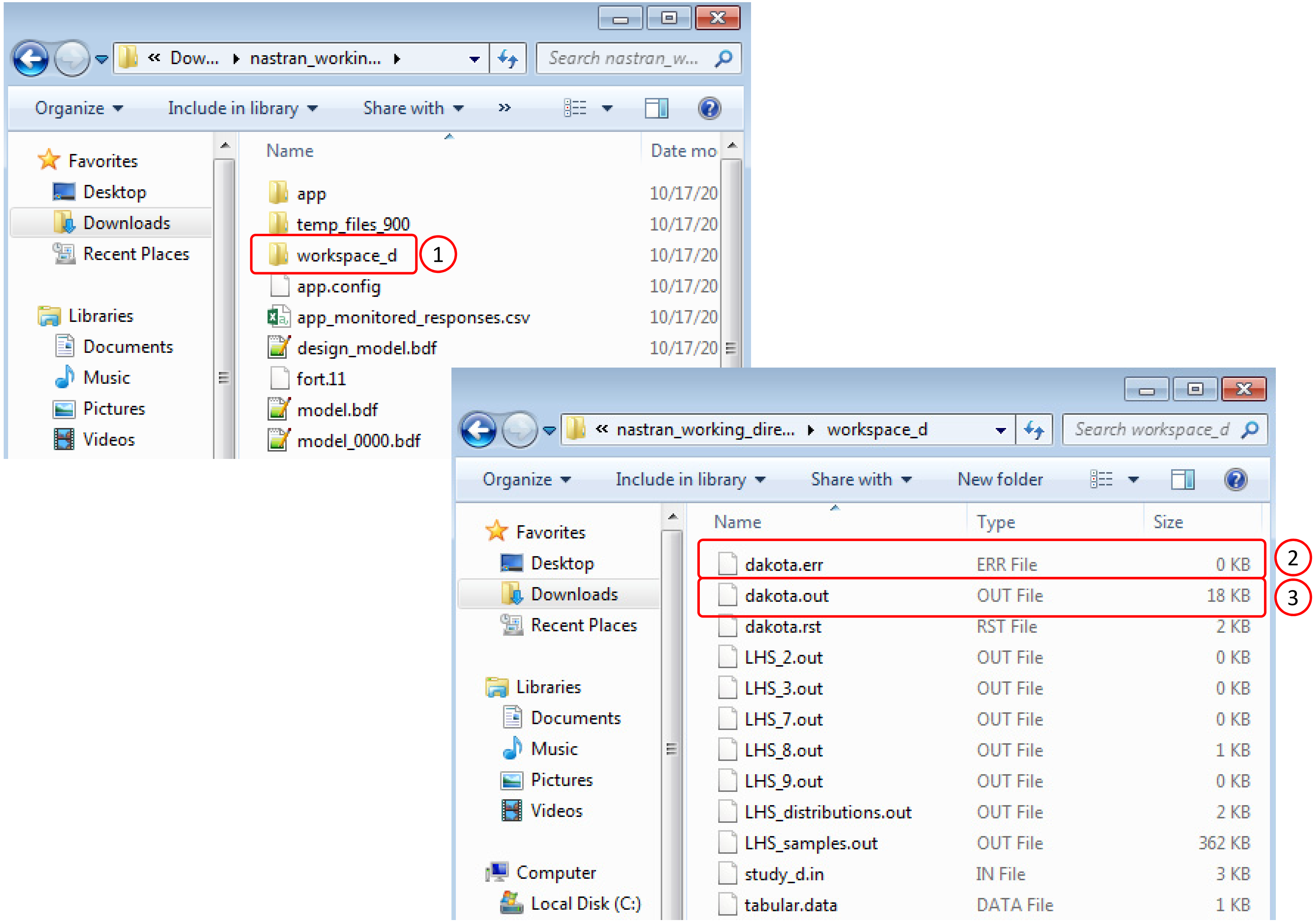


OUU Results

1. The results of the OUU are contained in the workspace_d directory

2. If there were any errors during the OUU, the errors are typically stored in the file dakota.err. Warnings in this file may be ignored. Notice in this example, the size of the file is 0KB, indicating the file is empty of error and warning messages.

3. The output of Dakota is contained in file dakota.out. Open this file in a text editor.



OUU Results

1. Once file dakota.out is opened in a text editor, scroll to the very end of the file and you will find the results of the OUU.
2. The optimal mean values for x1 and x2 are listed.
3. The objective at the optimum is displayed.
4. The constraint value at the optimum is displayed, which was a probability of failure. There is a 0% probability of exceeding the weight responses' upper bound of 2910.
5. The surrogate models were constructed based on training data from 40 MSC Nastran runs.
6. During each iteration, an uncertainty quantification using the sampling method with 5000 runs was performed. Since formulation 3 was used for the OUU, it was the surrogate model that was evaluated frequently, not the black box function (FEA solver). The surrogate model was evaluated 525,000 times during the OUU.

1 6 5 2 3 4

```
<<<<< Function evaluation summary (APPROX_INTERFACE 1): 525000 total (525000 new, 0 duplicate)
<<<<< Function evaluation summary (UQ_ACTUAL): 40 total (40 new, 0 duplicate)
<<<<< Best parameters          =
                      1.7105867032e-02 x1_mean
                      2.2310604413e-02 x2_mean
<<<<< Best objective function  =
                      2.2569999218e+02
<<<<< Best constraint values   =
                      0.0000000000e+00
<<<<< Best evaluation ID not available
(This warning may occur when the best iterate is comprised of multiple interface
evaluations or arises from a composite, surrogate, or transformation model.)


<<<<< Iterator conmin_mfd completed.
<<<<< Environment execution completed.
DAKOTA execution time in seconds:
  Total CPU        =     55.0279 [parent =     55.0279, child =      7e-06]
  Total wall clock =     134.503
```

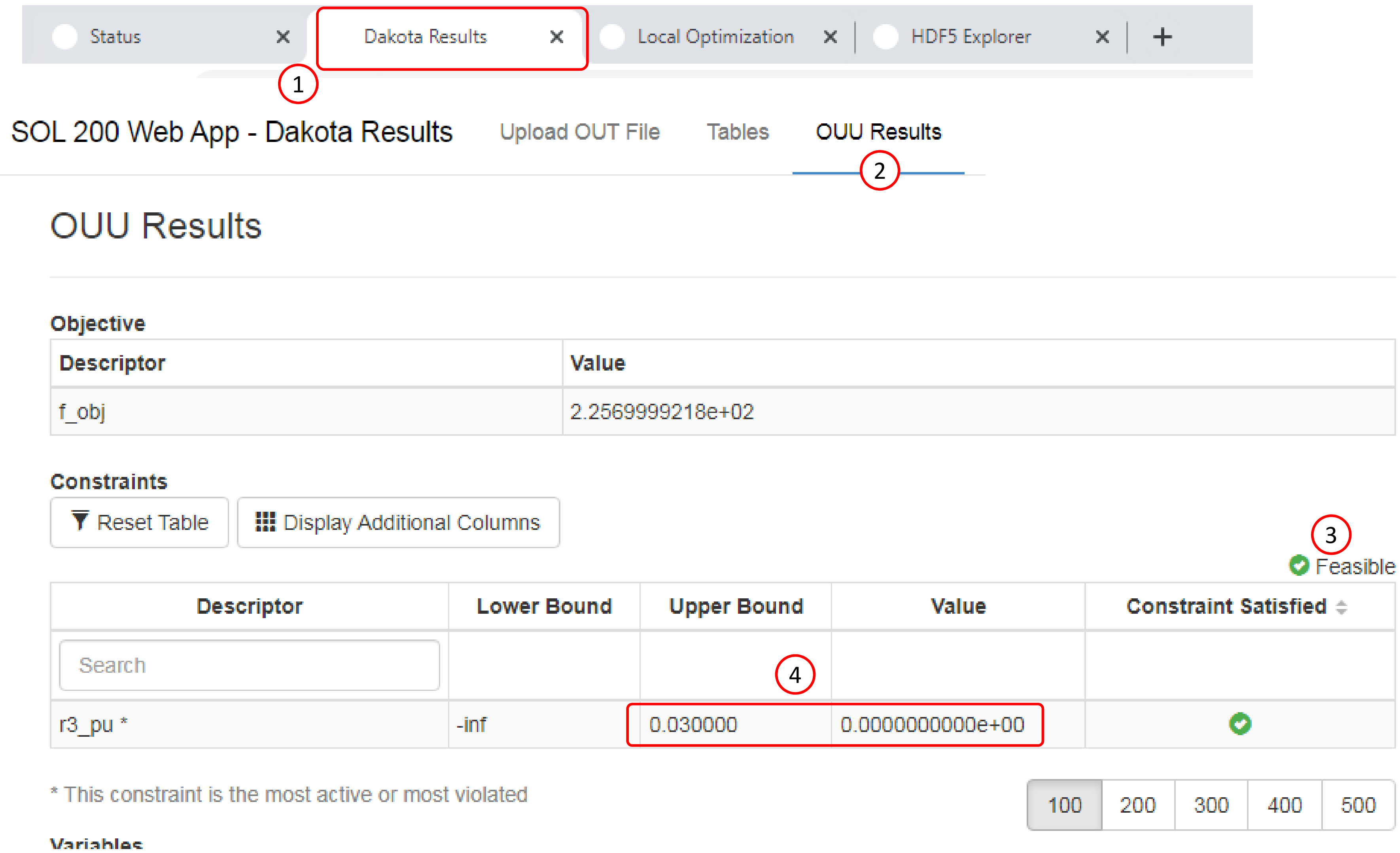
Discussion of Final Probabilities of Failure

The same results discussed on the previous page may be inspected in the web app.

1. Select the Dakota Results tab or window
2. Click OUU Results
3. Notice the final design is deemed feasible
4. On close inspection, it is shown that the probability of failure of 0.0 (0.0%) is less than the upper bound of 0.3 (3%).

Since the objective was minimize and the constraints are satisfied, the OUU has been a success so far.

One drawback to using formulation 3 is that the optimization solution is based on approximations of the response functions. The optimal solution must be confirmed to actually satisfy the constraints. The confirmation is done in part B.

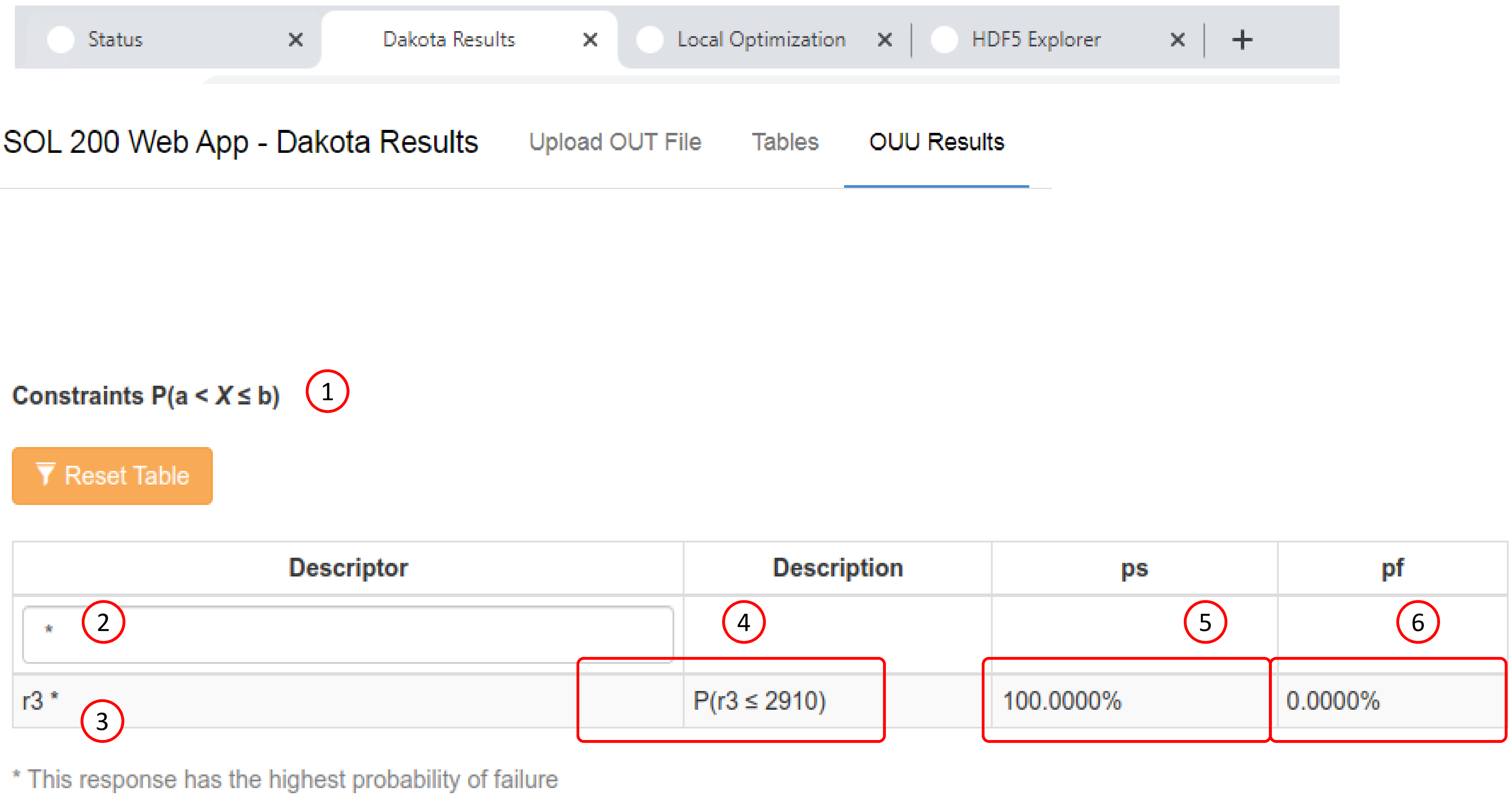


Discussion of Final Probabilities of Failure

The table on the previous page displayed separate probabilities of failure for the bounds, i.e. P(a < X) and P(X <b). There is a desire to know the combined probability P(a < X ≤ b). The probability for P(a < X ≤ b) is available by following these steps.

1. Navigate to section Constraints P(a < X ≤ b)
2. In the indicated search bar, search for character *
3. The search reveals responses that have the highest probability of failure.
4. The Description column displays the probabilities now consider both the lower and upper bound, i.e. P(a < X ≤ b).
5. The probability of survival P(a < X ≤ b) is displayed in column ps.
6. The probability of survival P(a > X OR b < X) is displayed in column pf.

The highest probability of failure is 0.0000%.



Part B – Verification of Robust Design Optimization Solution

Motivation

Part A - A robust design optimization was performed to improve the robustness of the design.

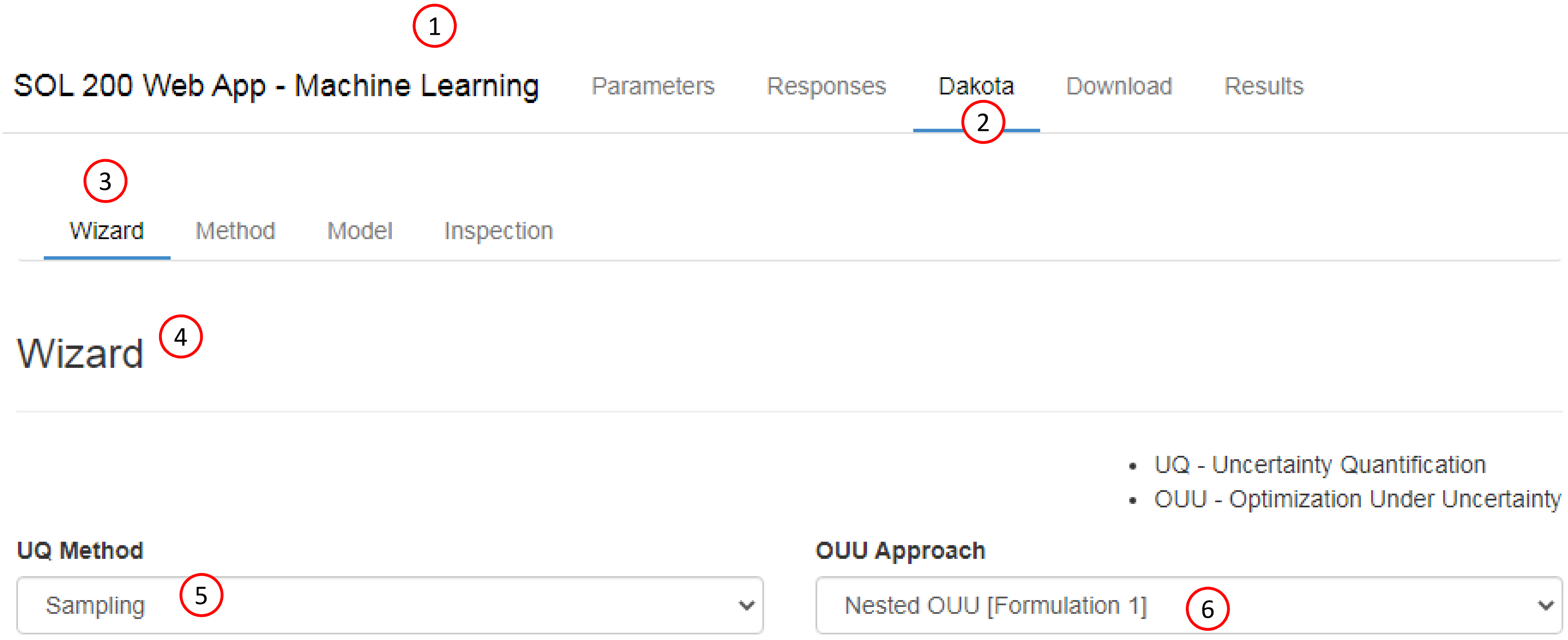
Part B - An LHS of size 50 (50 MSC Nastran runs) is evaluated and the tail probabilities are calculated. These tail probabilities are deemed the actual tail probabilities or actual probabilities of failure.

The approximated and actual probabilities of failure are compared to confirm the OUU solution is in fact feasible.

