Workshop – Optimization Under Uncertainty - 3 Bar Truss, Part 2 of 2

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

Goal: Use Optimization Under Uncertainty (OUU) to Limit Failure to 5%

Initial Analysis Model Prior To Optimization

Optimal Solution

- Variables
  - x1: 1.0
  - x2: 2.0
- Objective:
  - 4.8293382800e+00
- Max probability of failure:
  - ~0.00% (Actual after UQ with 80 run LHS)

Optimization for Deterministic Responses (MSC Nastran SOL 200)

Optimal Solution

- 8 MSC Nastran Runs
- Variables
  - x1: 8.3724E-01
  - x2: 3.2830E-01
- Objective:
  - 2.696380E+00
- Max probability of failure:
  - 53.75% (Actual probability after UQ with LHS of size 80 (80 MSC Nastran runs))

Optimization for Stochastic Responses (Sandia Dakota OUU)

Optimal Solution

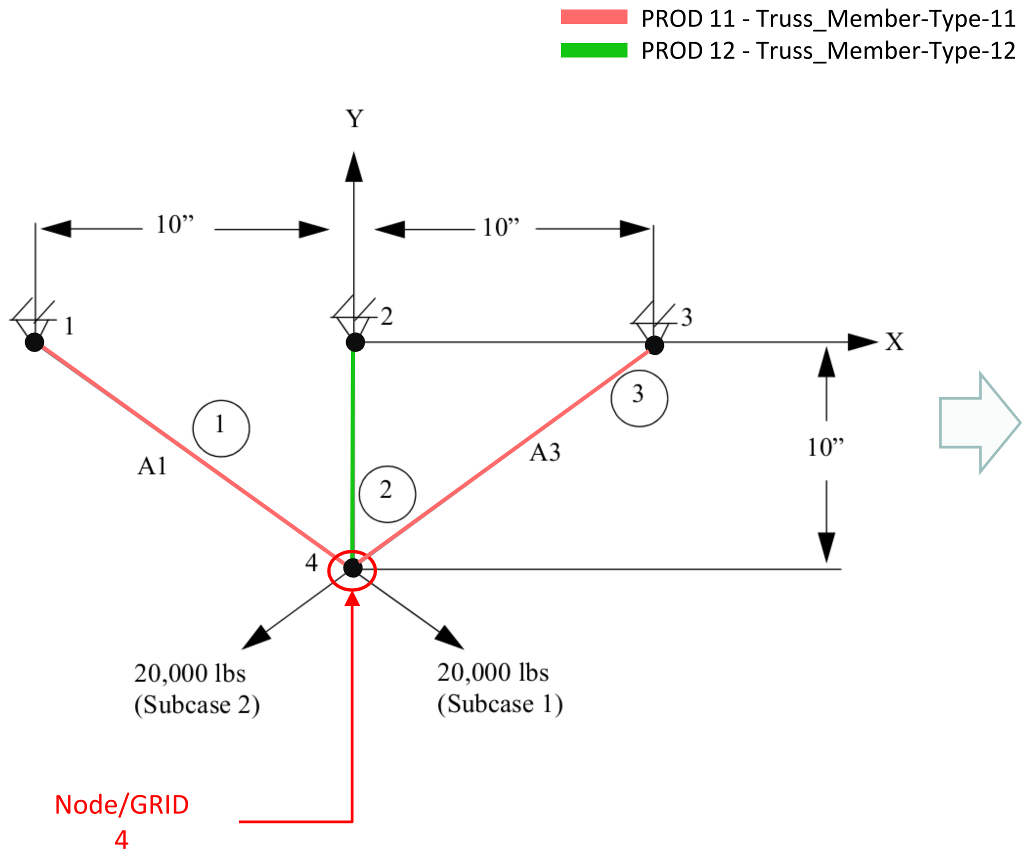
- 17 MSC Nastran Runs
- Variables
  - x1_mean: 9.5130797754e-01
  - x2_mean: 3.3670031639e-01
- Objective:
  - 3.0275633258e+00
- Max probability of failure:
  - 5.0% (Approximated probability after final OUU iteration)
  - 0.1402% (Actual probability after UQ with LHS of size 80 (80 MSC Nastran runs))

Uncertainty Quantification Problem Statement

Design Variables