Uncertainty Quantification

1. Return to the Machine Learning web app
2. Click Dakota
3. Navigate to section Wizard
4. Click Wizard
5. Set UQ Method to Sampling
6. Set the OUU Approach to Nested OUU [Formulation 1]

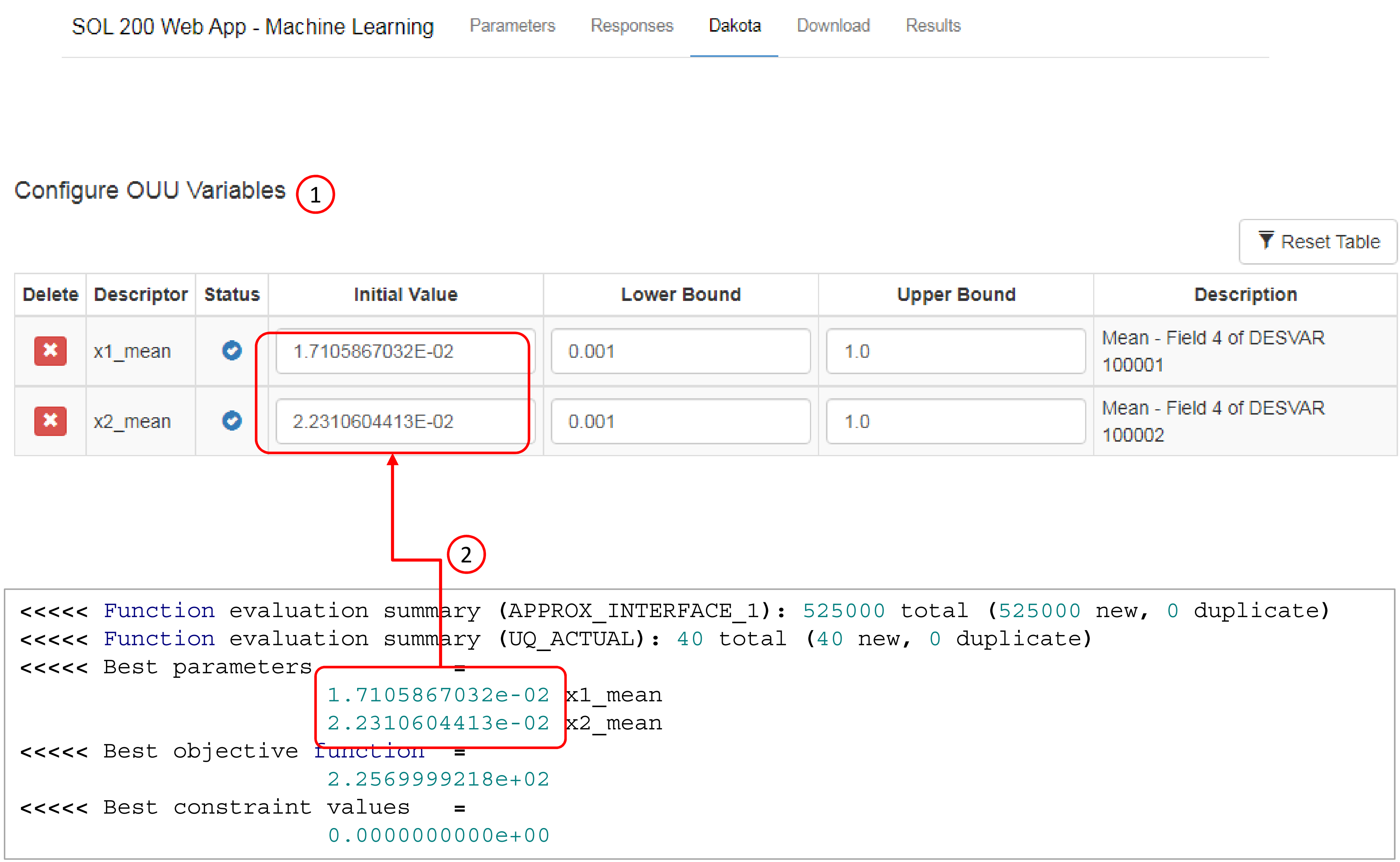
The goal is to perform an uncertainty quantification and run the optimization procedure only to compute the constraint values, i.e. probabilities of failure. Later on, max_function_evaluations is set to 1 to allow the optimization routine to calculate only constraint values and terminate with zero iterations.



Uncertainty Quantification

1. Navigate to section Configure OUU Variables
2. Take the optimal variable values after the OUU and replace the old initial values for the OUU variables.

The idea is to determine the new probabilities of failure at the optimal variable values.



Uncertainty Quantification

1. Navigate to section Configure OUU Constraints
2. Set the Probability of Failure for Upper Bound to 5

- A maximum probability of failure of 5% is used so that the constraints are relative to 5%, not 3%.

Configure OUU Constraints (1)

Statistics to compute at each response level

Probabilities

Reset Table

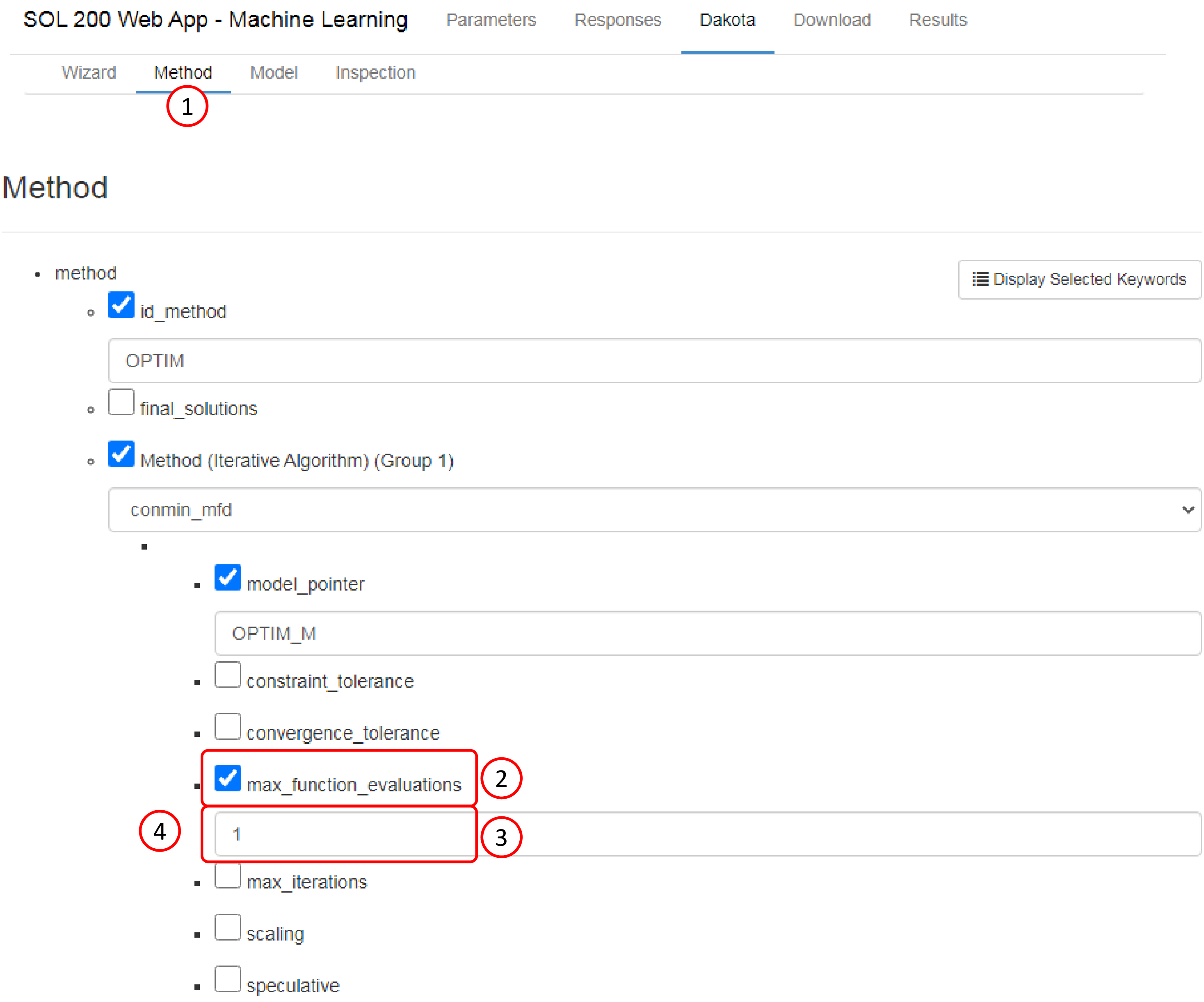
Delete	Descriptor	Status	Probability of Failure for Lower Bound [%]	Lower Bound	Upper Bound	Probability of Failure for Upper Bound [%]
✖	r1	✔				
✖	r2	✔				
✖	r3	✔			2910	5

(5)

Uncertainty Quantification

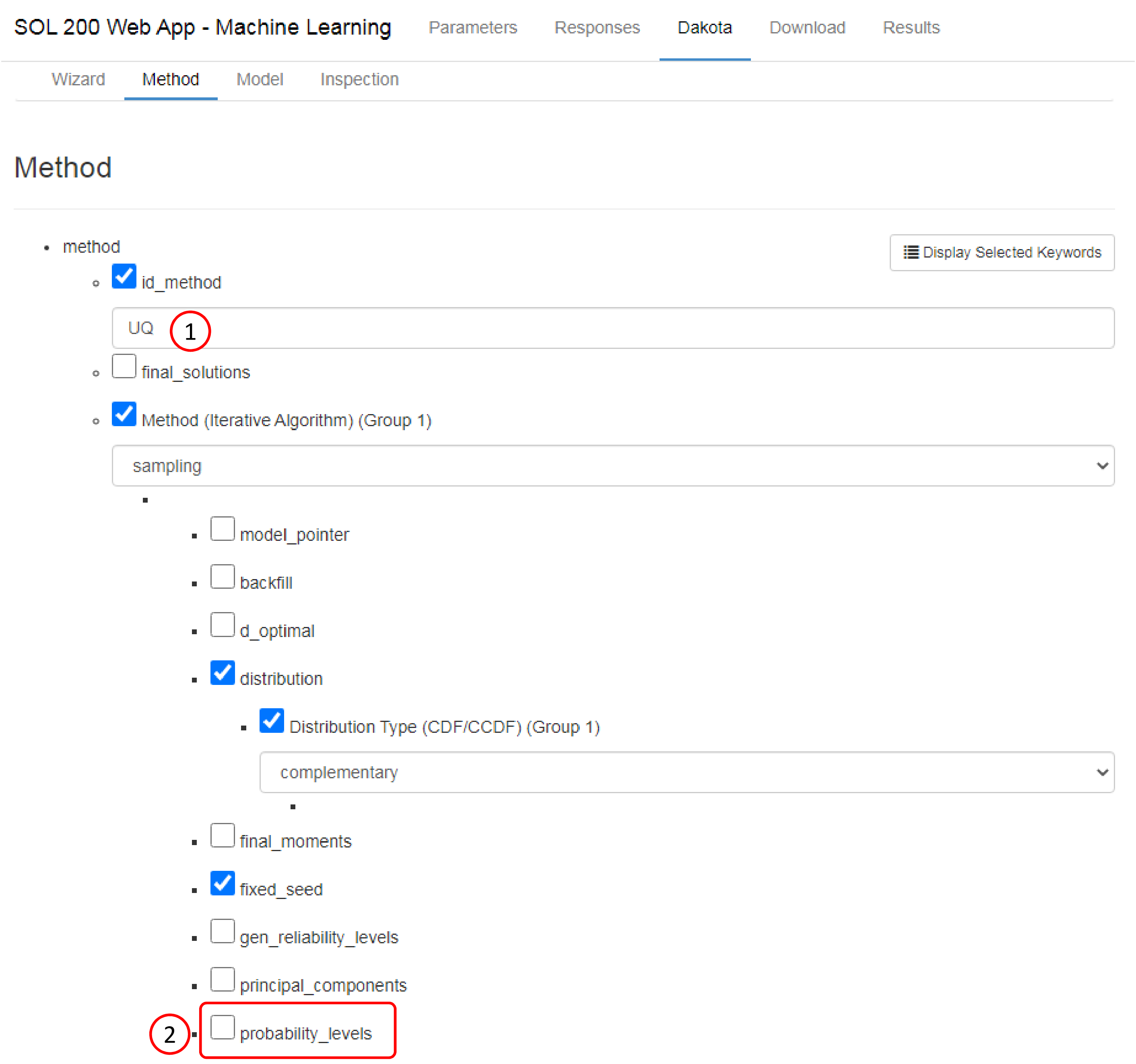
1. Click Method
2. Mark the indicated checkbox to turn on the keyword max_function_evaluations
3. Set the indicated input box to 1
4. Reminder! Ensure max_function_evaluations is set to 1. This is a step that is very easy to overlook.

- The goal is to perform an uncertainty quantification and run the optimization procedure only to compute the constraint values, i.e. probabilities of failure. The keyword max_function_evaluations is set to 1 to allow the optimization routine to calculate only constraint values and terminate with zero iterations.



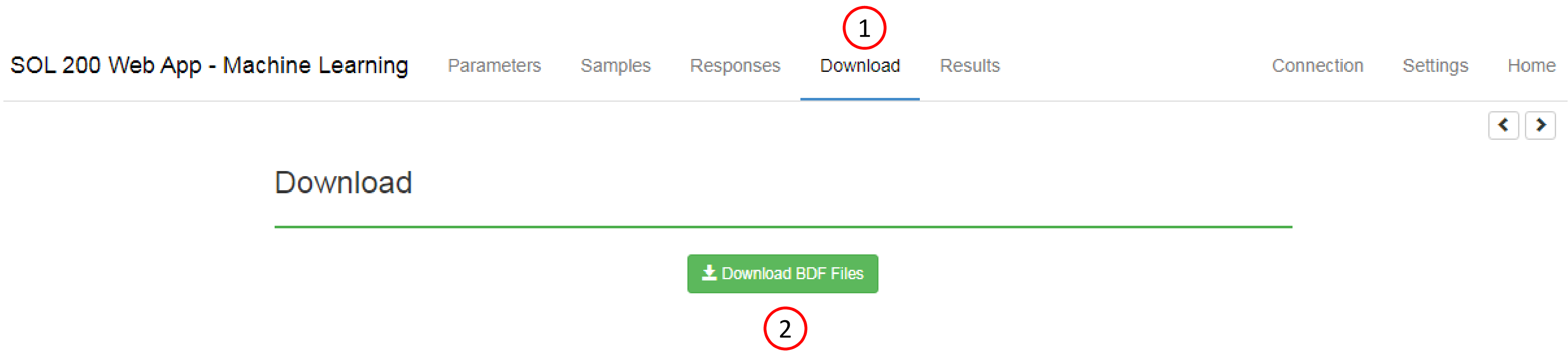
Uncertainty Quantification

1. Scroll to the method keyword block with id_method=UQ
2. Deselect the checkbox for probability_levels



Download

1. Click Download
2. Click Download BDF Files

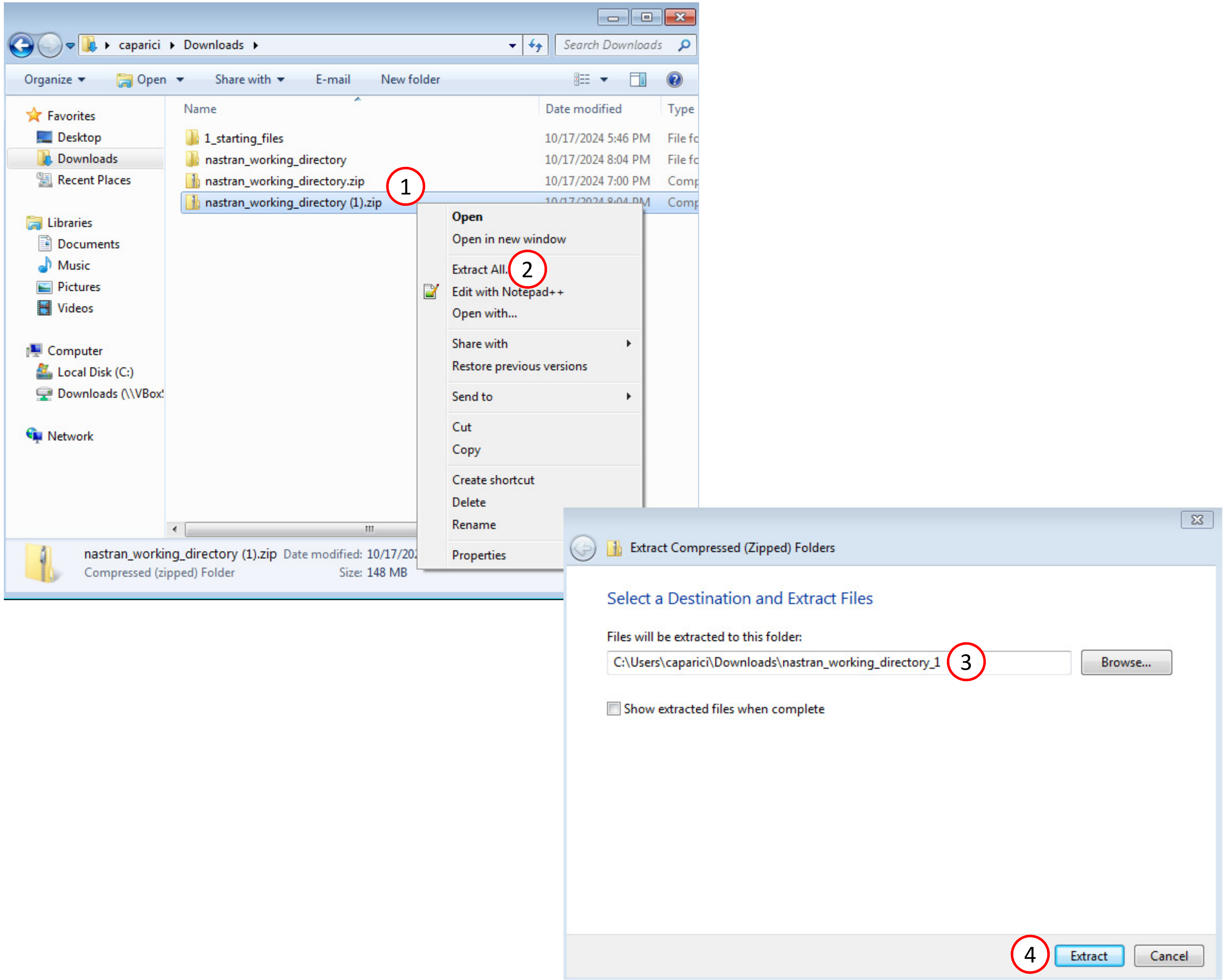


Start MSC Nastran

A new .zip file has been downloaded

1. Right click on the file
2. Click Extract All
3. It is good practice to avoid special characters and spaces in paths, directory names and file names. Name the final directory: nastran_working_directory_1.
4. Click Extract on the following window

- Always extract the contents of the ZIP file to a new, empty folder.



Start Desktop App

1. Inside of the new folder, double click on Start Desktop App
2. Click Open, Run or Allow Access on any subsequent windows
3. The Desktop App will now start

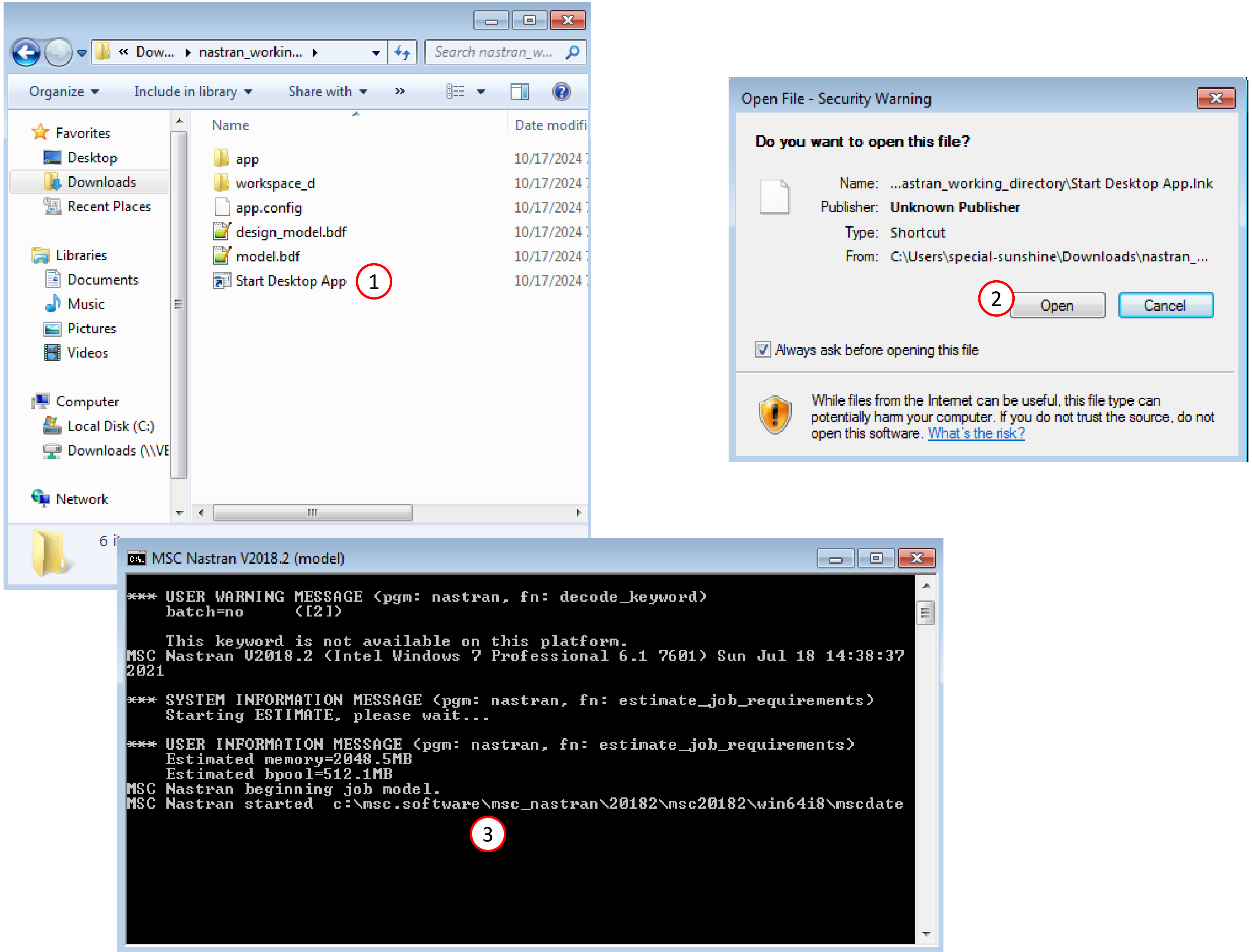
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click “Start Desktop App” to display the results.

Using Linux?

Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
 cd ./nastran_working_directory
3) Use this command to start the process
 ./Start_MSC_Nastran.sh

In some instances, execute permission must be granted to the directory. Use this command. This command assumes you are one folder level up.

sudo chmod -R u+x ./nastran_working_directory



Status

- While MSC Nastran is running, a status page will show the current state of MSC Nastran

SOL 200 Web App - Status

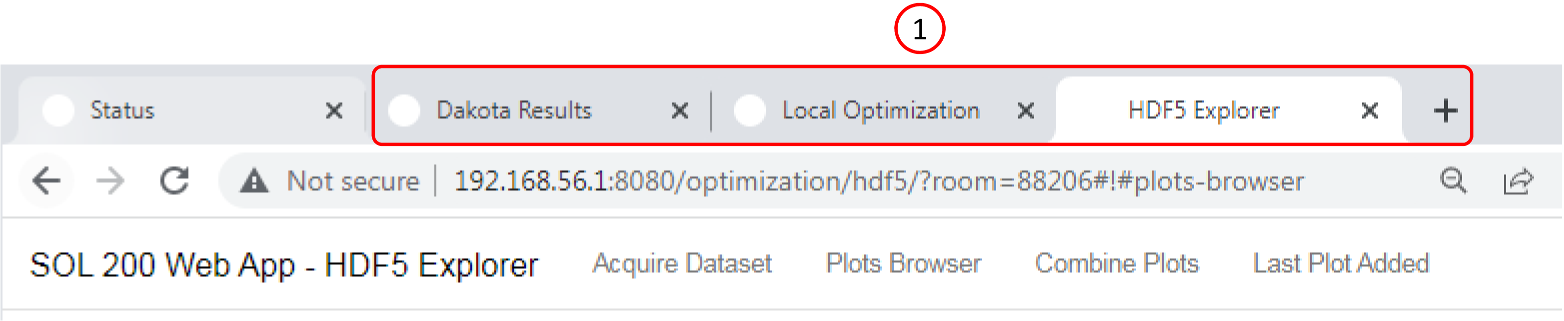
Python MSC Nastran

Status

Name	Status of Job	Design Cycle	RUN TERMINATED DUE TO
model.bdf	Running	None	

Completion

1. The process is complete when the indicated web apps are opened.



No Optimization

1. Open file dakota.out in a text editor. Scroll to the very end of the file and you will find the results.
2. An LHS of size 50 (50 MSC Nastran runs) was evaluated to determine the probabilities
3. Since the keyword max_function_evaluations was set to 1, the optimizer terminates after all 50 runs are complete and zero optimization iterations are performed. Recall the goal is to just run the optimization procedure to calculate the constraint values.

```
------------------------------
Begin     UQ_I Evaluation   50
------------------------------
Parameters for evaluation 50:
                      1.7945510952e-02 x1
                      2.3410787140e-02 x2
blocking fork
Active response data for UQ_I evaluation 50:
Active set vector = { 1 1 1 }
                      1.3263327000e+02 r1
                      2.8299534000e+01 r2
                      2.8715615000e+03 r3
Active response data from sub_iterator:
Active set vector = { 1 1 1 1 0 0 1 }
                      1.2025857498e+02 mean_r1
                      1.3528718043e+01 std_dev_r1
                      3.2386400460e+01 mean_r2
                      5.1314392890e+00 std_dev_r2
                      0.0000000000e+00 ccdf_plev1_r3
---------------------------
NestedModel Evaluation    1 results:
---------------------------
Active response data from nested mapping:
Active set vector = { 1 1 }
                      2.0862544743e+02 f_obj
                      0.0000000000e+00 r3_pu

Iteration terminated: max_function_evaluations limit has been met.
<<<<< Function evaluation summary (UQ_I): 50 total (50 new, 0 duplicate)
<<<<< Best parameters          =
                      1.7105867032e-02 x1_mean
                      2.2310604413e-02 x2_mean
<<<<< Best objective function  =
                      2.0862544743e+02
<<<<< Best constraint values   =
                      0.0000000000e+00
```

Results

1. The mean and standard deviation of responses r1 and r2 are listed. Record these values for later use.

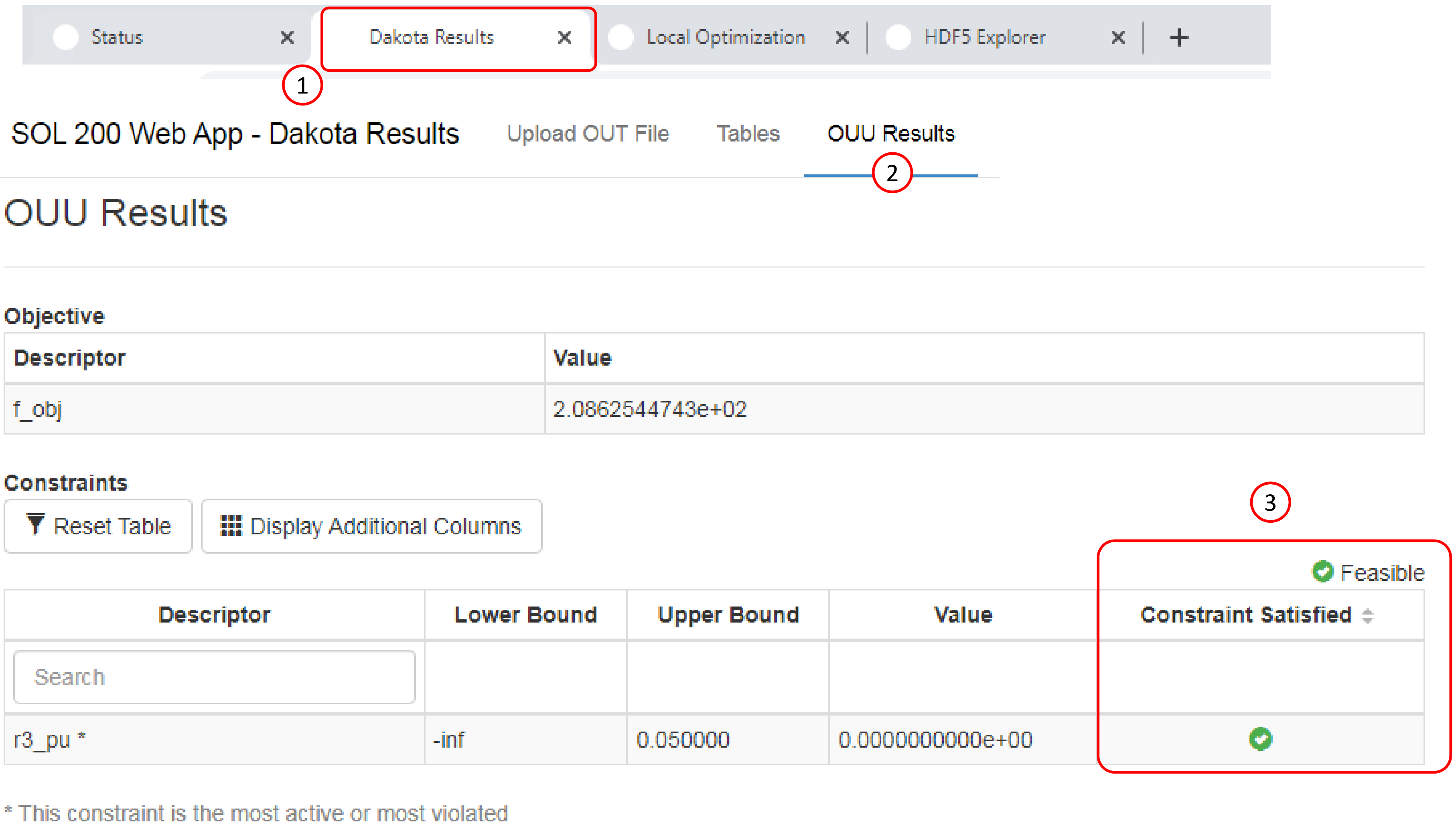
```
-----------------------------
Begin     UQ_I Evaluation   50
-----------------------------
Parameters for evaluation 50:
                      1.7945510952e-02 x1
                      2.3410787140e-02 x2
blocking fork
Active response data for UQ_I evaluation 50:
Active set vector = { 1 1 1 }
                      1.3263327000e+02 r1
                      2.8299534000e+01 r2
                      2.8715615000e+03 r3
Active response data from sub_iterator:
Active set vector = { 1 1 1 1 0 0 1 }
                      1.2025857498e+02 mean_r1
                      1.3528718043e+01 std_dev_r1
                      3.2386400460e+01 mean_r2
                      5.1314392890e+00 std_dev_r2
                      0.0000000000e+00 ccdf_plev1_r3
---------------------------
NestedModel Evaluation    1 results:
---------------------------
Active response data from nested mapping:
Active set vector = { 1 1 }
                      2.0862544743e+02 f_obj
                      0.0000000000e+00 r3_pu

Iteration terminated: max_function_evaluations limit has been met.
<<<<< Function evaluation summary (UQ_I): 50 total (50 new, 0 duplicate)
<<<<< Best parameters          =
                      1.7105867032e-02 x1_mean
                      2.2310604413e-02 x2_mean
<<<<< Best objective function  =
                      2.0862544743e+02
<<<<< Best constraint values   =
                      0.0000000000e+00
```

1

Discussion of Final Probabilities of Failure

The same results discussed on the previous page may be inspected in the web app.

1. Select the Dakota Results tab or window
2. Click OUU Results
3. Notice the final design is deemed feasible and all the individual constraints are satisfied



Comparison of Approximate and Actual p_f

It nearly all cases, surrogate models always have some error in reflecting the true response function. In part B, a UQ is performed to determine the actual objective and constraint values. When comparing the approximate results form part A with the actual results from part B, there are slight differences in the objective and constraints, as expected. More importantly, the solution is confirmed to be feasible.

Response	Part A – OUU Formulation 3 (Surrogate Model)	Part B - UQ Generated
Comments	The objective and constraint values are based on a surrogate model, which is an approximation of the true response function	These are the values after an LHS of size 50. These are deemed the actual or true objective and constraint values
Objective	2.2569999218e+02	2.0862544743e+02
Constraint	0.0000000000e+00	0.0000000000e+00

Comparison of Initial and Final Spread

Recall that the goal was to improve the robustness of the design for responses r1 and r2. This was expressed by minimizing this objective function

$$1 * r_{1,mean} + 3 * r_{1,standard\ deviation} + 1 * r_{2,mean} + 3 * r_{2,standard\ deviation}$$

Note the mean and standard deviation for responses r1 and r2 are lower after optimization. This robust design optimization has been a success.

When this tutorial was first made, the FINAL solution shown in table 1 was obtained. When this tutorial is repeated with a different operating system, a setting is slightly different, or a different version of Sandia Dakota or MSC Nastran is used, a different OUU solution may be obtained. Table 2 displays the OUU solution when this tutorial was repeated and is a valid. The first and second solution are both valid because the final designs is more robust that the initial design, i.e. the standard deviations are reduced.

Dynamics response surfaces, such as acoustic pressure, are notoriously non-smooth and characterized by numerous minimums and maximums. Multiple optimization solutions will be found and is expected.

On the next page, values from table 1 are used.

Table 1 - Solution 1

	Mean	Standard Deviation
INITIAL (x1_mean, x2_mean) = (.02047, .02596)		
r1	2.1090780780e+02	4.2544486407e+01
r2	4.9869224760e+01	1.4092696547e+01
FINAL (x1_mean, x2_mean) = (1.71058e-02, 2.23106e-02)		
r1	1.2025857498e+02	1.3528718043e+01
r2	3.2386400460e+01	5.1314392890e+00

Table 2 - Solution 2

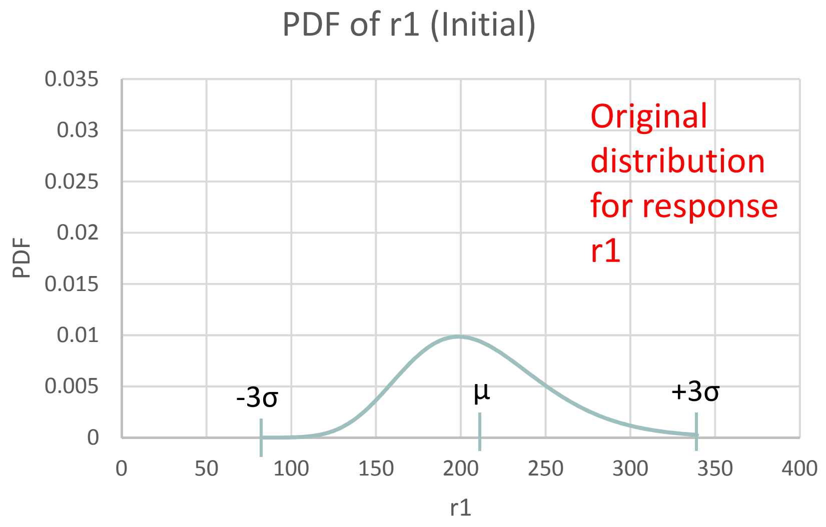
	Mean	Standard Deviation
FINAL (x1_mean, x2_mean) = (1.8177076969e-02, 2.1756058571e-02)		
r1	1.1443455139e+02	1.3366511477e+01
r2	3.1635479161e+01	5.2464148187e+00

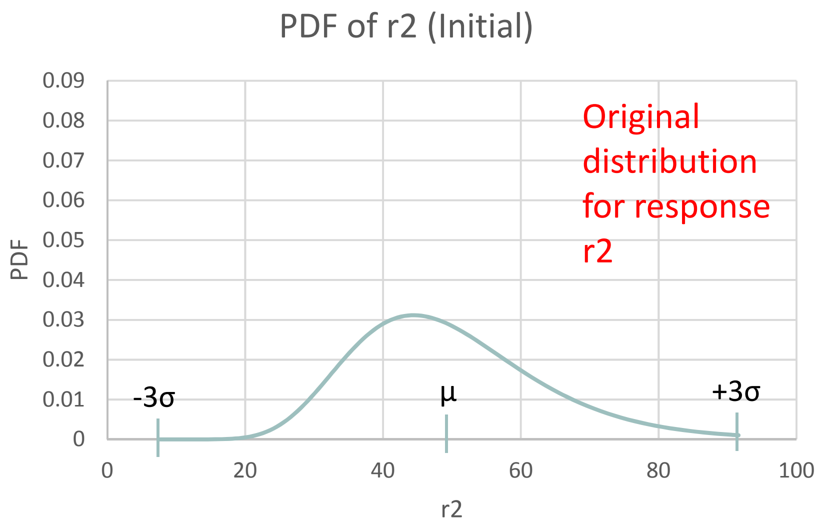
Comparison of Initial and Final Spread

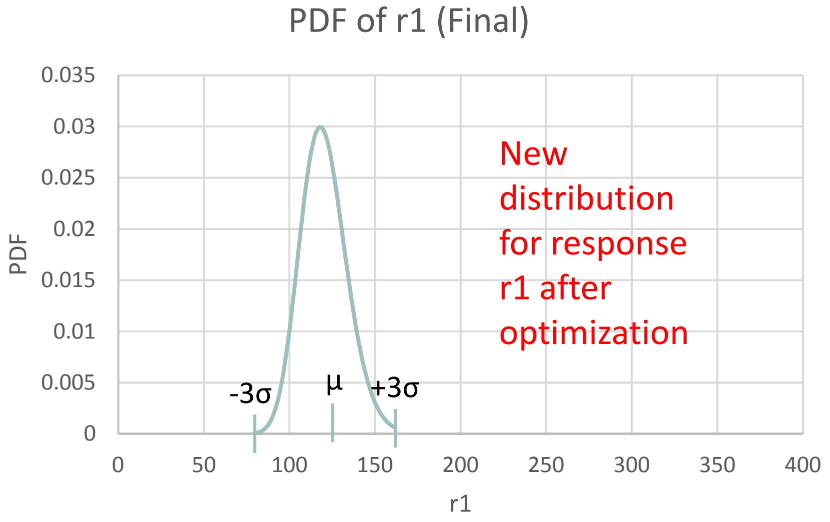
The initial and final distributions are plotted. It is assumed the distributions are lognormal.

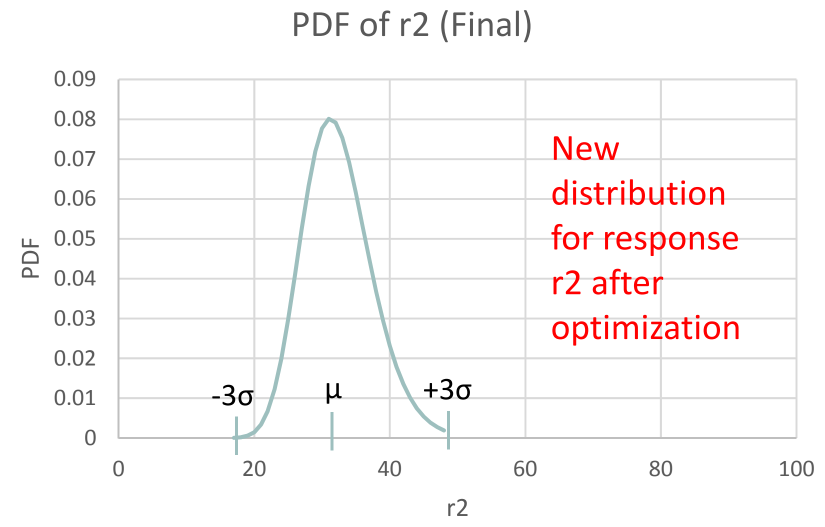
The initial and final means and standard deviations are used to create lognormal PDF plots.

It is visually confirmed the final spreads have been minimized. The robust design optimization has been a success.









End of Tutorial

Appendix

Appendix Contents

Interpreting the Dakota Input File

The Dakota input file has a distinct format that is not like the MSC Nastran bulk data file format. The following pages describe the meaning of some of the Dakota keywords such as primary_response_mapping, secondary_response_mapping, etc.

study_d.in

```
model
   id_model 'OPTIM_M'
   responses_pointer 'OPTIM_R'
   variables_pointer 'OPTIM_V'
      nested
         sub_method_pointer 'UQ'
            primary_response_mapping  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
            secondary_response_mapping
              0.  0.  0.  0.  1.  0.  0.  0.  0.  0.
              0.  0.  0.  0.  0.  1.  0.  0.  0.  0.
              0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
              0.  0.  0.  0.  0.  0.  0.  0.  0.  1.
            primary_variable_mapping  'x1'  'x2'
            secondary_variable_mapping  'mean'  'mean'


method
   id_method 'UQ'
      sampling
         model_pointer 'UQ_M'
         distribution
               complementary
         response_levels  -20000  20000  -20000  20000
            num_response_levels  0  2  2
         sample_type
               lhs
         samples 5000
         seed 12347
```

Interpreting the Dakota Input File

- The interface keyword is used to define the executable of a black box function. In this exercise, the analysis_drivers keyword points to an executable called desktop_app_a. This executable runs MSC Nastran automatically whenever parameter inputs xi are provided and returns responses ri.
- Analysis drivers are by far the costliest component to develop when implementing uncertainty quantification or optimization under uncertainty, and often require weeks of development to construct analysis drivers. The SOL 200 Web App includes a run ready analysis driver for MSC Nastran and saves substantial development time.

```
interface
   id_interface 'UQ_ACTUAL'
   analysis_drivers  './app/desktop_app_a --analysis_driver_dakota'
         fork
```

Interpreting the Dakota Input File

1. The responses keyword is used to define the responses output by the black box function. From what is defined, the black box function returns 3 responses, zero gradients and zero hessians. To help differentiate the responses, descriptors r1, r2 and r3 are used for the 3 responses.

2. Notice the sampling method is defined, which is a method used for uncertainty quantification.

3. Since the distribution is set to complementary, the tail probabilities outputted will be complementary cumulative distribution function (CCDF) values. Alternatively, cumulative may be used. In this exercise, it is assumed complementary is used throughout.

4. The response_levels keyword is used to specify the values for which probabilities are requested. Notice the bound values of -20000 and 20000 are used.

5. The num_response_levels keyword is used to map the response levels to each response. In this example, the num_response_levels ‘0 2 2’ is read as follows: The first zero response levels are associated with response r1, the next 2 response levels are associated with r2, and the next 2 response levels are associated with r3. Response r1 is the weight, and r2 and r3 are the stress responses. Probabilities are requested for only the stress responses r2 and r3, not r1.

6. Latin hypercube sampling (LHS) is used with size 5,000 samples. LHS employs a random number generator. Random number generators are algorithms, and if certain initial conditions are defined, the random number generator will repeatedly output the same number. The seed is used as an initial condition that helps replicate the same LHS. The seed can be any positive integer and will generate the same LHS values for the same seed value.

```
method
   id_method 'UQ'
(2)   sampling
      model_pointer 'UQ_M'
      distribution
            complementary  (3)                           (4)
      response_levels  -20000  20000  -20000  20000
         num_response_levels  0  2  2
      sample_type                                        (5)
            lhs
      samples 5000      (6)
      seed 12347

responses
   id_responses 'UQ_R'
   descriptors  'r1'  'r2'  'r3'  (1)
      no_gradients
      no_hessians
      response_functions 3
```

Interpreting the Dakota Input File

1. The keywords primary_response_mapping and secondary_response_mapping keywords are the most confounding for new users and are explained next.

2. When a UQ method is employed, e.g. sampling, local_reliability, etc. each response will output a mean and standard deviation (2 outputs). If N response_levels were defined for response ri, N additional outputs are available. In this example, r1 outputs a mean and standard deviation. Response r2 outputs a mean, standard deviation and 2 probabilities. Response r3 also outputs a mean, standard deviation and 2 probabilities. For this example, there are a total of 10 statistical quantities and are stored in the indicated column vector.

```
model
   id_model 'OPTIM_M'
   responses_pointer 'OPTIM_R'
   variables_pointer 'OPTIM_V'
      nested
         sub_method_pointer 'UQ'
            primary_response_mapping   1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
            secondary_response_mapping
              0.  0.  0.  0.  1.  0.  0.  0.  0.  0.
              0.  0.  0.  0.  0.  1.  0.  0.  0.  0.
              0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
              0.  0.  0.  0.  0.  0.  0.  0.  0.  1.
            primary_variable_mapping   'x1'  'x2'
            secondary_variable_mapping  'mean'  'mean'

method
   id_method 'UQ'
      sampling
         model_pointer 'UQ_M'
         distribution
               complementary
         response_levels  -20000  20000  -20000  20000
            num_response_levels  0  2  2
         sample_type
               lhs
         samples 5000
         seed 12347

responses
   id_responses 'UQ_R'
   descriptors  'r1'  'r2'  'r3'
      no_gradients
      no_hessians
      response_functions 3
```

(1) refers to the primary_response_mapping and secondary_response_mapping keywords (boxed).

(2)

$$
\begin{bmatrix} r1_{mean} \\ r1_{std} \\ r2_{mean} \\ r2_{std} \\ r2_{pl} \\ r2_{pu} \\ r3_{mean} \\ r3_{std} \\ r3_{pl} \\ r3_{pu} \end{bmatrix}
\begin{matrix} \\ \\ \\ \\ P(-20000 < r2) \\ P(\ \ 20000 < r2) \\ \\ \\ P(-20000 < r3) \\ P(\ \ 20000 < r3) \end{matrix}
$$

Interpreting the Dakota Input File

Keywords primary_response_mapping and secondary_response_mapping define matrices. The product of these matrices and the column vector define the objective and constraint responses.

```
primary_response_mapping
  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
secondary_response_mapping
  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.
```

primary_response_mapping

$$Objective\ Response = [1.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.] * \begin{bmatrix} r1_{mean} \\ r1_{std} \\ r2_{mean} \\ r2_{std} \\ r2_{pl} \\ r2_{pu} \\ r3_{mean} \\ r3_{std} \\ r3_{pl} \\ r3_{pu} \end{bmatrix}$$

secondary_response_mapping

$$Constraint\ Responses = \begin{bmatrix} g1 \\ g2 \\ g3 \\ g4 \end{bmatrix} = \begin{bmatrix} 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. \end{bmatrix} * \begin{bmatrix} r1_{mean} \\ r1_{std} \\ r2_{mean} \\ r2_{std} \\ r2_{pl} \\ r2_{pu} \\ r3_{mean} \\ r3_{std} \\ r3_{pl} \\ r3_{pu} \end{bmatrix} = \begin{bmatrix} r2_{pl} \\ r2_{pu} \\ r3_{pl} \\ r3_{pu} \end{bmatrix}$$

Interpreting the Dakota Input File

1. A different responses keyword is used to define the responses used during the OUU. Notice 1 objective response and 4 inequality constraints are defined.
2. The bounds specify the bounds for probability of survival and failure.

(1)

```
responses
   id_responses 'OPTIM_R'
   descriptors  'f_obj'  'r2_pl'  'r2_pu'  'r3_pl'  'r3_pu'
      numerical_gradients
      no_hessians
      objective_functions 1
         nonlinear_inequality_constraints 4
            lower_bounds  0.950000  -inf  0.950000  -inf
            upper_bounds  inf  0.050000  inf  0.050000
```

(2)

primary_response_mapping

$$Objective\ Response = [1.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.\ 0.] * \begin{bmatrix} r1_{mean} \\ r1_{std} \\ r2_{mean} \\ r2_{std} \\ r2_{pl} \\ r2_{pu} \\ r3_{mean} \\ r3_{std} \\ r3_{pl} \\ r3_{pu} \end{bmatrix}$$

secondary_response_mapping

$$Constraint\ Responses = \begin{bmatrix} g1 \\ g2 \\ g3 \\ g4 \end{bmatrix} = \begin{bmatrix} 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. \end{bmatrix} * \begin{bmatrix} r1_{mean} \\ r1_{std} \\ r2_{mean} \\ r2_{std} \\ r2_{pl} \\ r2_{pu} \\ r3_{mean} \\ r3_{std} \\ r3_{pl} \\ r3_{pu} \end{bmatrix} = \begin{bmatrix} r2_{pl} \\ r2_{pu} \\ r3_{pl} \\ r3_{pu} \end{bmatrix}$$

Cumulative and Complementary Probabilities

CDF and CCDF Values

Dakota outputs either *cumulative distribution function* (CDF) values or *complementary cumulative distribution function* (CCDF) values. Only one of these values may be output, not both together.

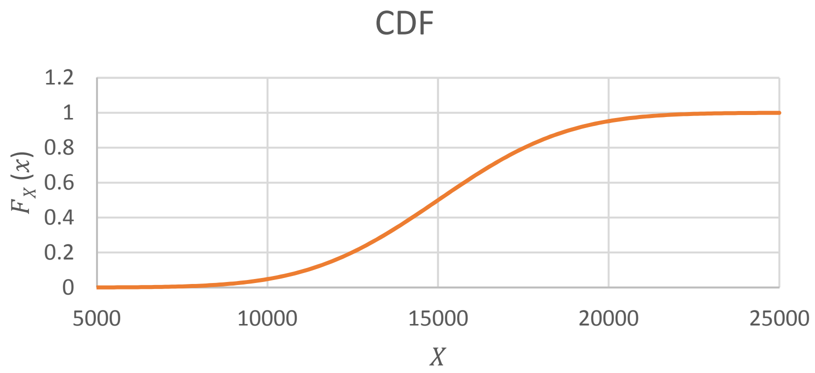
It must be decided if CDF or CCDF values are used throughout the UQ or OUU.

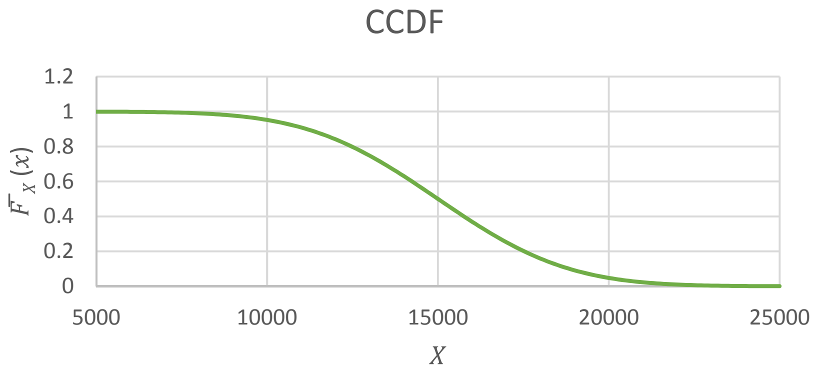
The CDF and CCDF are related by the following relationships

CDF = $F_X\,(X)$

CCDF = $\bar{F}_X(x)$ = 1 - $F_X\,(X)$

The following is information regarding the differences between CDF and CCDF values.






```
method
   id_method 'UQ'
      local_reliability
         model_pointer 'UQ_M'
         distribution
             cumulative
         response_levels  10000  20000
```

```
method
   id_method 'UQ'
      local_reliability
         model_pointer 'UQ_M'
         distribution
             complementary
         response_levels  10000  20000
```

CDF and CCDF Values

Consider a random variable X that corresponds to the axial stress of a truss member and is allowed to range between a lower bound of 10,000 and an upper bound of 20,000. X has a mean of 15000 and standard deviation of 3000.

- For the upper bound, if CDF values are used, the probability of survival is

  p_s = P(X ≤ 20000).

- For the upper bound, if CCDF values are used, the probability of failure is

  p_f = P(20000 < X).

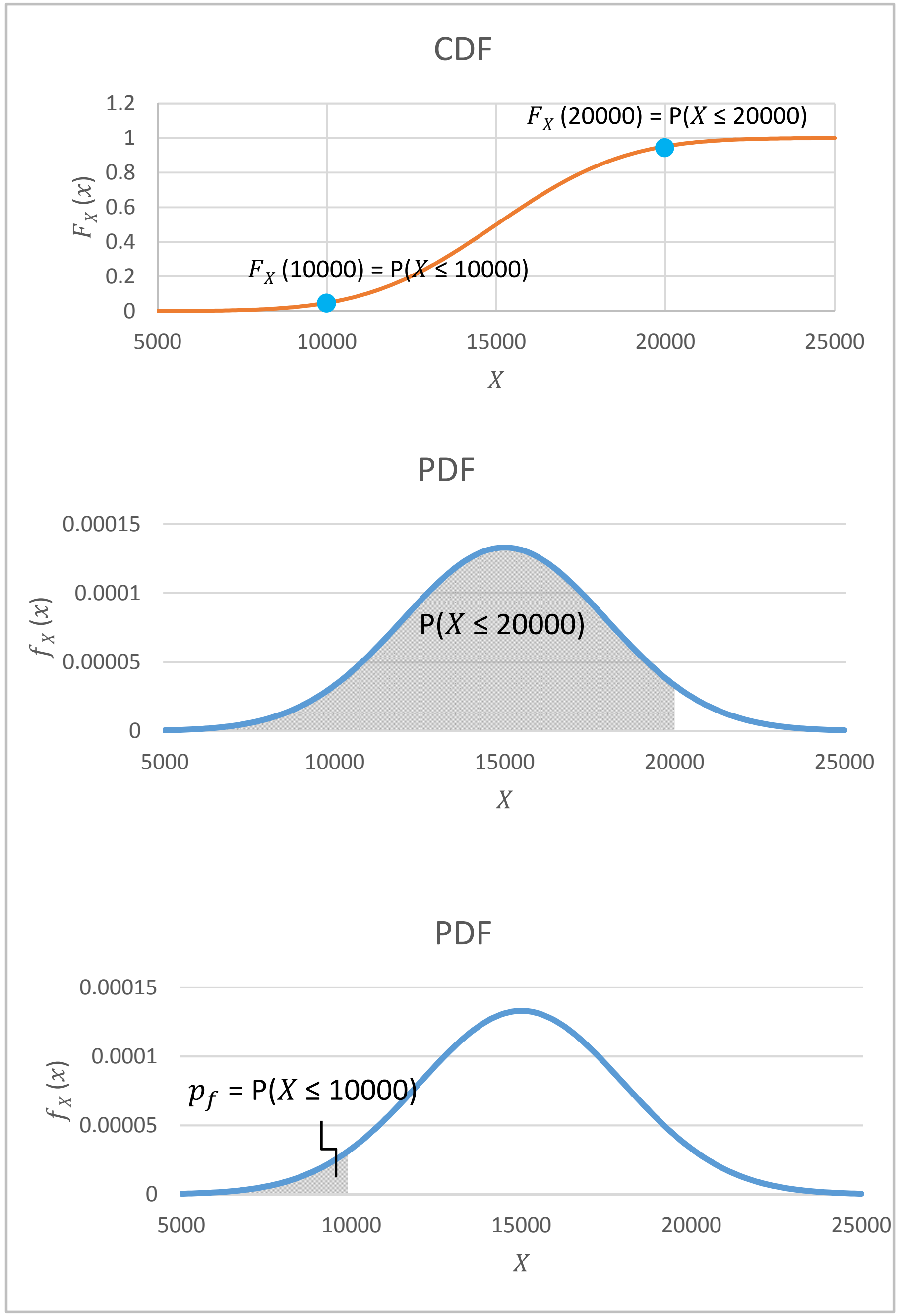
- For the lower bound, if CDF values are used, the probability of failure is

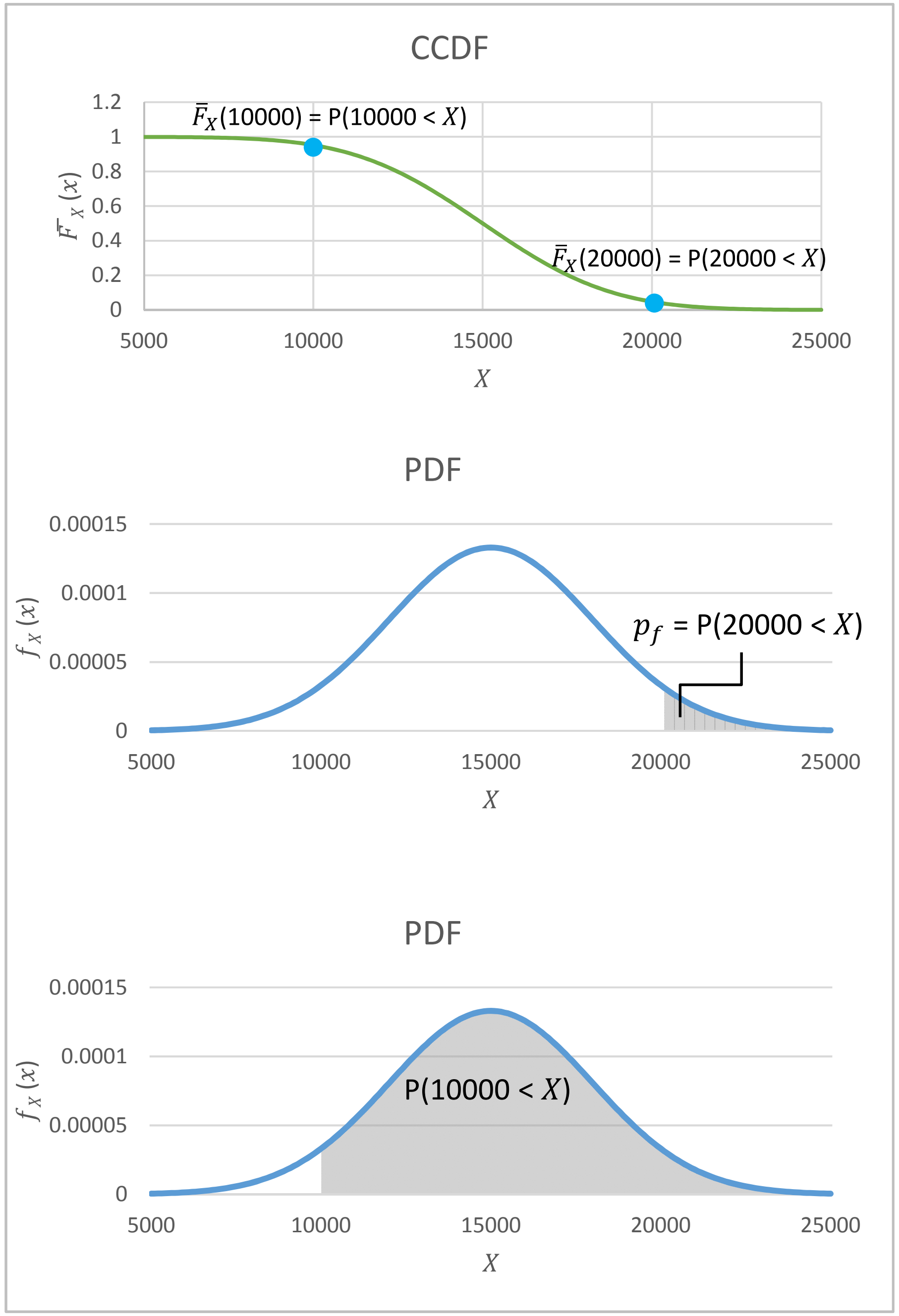
  p_f = P(X ≤ 10000).

- For the lower bound, if CCDF values are used, the probability of survival is

  p_s = P(10000 < X).

The use of CDF or CCDF values leads to a mixture of p_f and values p_s when configuring an OUU.





Dakota Output

- Consider the output from Dakota after an uncertainty quantification. Probabilities are output for response levels 10000 and 20000.
- If the cumulative option is used, the probabilities are P($X \le x$).
- If the complementary option is used, the probabilities are P($x < X$).
  - For response level 10000, the probability output is a probability of survival.
  - For response level 20000, the probability output is a probability of failure.

Dakota Input File

```
method
    id_method 'UQ'
        local_reliability
            model_pointer 'UQ_M'
            distribution
                cumulative
            response_levels  10000  20000
```

Dakota Output File

```
Level mappings for each response function:
Cumulative Distribution Function (CDF) for r1:
     Response Level  Probability Level  Reliability Index  General Rel Index
     --------------  -----------------  -----------------  -----------------
   1.0000000000e+04   5.0000000000e-02
   2.0000000000e+04   9.5000000000e-01
```

p_f = P($X \le 10000$)

p_s = P($X \le 20000$)

Dakota Input File

```
method
    id_method 'UQ'
        local_reliability
            model_pointer 'UQ_M'
            distribution
                complementary
            response_levels  10000  20000
```

Dakota Output File

```
Level mappings for each response function:
Complementary Cumulative Distribution Function (CCDF) for r1:
     Response Level  Probability Level  Reliability Index  General Rel Index
     --------------  -----------------  -----------------  -----------------
   1.0000000000e+04   9.5000000000e-01
   2.0000000000e+04   5.0000000000e-02
```

p_s = P($10000 < X$)

p_f = P($20000 < X$)

CDF and CCDF Values

The Dakota input files are configured to use distribution=complementary, which triggers the output of CCDF values.

Suppose at most the probability of failure of 0.05 (5%) is imposed. The bounds on the probabilities are as follows.

- For the upper bound, the quantity available is the probability of failure, so this quantity is directly constrained to at most 5%.

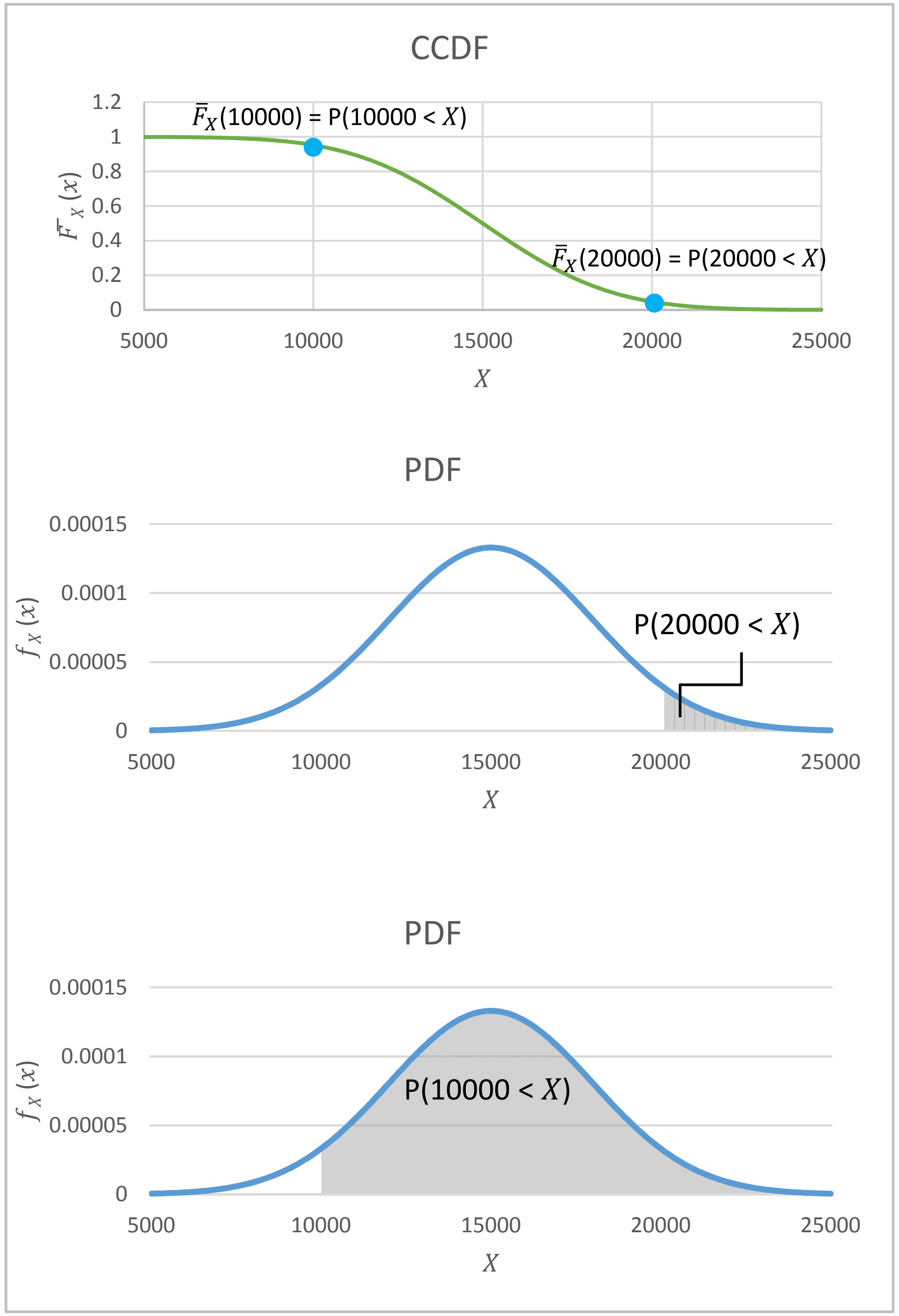
$$p_f = \mathrm{P}(20000 < X) < 0.05$$

- For the lower bound, the quantity available is the probability of survival. If at the most, a 5% probability of failure is imposed, this is equivalent to saying the probability of survival is greater than 95%. The constraint on the probability of survival is as follows:

$$0.95 < p_s = \mathrm{P}(10000 \leq X).$$

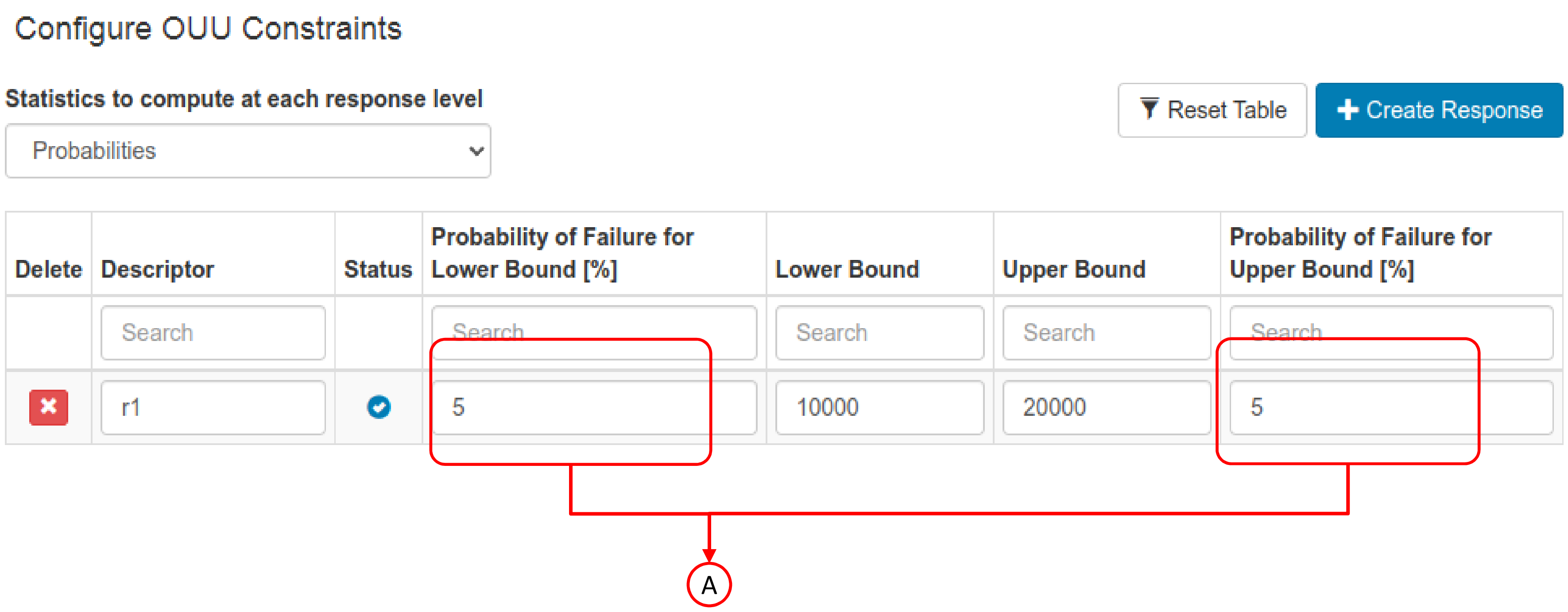
Dakota Input File

```
method
    id_method 'UQ'
        sampling
            model_pointer 'UQ_M'
              distribution
                         complementary
            sample_type
                    lhs
            samples 5000
            seed 12347
```



CDF and CCDF Values

A. In the web app, you supply limits on probabilities of failure for both the lower and upper bound. Internally, the web app is automatically managing the constraints for probabilities of failure and survival.



CDF and CCDF Values

Assume probabilities have been selected and constrained.

Configure UQ Responses

Statistics to compute at each response level

Probabilities
- Probabilities
- Reliabilities
- Generalized Reliabilities

Let

r1_pl = P(10000 < r2)

r1_pu = P(20000 < r2)

A. Refer to the table titled Configure OUU Objective and Additional Constraints

B. Close inspection of the final bounds shows that constraints on probability of survival P(10000 < r1) are provided for the lower bound of 10000, but constraints on probability of failure P(20000 < r1) are provided for the upper bound of 20000. This is because the complementary (CCDF) option was used.

Configure OUU Objective and Additional Constraints (A)

+ Create Constraint

	Objective (f_obj, g1)		**Constraint 1 (r1_pl)** ✖		**Constraint 2 (r1_pu)** ✖	
Label	**Include**	**Scale Factor**	**Include**	**Scale Factor**	**Include**	**Scale Factor**
r1_mean	☑	1.	☐		☐	
r1_standard_deviation	☐		☐		☐	
r1_p1	☐		☑	1.	☐	
r1_p2	☐		☐		☑	1.
Lower Bound		(B)	0.950000			
Upper Bound					0.050000	

CDF and CCDF Values

Some readers may be tempted to combine the probabilities and express a probability of survival as follows:

$$P(10000 < X \leq 20000).$$

If a maximum of 5% probability of failure is imposed and CDF values are available, the constraint is as follows:

$$0.95 < P(10000 < X \leq 20000).$$

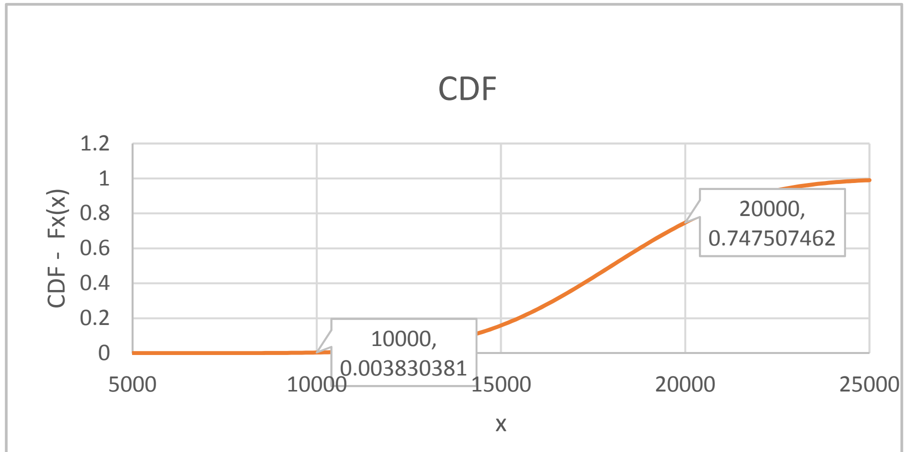
While this is valid, there is a drawback. A single probability value does not indicate if the distribution is violating the lower or upper bound.

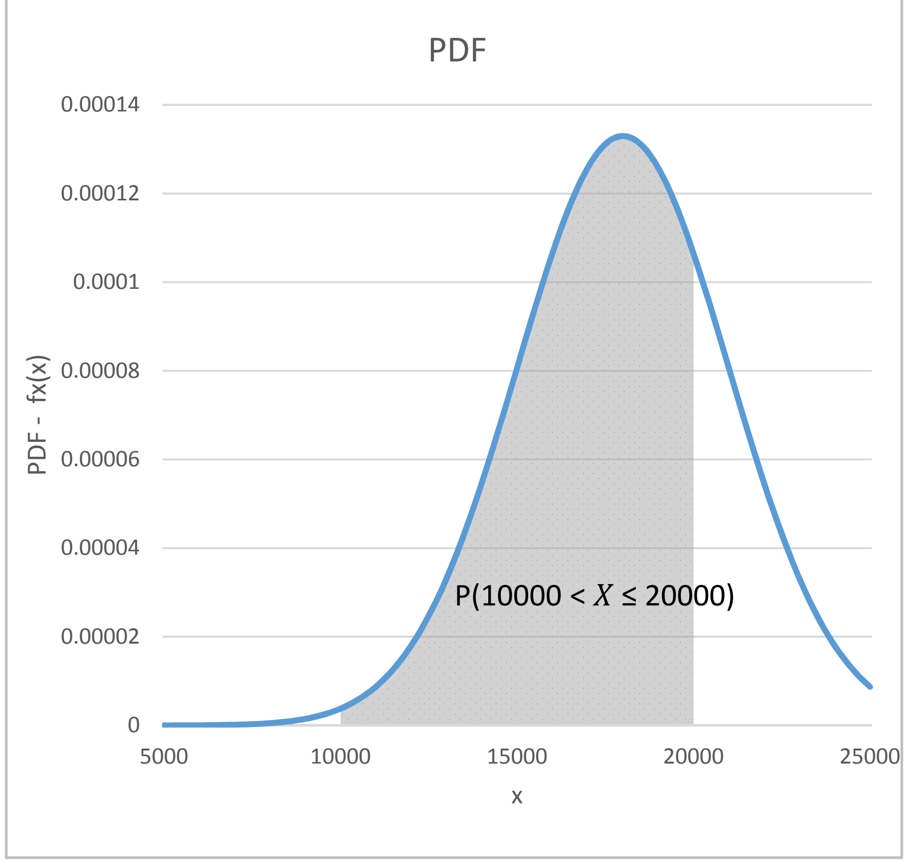
For example, suppose the following single probability is used: $P(10000 < X \leq 20000) = 0.74$ (74% survival). Since this single probability is less than the desired 95%, failure is expected. With a single probability, it is not known if the distribution is violating the lower or upper bound.

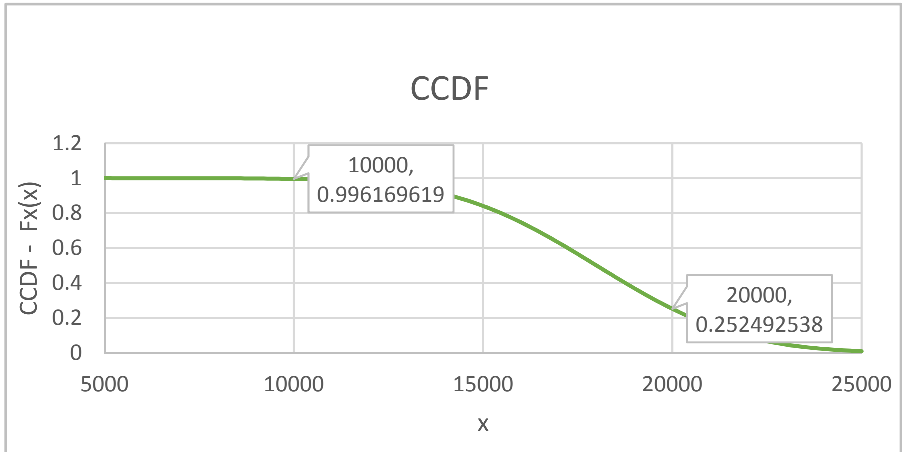
If separate probabilities are constrained, one for the lower and upper bounds, it makes it simpler to identify which of the bounds is being violated.

Consider the distribution shown on the right.

- For the upper bound (20000), the probability of failure is 25.25%. Since the maximum probability of failure is 5%, the probability of failure of the upper bound is violated.
- For the lower bound (10000), the probability of survival is 99.61%. The equivalent probability of failure is 0.38% and is within the 5% imposed.







CDF and CCDF Values

Final Comments

During the optimization under uncertainty (OUU), the mean and standard deviation of the response's distribution will vary. The variation depends on the shape of the response function.

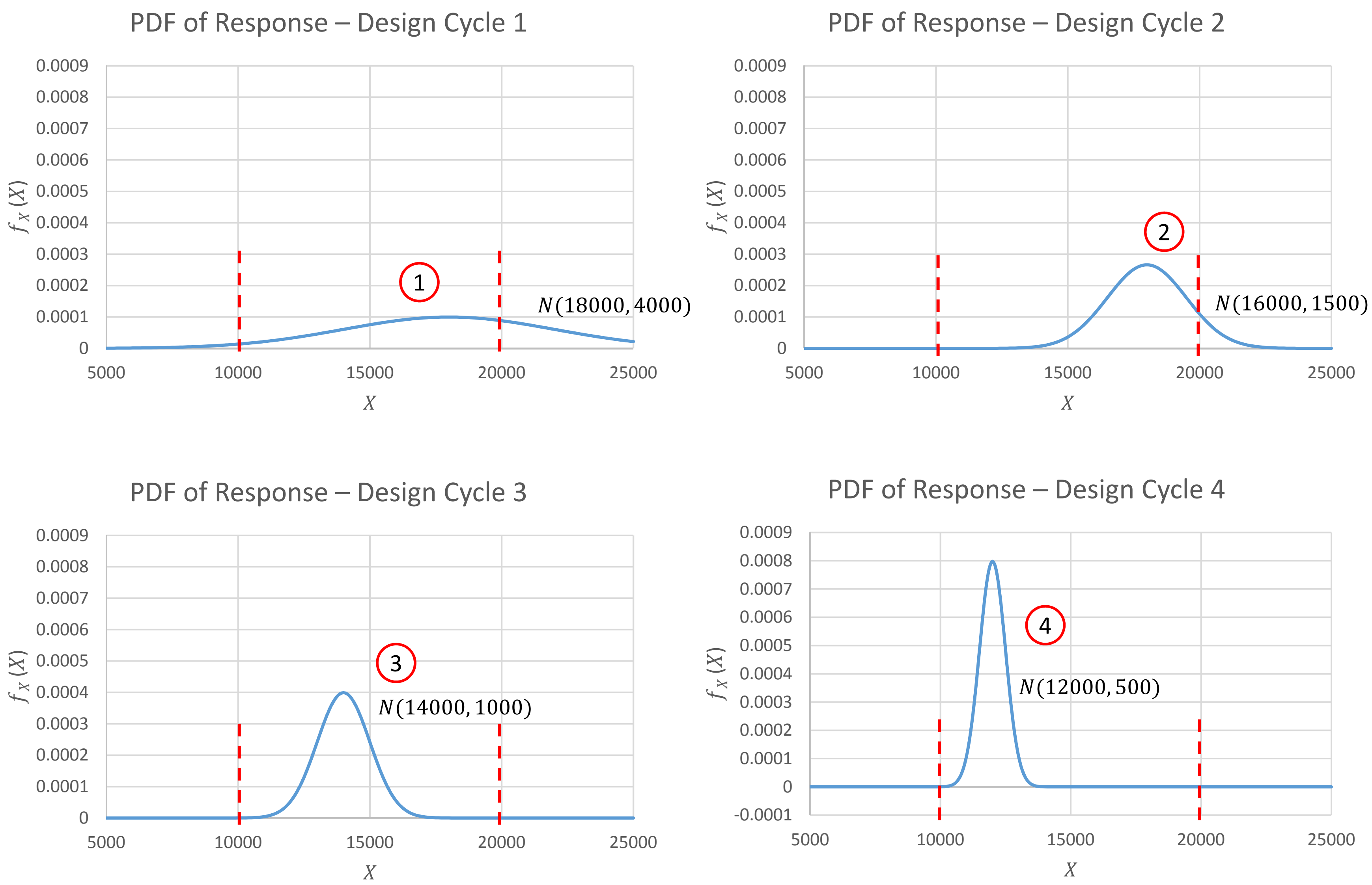
To the right is an example of the distribution of a response during an OUU.

1. The standard deviation is too large and the probabilities of failure for both the lower and upper bound are greater than 5%. The design is infeasible.

2. The mean has moved far enough to the right such that the probability of failure for the upper bound is greater than 5%. The design is infeasible.

3. The mean is approximately half way between the lower and upper bound and yields a probability of failure within 5% for both lower and upper bounds. The design is feasible.

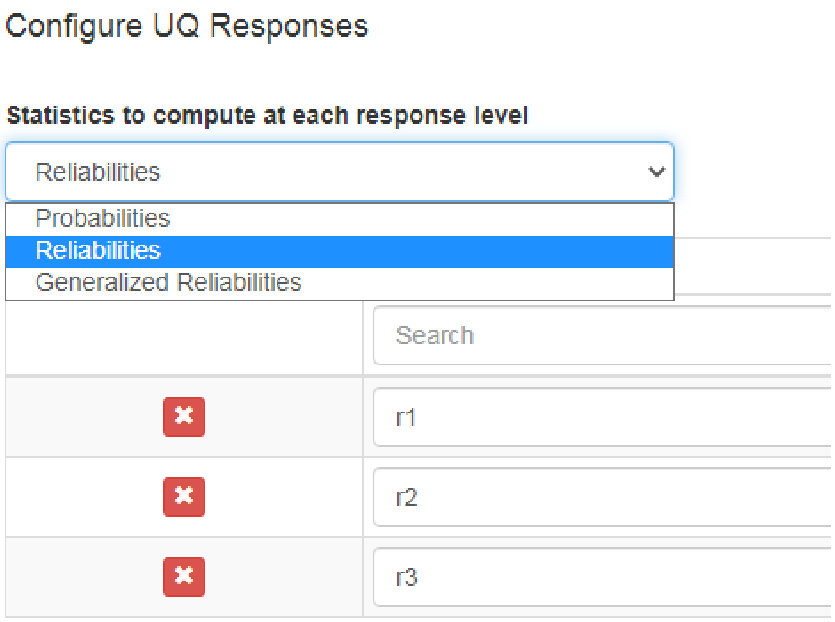
4. While the mean is close to the lower bound, the standard deviation is small enough such that probability of failure for the lower bound is less than 5%. The design is feasible.



Probabilities, Reliability Index and Generalized Reliability Index

Probabilities, Reliability Index and Generalized Reliability Index

When configuring an OUU and constraining probabilities of failure, you have the option of constraining probabilities, reliability indices or generalized reliability indices. The following is a brief description of each.



What is probability?

The likelihood of a random variable X exceeding a response level is denoted as a probability, e.g. P($X \le$ a).

Consider a random variable X with a mean of 15000, standard deviation of 3000, and bounded between response levels 10000 and 20000.

If cumulative distribution function (CDF) values are available, the following probabilities may be determined.

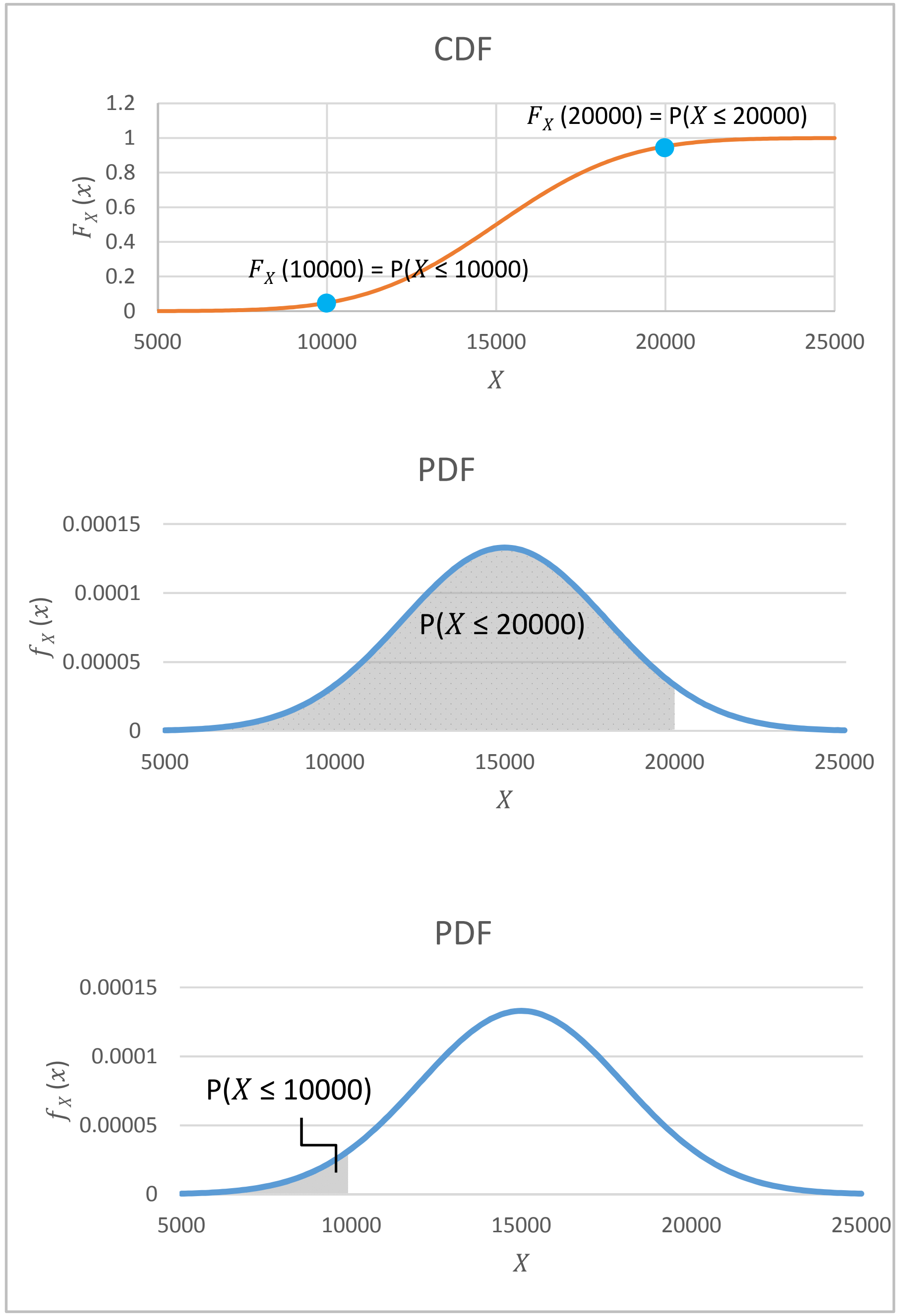
- P($X \le 20000$)
- P($X \le 10000$)

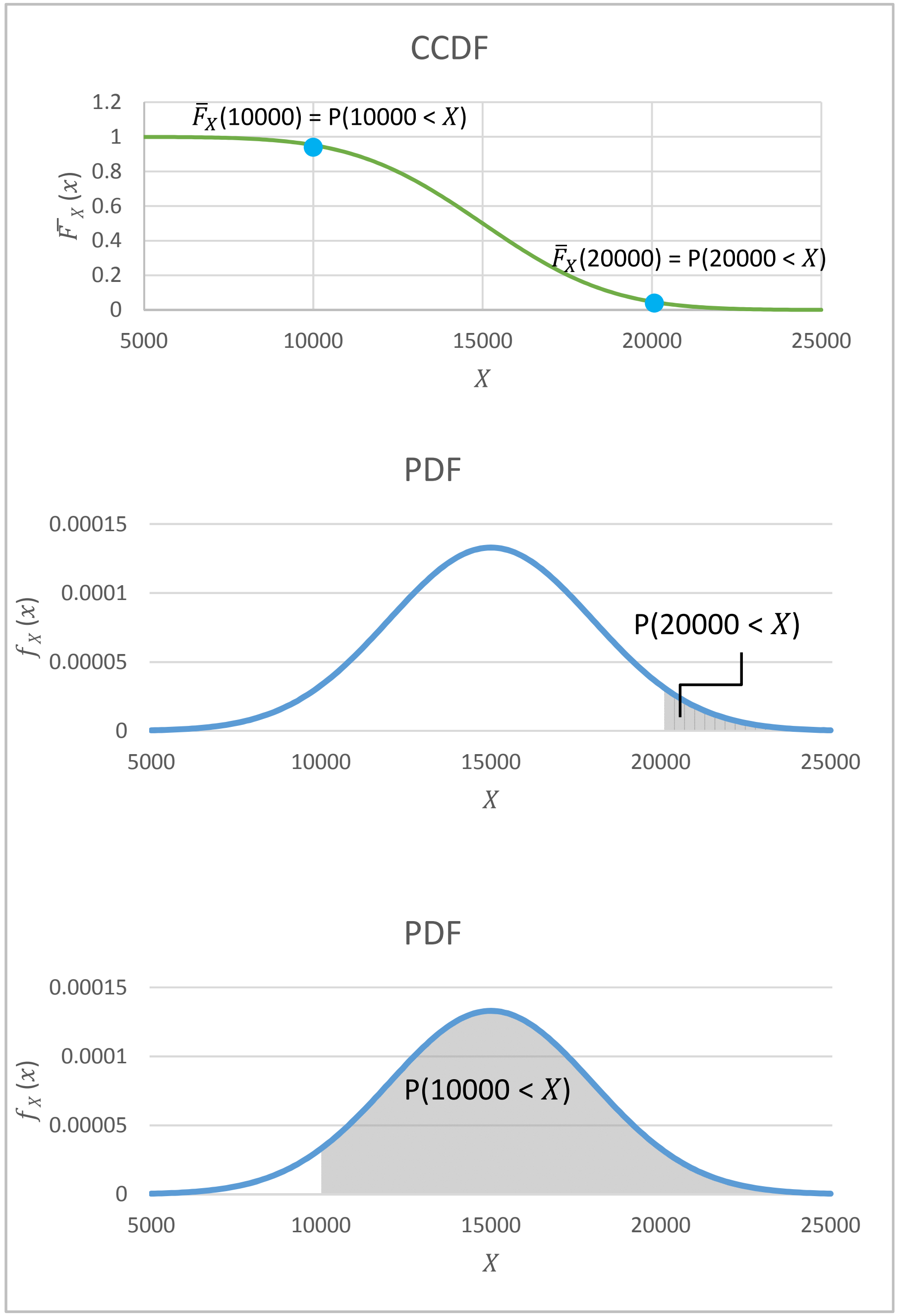
If complementary cumulative distribution function (CCDF) values are available, the following probabilities may be determined.

- P($20000 < X$)
- P($10000 < X$)

The CDF ($F_X(x)$) and CCDF ($\bar{F}_X(x)$) are related by the following expression.

$$F_X(x) = 1.0 - \bar{F}_X(x)$$





What is probability?

Also, the following probability may be determined.

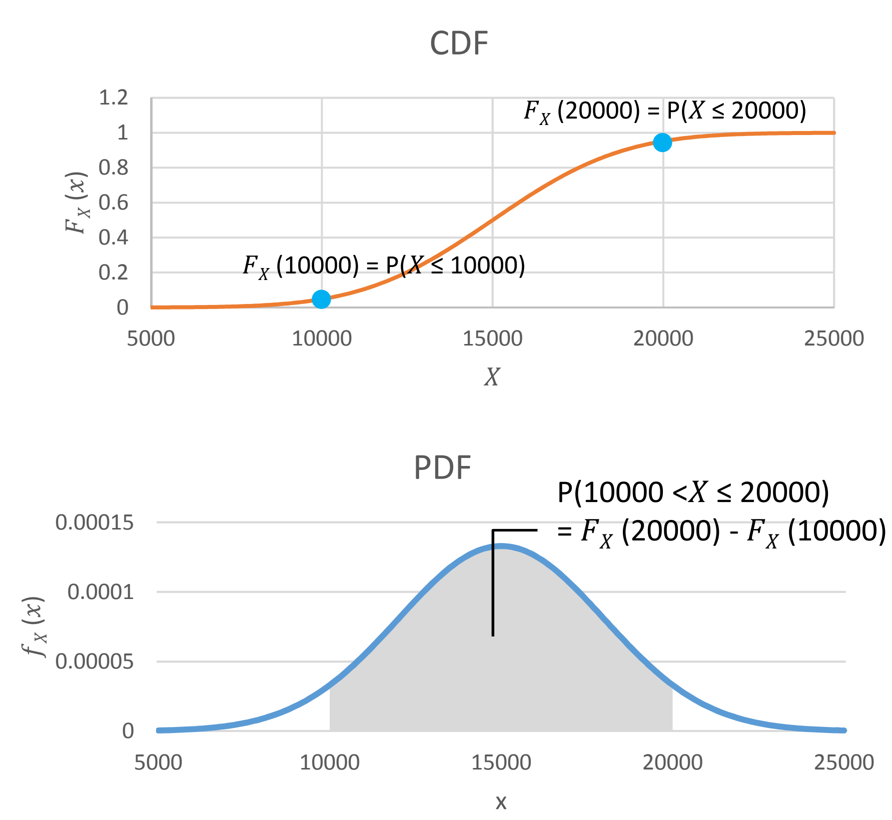
- P(10000 < X ≤ 20000)

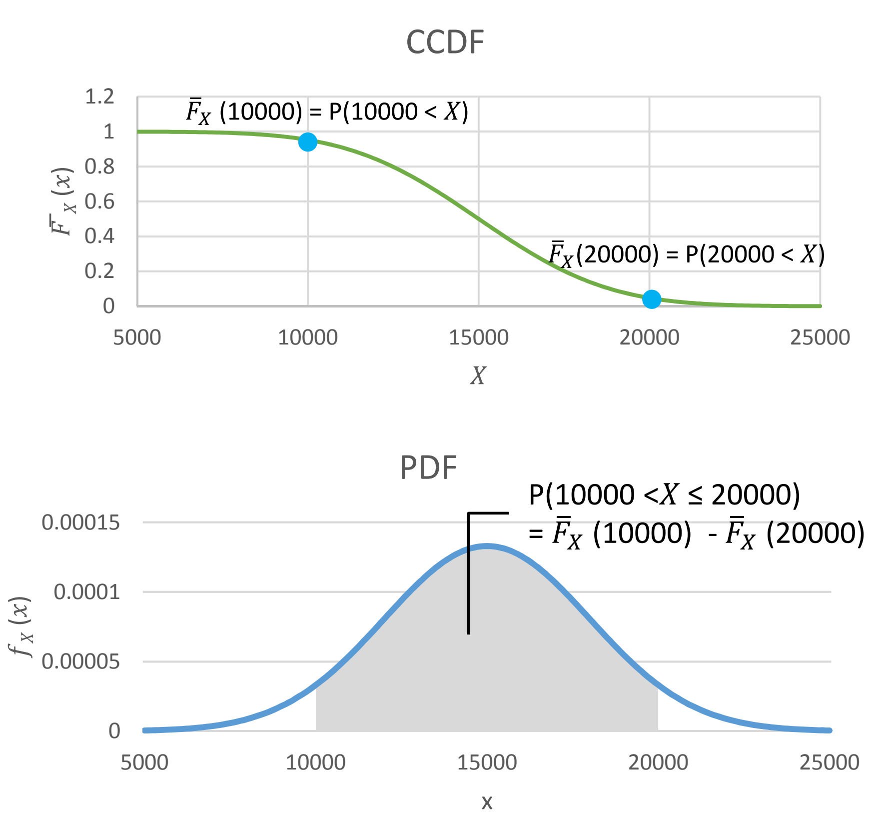
If cumulative distribution function (CDF) values are available, this probability may be determined as follows.

P(10000 < X ≤ 20000)
= P(X ≤ 20000) - P(X ≤ 10000)
= F_X (20000) - F_X (10000)

If complementary cumulative distribution function (CCDF) values are available, this probability may be determined as follows.

P(10000 < X ≤ 20000)
= P(10000 < X) - P(20000 < X)
= $\bar{F}_X$ (10000) - $\bar{F}_X$ (20000)





What is $\Phi(x)$?

$\Phi(x)$ is the cumulative distribution function of a standardized normal distribution.

A standardized normal distribution is a normal distribution with mean 0 and standard deviation of 1.

$$X \sim N(0,1)$$

Consider a random variable X that has a normal distribution

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

The probability density function (PDF) for a normal distribution is as follows

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

The cumulative distribution function (CDF) for a normal distribution is as follows

$$F_X(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)\right]$$

Where erf is defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\,dt\,.$$

The CDF of a standardized normal distribution (μ=0, σ=1) is as follows

$$\Phi(x) = F_X(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

What is a reliability index?

Per the Dakota Reference Manual, “CDF/CCDF reliabilities are calculated for specified response levels by computing the number of sample standard deviations separating the sample mean from the response level.” The response level may either be the lower or upper bound. The reliability, often known as the reliability index, is defined as:

$$\beta = \frac{\mu_{ri} - Response\ Level}{\sigma_{ri}}$$

When the CDF option is used, the probability and reliability index β are related via the following expression:

$$p(X \le x) = \Phi(-\beta)$$

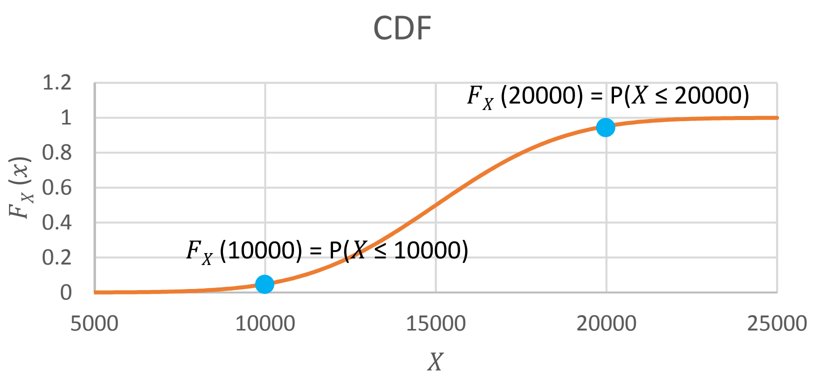
When the CCDF option is used, the probability and reliability index $\bar{\beta}$ are related via the following expression:

$$p(x < X) = \Phi(-\bar{\beta})$$

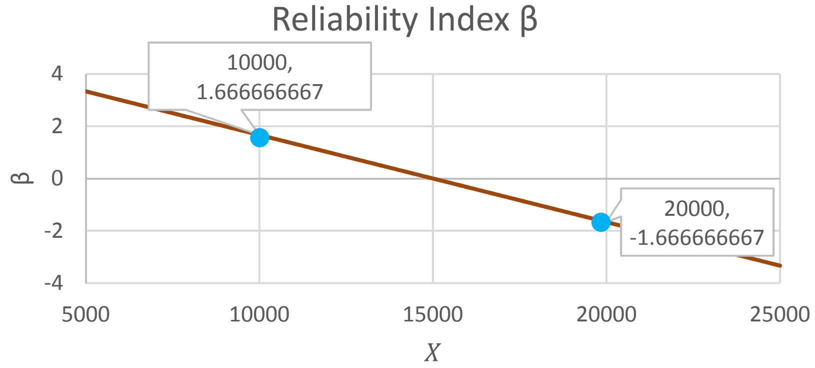
Constraining reliability indices is equivalent to constraining probabilities.

The reliability index applies to normal or lognormal distributions.

When using local reliability methods for UQ, OUU converges faster when constraining reliability indices, not probabilities.

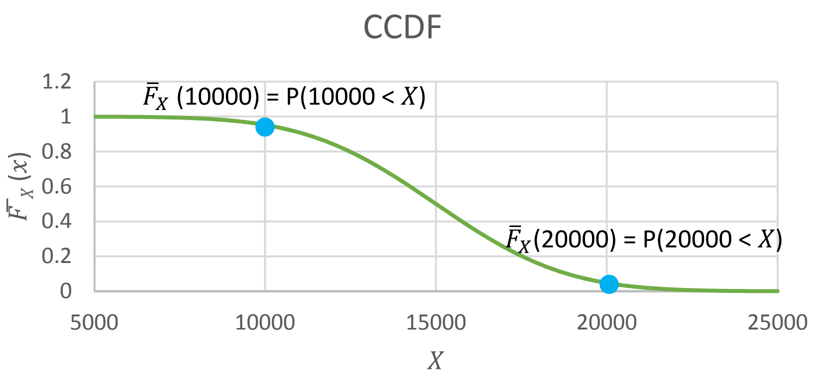


$$p(X \le x) = F_X(x) = \Phi(-\beta)$$

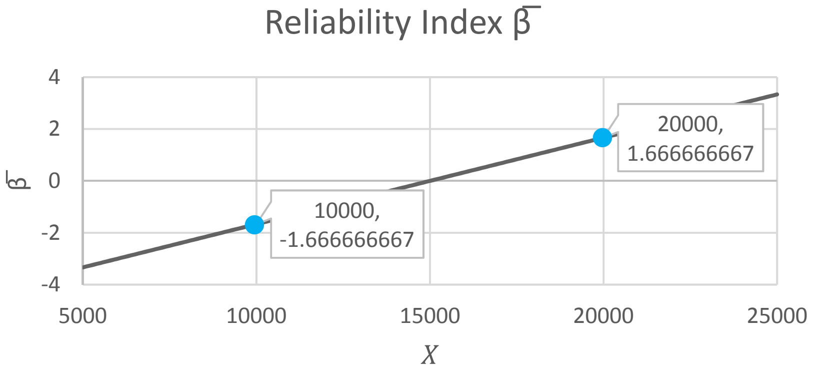


$$\beta = \frac{15000 - X}{3000}$$

Recall the following: The random variable X has a mean of 15000, standard deviation of 3000, and bounded between response levels 10000 and 20000.



$$p(x < X) = \bar{F}_X(x) = \Phi(-\bar{\beta})$$



$$\bar{\beta} = \frac{X - 15000}{3000}$$

$$\bar{\beta}_{20000} = \frac{20000 - 15000}{3000} = 1.67$$

$$\bar{\beta}_{10000} = \frac{10000 - 15000}{3000} = -1.67$$

What is a reliability index?

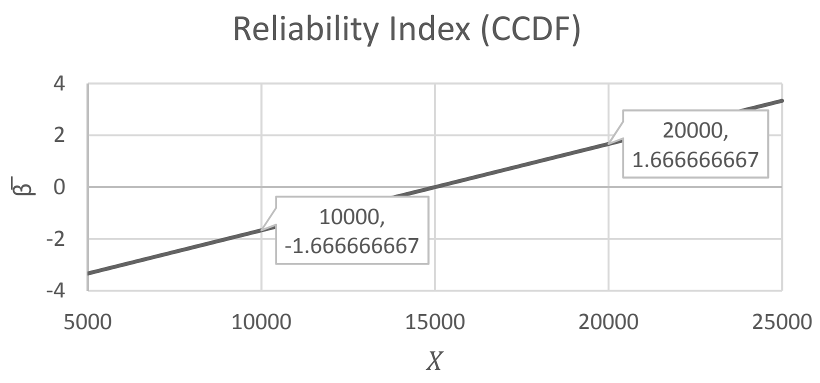
The goal is to constrain the following probabilities to at most 5% failure.

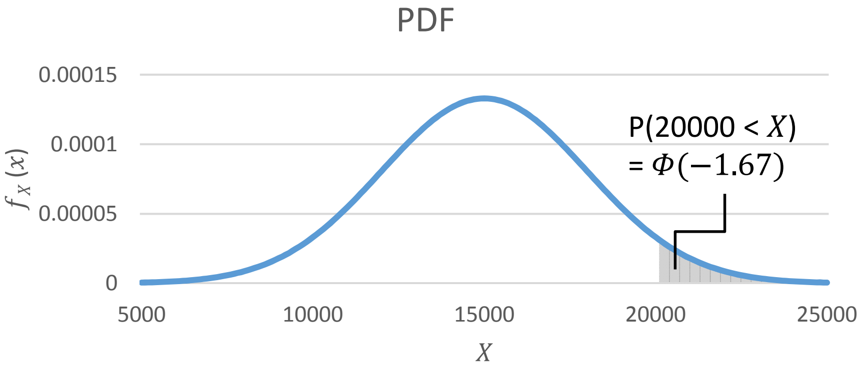
$p_{f,\ lower} = P(X \leq 10000) < 0.05$

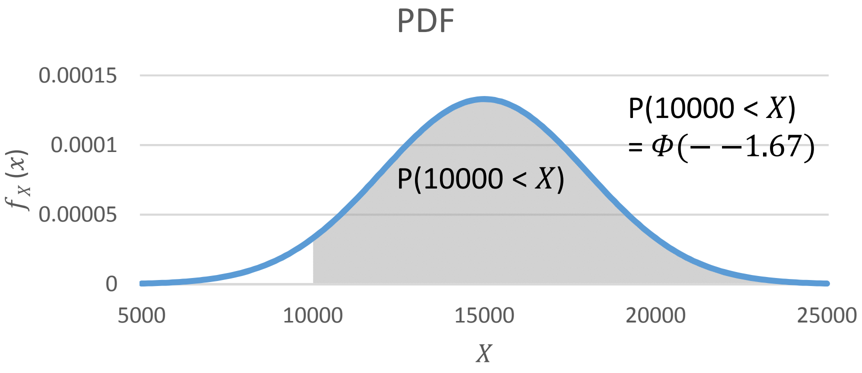
$p_{f,\ upper} = P(20000 < X) < 0.05$

Consider the CCDF reliability indices $\bar{\beta}$. The same constraints on probability of failure are expressed as constraints on reliability indices.

Bound	Probability of Failure	Constraint on Probability of Failure	Equivalent Constraint but with Reliability Indices
Upper bound =20000	$p_{f,\ lower} = P(20000 < X)$	$p_{f,\ lower} < 0.05$	$1.67 < \bar{\beta}_{lower}$
Lower bound =10000	$p_{f,\ upper} = P(X \leq 10000)$ $= 1 - P(10000 < X)$	$p_{f,\ upper} < 0.05$	$\bar{\beta}_{upper} < -1.67$







What is a generalized reliability index?

So far, *reliability indices* have been discussed. There is another type of reliability index named *generalized reliability index* that is worth briefly mentioning.

What is a limit state function?

The limit state function is the response function, e.g. stress, displacement, etc.

What are generalized reliabilities?

It has been assumed the limit state function is linear, so its *reliability index* is simply defined as:

$$\beta = -\Phi^{-1}(p).$$

When the limit state function is nonlinear, a *generalized reliability index*[1] is more suitable and is defined as:

$$\beta_{gen} = -\Phi^{-1}\left(\int_{S_a} \Phi(u_1)\Phi(u_2)\ldots\Phi(u_n)\right)$$

No modifications are necessary to the exercise, but note the following.

A. Generalized reliability indices are output by Dakota by using the keyword gen_reliabilities.

B. If performing a UQ only, the Dakota output tables will have generalized reliability index values in the column name "General Rel Index"

References

1. Ditlevsen, O. "Generalized Second Moment Reliability index." *Journal of Structural Mechanics*, Vol. 7, No. 4, pp. 435-451, 1979.

```
method
  id_method 'UQ'
    local_reliability
      model_pointer 'UQ_M'
      distribution
          complementary
      response_levels -20000 20000 -20000 20000
        compute
            gen_reliabilities      (A)
        num_response_levels 0 2 2
```

```
Level mappings for each response function:
Complementary Cumulative Distribution Function (CCDF) for r2:

     Response Level  Probability Level  Reliability Index  General Rel Index
     --------------  -----------------  -----------------  -----------------
```

(B)

Configuring bounds for probabilities of failure in Sandia Dakota

Configuring bounds for probabilities of failure in Sandia Dakota

1. The Dakota input file study_d.in shows the bounds for probability of survival and failure are defined.
2. Notice the keyword distribution is set to complementary.

- The values displayed on this page are from a separate OUU and should not be confused with the values from the OUU configured in this workshop.

study_d.in

```
responses
   id_responses 'OPTIM_R'
   descriptors  'f_obj'  'r2_pl'  'r2_pu'  'r3_pl'  'r3_pu'
      numerical_gradients
      no_hessians
      objective_functions 1
         nonlinear_inequality_constraints 4
            lower_bounds  0.950000  -inf  0.950000  -inf
            upper_bounds  inf  0.050000  inf  0.050000        (1)

method
   id_method 'UQ'
      sampling
         model_pointer 'UQ_M'
         distribution
               complementary   (2)
         response_levels  -20000  20000  -20000  20000
            num_response_levels  0  2  2
         sample_type
               lhs
         samples 5000
         seed 12347
```

Configuring bounds for probabilities of failure in Sandia Dakota

The Dakota output is reporting probabilities under the constraints section.

1. The values of 1.0 represent the probability of survival (p_s) for the lower bounds of -20000. Since the goal was to ensure the p_s was greater then 0.95 and the final value was 1.0, the constraint is satisfied.
2. For the other values of 0.05055, these represent probability of failure (p_f) for the upper bounds of 20000. Since the goal was to ensure this value was at most 0.05 and since the final value was 0.05055, the constraint is slightly violated.
3. When probabilities were constrained internally during the OUU, a total of 25 MSC Nastran runs were required for convergence.

- The values displayed on this page are from a separate OUU and should not be confused with the values from the OUU configured in this workshop.

```
<<<<< Function evaluation summary (UQ_I): 30 total (25 new, 5 duplicate)
<<<<< Best parameters          =
                      9.0964936275e-01 x1_mean
                      3.1138241054e-01 x2_mean
<<<<< Best objective function  =
                      2.8842592000e+00
<<<<< Best constraint values   =
                      1.0000000000e+00
                      5.0551279430e-02
                      1.0000000000e+00
                      5.0551279430e-02
<<<<< Best evaluation ID not available
(This warning may occur when the best iterate is comprised of multiple interface
evaluations or arises from a composite, surrogate, or transformation model.)


<<<<< Iterator conmin_mfd completed.
<<<<< Environment execution completed.
DAKOTA execution time in seconds:
  Total CPU        =    101.755 [parent =    101.755, child = -1.42109e-14]
  Total wall clock =    106.657
```

① p_s = P(-20000 < X)

② p_f = P(20000 < X)

③ (25 new,

Final Comment

For this example, it was stated that a maximum 5% probability of failure was desired.

1. One option is to constrain the probabilities directly.
2. An alternative is to constrain equivalent reliability indices.

When the local reliability is used for UQ, it is shown that constraining equivalent reliabilities yields faster optimizations than directly constraining probabilities. Also, both approaches yield nearly the same optimal solution, so constraining reliabilities or probabilities are both appropriate. Constraining reliabilities is preferred since it produces faster optimizations.

- The values displayed on this page are from a separate OUU and should not be confused with the values from the OUU configured in this workshop.

Quantity of Interest Constrained	Number of MSC Nastran Runs to Converge
Reliabilities	17
Probabilities	25

OUU – Constraining reliabilities (2)

```
<<<<< Function evaluation summary (UQ_I): 22
total (17 new, 5 duplicate)
<<<<< Best parameters          =
                      9.0702483418e-01 x1_mean
                      3.1924786716e-01 x2_mean
<<<<< Best objective function  =
                      2.8847015000e+00
<<<<< Best constraint values   =
                     -4.9722756150e+01
                      1.6444557973e+00
                     -4.9722756150e+01
                      1.6444557973e+00
<<<<< Best evaluation ID not available
(This warning may occur when the best iterate is
comprised of multiple interface
evaluations or arises from a composite,
surrogate, or transformation model.)


<<<<< Iterator conmin_mfd completed.
<<<<< Environment execution completed.
DAKOTA execution time in seconds:
  Total CPU        =      80.795 [parent =
80.795, child = 1.42109e-14]
  Total wall clock =      80.867
```

OUU – Constraining probabilities (1)

```
<<<<< Function evaluation summary (UQ_I): 30
total (25 new, 5 duplicate)
<<<<< Best parameters          =
                      9.0964936275e-01 x1_mean
                      3.1138241054e-01 x2_mean
<<<<< Best objective function  =
                      2.8842592000e+00
<<<<< Best constraint values   =
                      1.0000000000e+00
                      5.0551279430e-02
                      1.0000000000e+00
                      5.0551279430e-02
<<<<< Best evaluation ID not available
(This warning may occur when the best iterate is
comprised of multiple interface
evaluations or arises from a composite,
surrogate, or transformation model.)


<<<<< Iterator conmin_mfd completed.
<<<<< Environment execution completed.
DAKOTA execution time in seconds:
  Total CPU        =     101.755 [parent =
101.755, child = -1.42109e-14]
  Total wall clock =     106.657
```

Configuring bounds for both UQ and OUU variables in Sandia Dakota

Configuring bounds for both UQ and OUU variables in Sandia Dakota

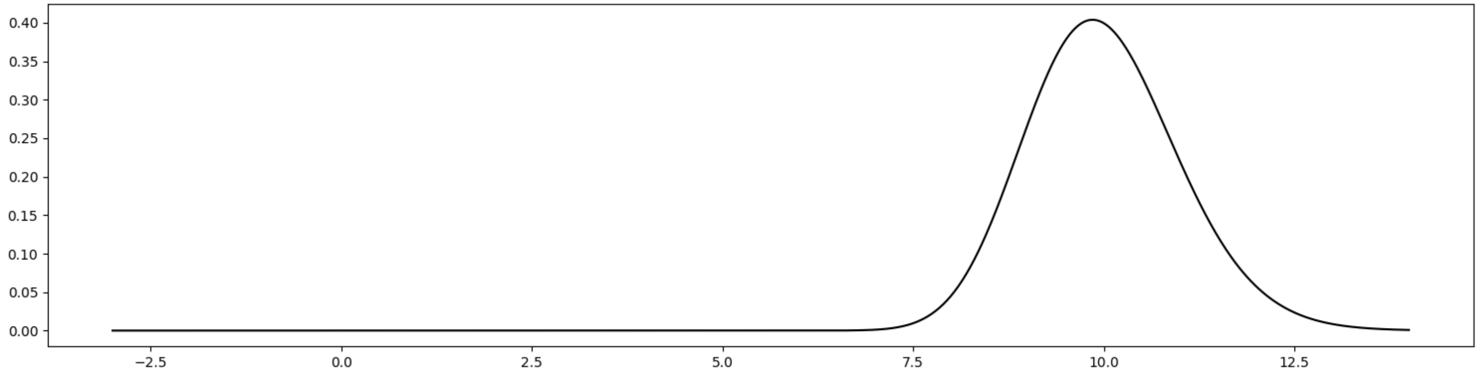
The following applies if uncertain variables have a normal or lognormal distribution.

When performing optimization under uncertainty with Sandia Dakota and configuring bounds for both the uncertain variables and the optimization variables, the displayed errors are sometimes encountered.

This brief presentation discusses the cause and solution for this error.

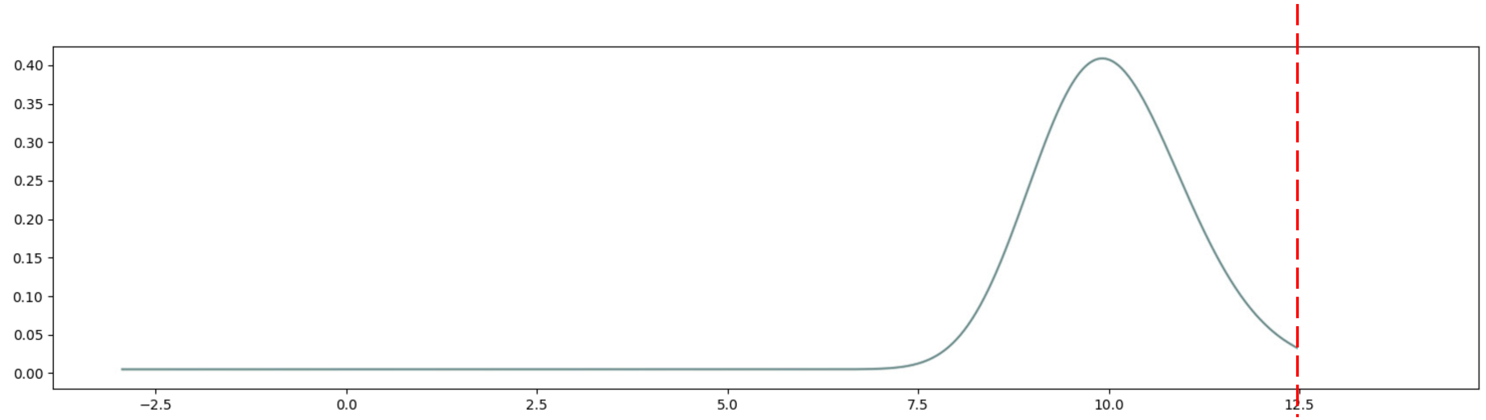
```
File LHS.ERR

1           Lower bound of a bounded normal or lognormal
           distribution must be less than the 0.999 quantile.
           Found in Distribution #    2
 Error was detected during LHS run


1           Upper bound of a bounded normal or lognormal
           distribution must be greater than the 0.001 quantile.
           Found in Distribution #    2
 Error was detected during LHS run
```

Consider an uncertain variable's lognormal distribution with a mean of 10.0 and standard deviation of 0.01.

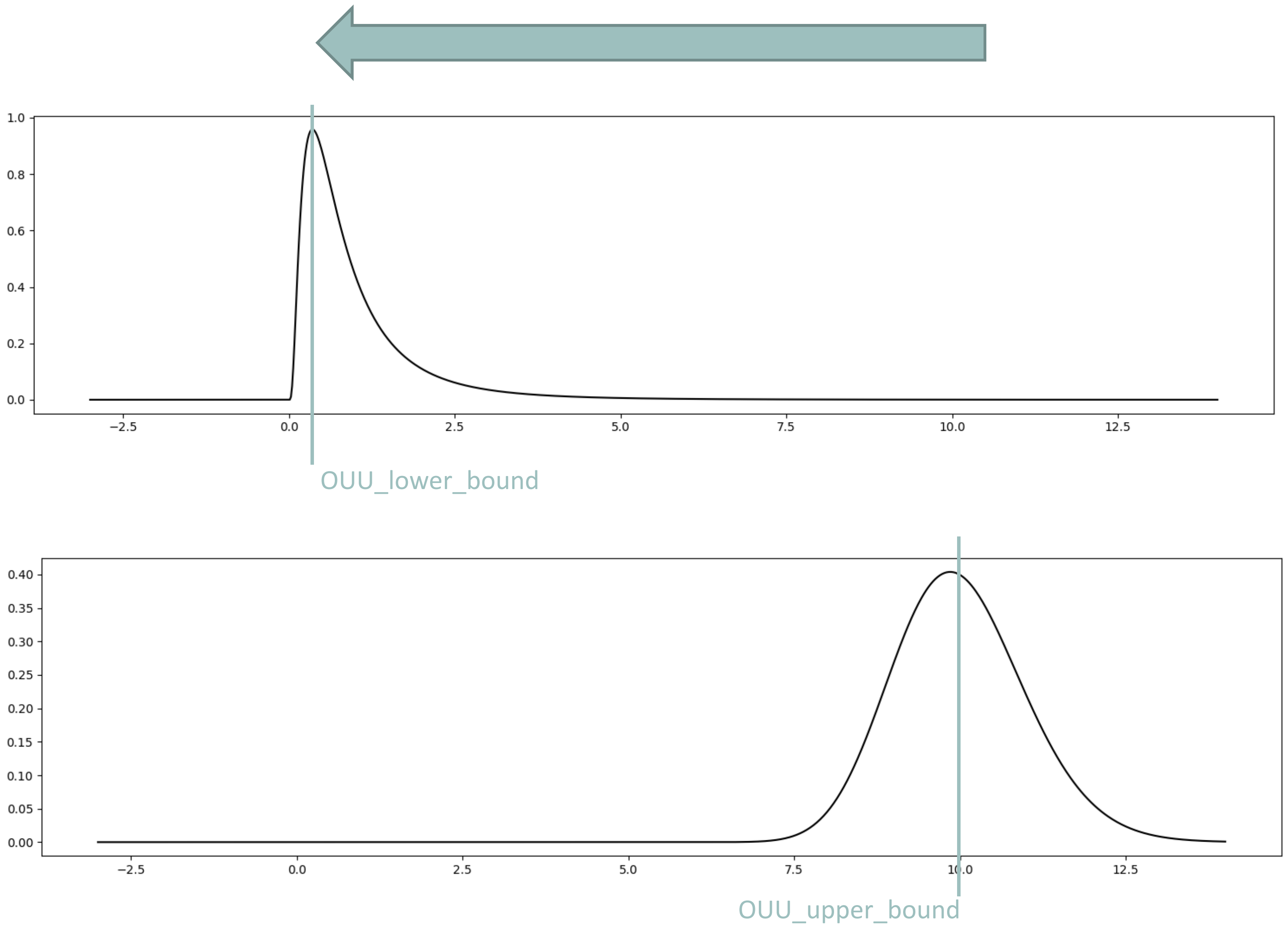


Suppose an upper bound on the distribution was equal to 12.5. No draws or samples will exceed the value of 12.5.

The bounds imposed on uncertain variables are termed the *UQ bounds*.



Direction of variable’s mean during the optimization.



During OUU, the mean of the variables may be varied and optimized.

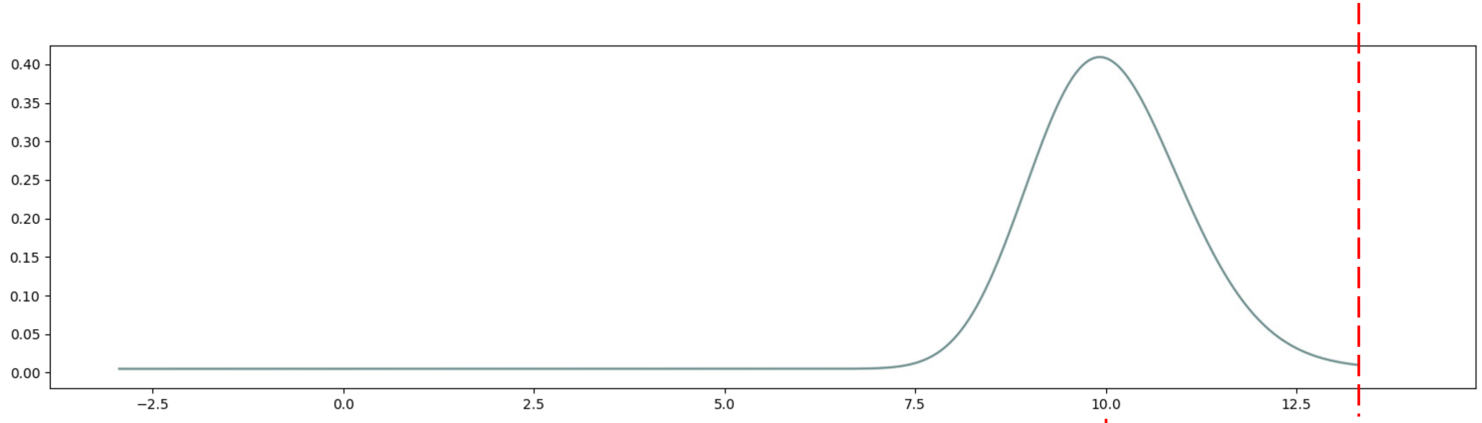
Consequently, the distribution for each variable will change as the mean varies during the optimization.

In this example, the variable’s mean is allowed to vary between 1.0 and 10.0. Notice the change in its distribution. These bounds are termed the *OUU bounds*.

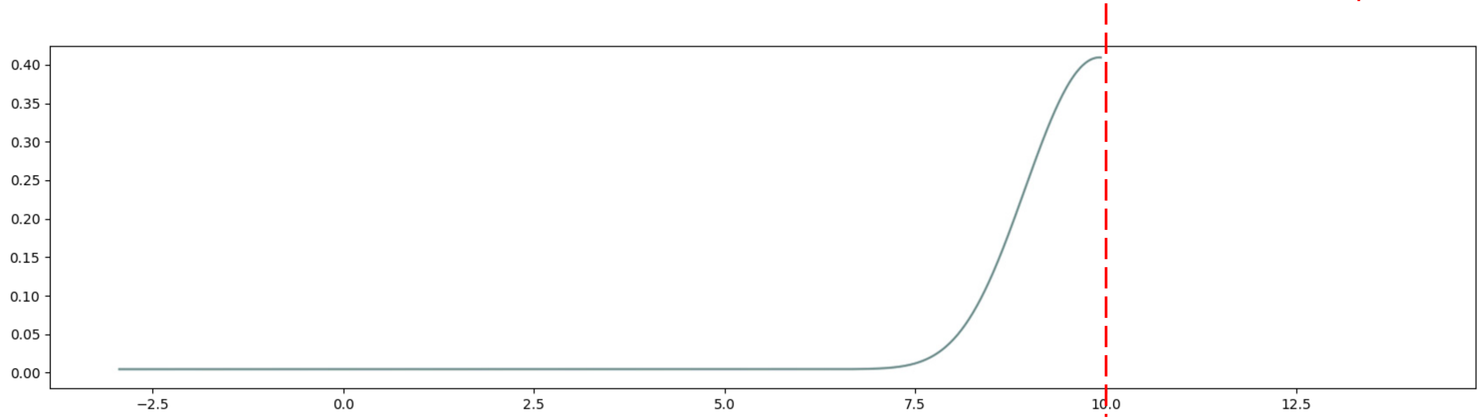
Suppose the OUU variable's initial value is at the upper bound of the OUU variable, which is 10.0.

Three different UQ upper bounds are displayed.

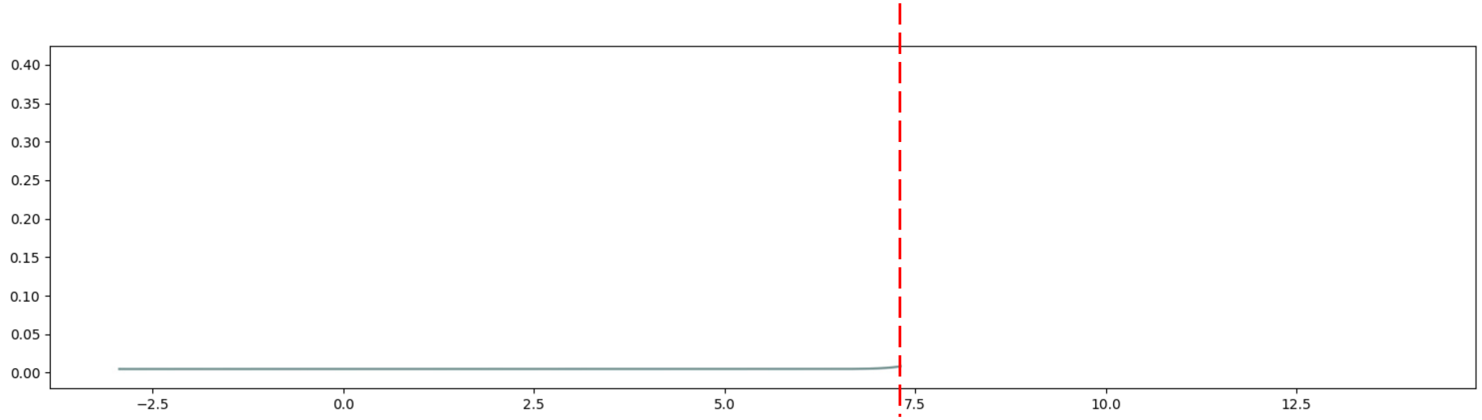
If the UQ or OUU upper bounds are not properly configured, there will be a nearly 0% probability of drawing a sample from the distribution. This 0% probability causes the error.



P(xi < UQ_upper_bound) = 99.9%



P(xi < UQ_upper_bound) = 50%



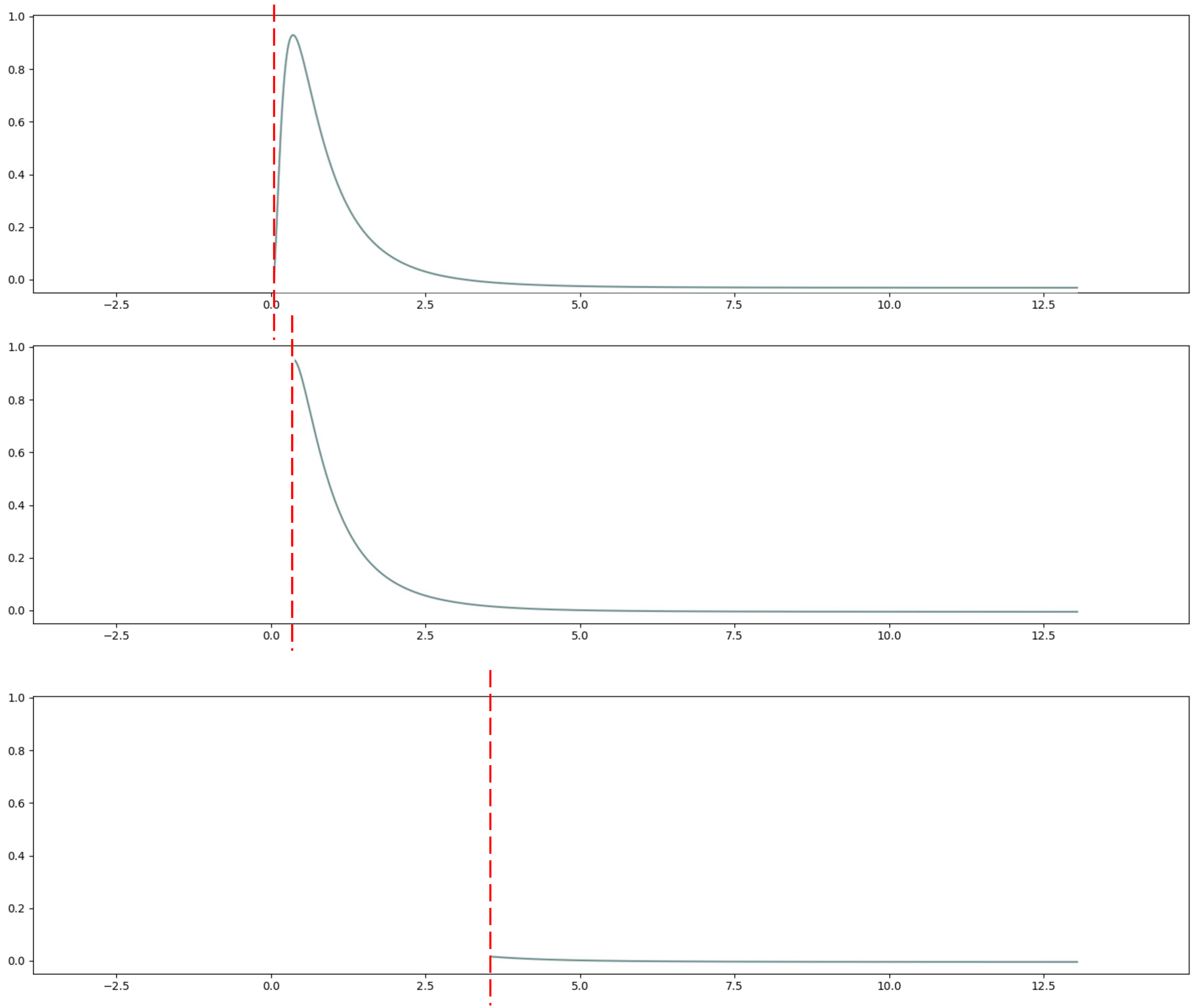
P(xi < UQ_upper_bound) = .1%

The probability of drawing a sample less than the UQ upper bound is less than .1%, which is very rare. The OUU procedure terminates due to inability to draw samples.

Similarly for the lower bound, suppose the OUU variable's initial value is at the lower bound of the OUU variable, which is 1.0.

Three different UQ lower bounds are displayed.

If the UQ or OUU lower bounds are not properly configured, there will be a nearly 0% probability of drawing a sample from the distribution. This 0% probability causes the error.



P(xi > UQ_lower_bound) = 99.9%

P(xi > UQ_lower_bound) = 50%

P(xi > UQ_lower_bound) = .1%

The probability of drawing a sample greater than the UQ lower bound is less than .1%, which is very rare. The OUU procedure terminates due to inability to draw samples.

Sandia Dakota flags problematic UQ and OUU bounds with this message.

```
LHS.ERR

1           Lower bound of a bounded normal or lognormal
           distribution must be less than the 0.999 quantile.
           Found in Distribution #    2
 Error was detected during LHS run


1           Upper bound of a bounded normal or lognormal
           distribution must be greater than the 0.001 quantile.
           Found in Distribution #    2
 Error was detected during LHS run
```

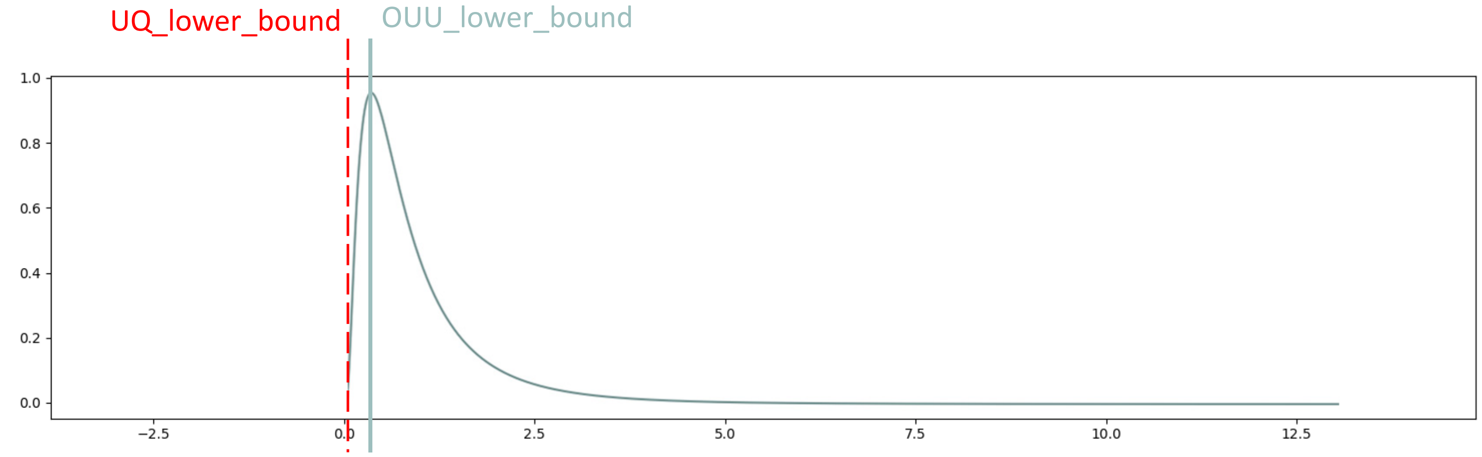
For first time users, the best practice is to ensure the following

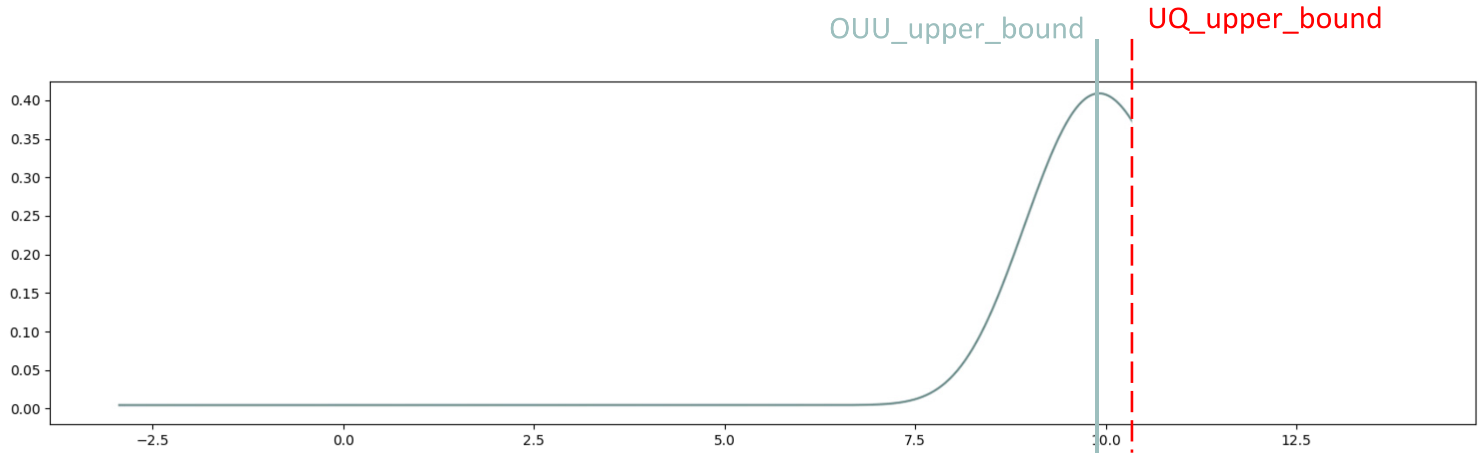
UQ_lower_bound < OUU_lower_bound

And

OUU_upper_bound < UQ_upper_bound.

For the same example, recall that the OUU bounds were between 1.0 and 10.0. The UQ bounds should be wider or outside of the OUU bounds.





More experienced and daring users will find that the recommendation is not absolute. The actual requirement is the following.

UQ_lower_bound < 0.999 quantile of the distribution when the OUU variable's mean is at OUU_lower_bound

And

UQ_upper_bound > 0.001 quantile of the distribution when the OUU variable's mean is at OUU_upper_bound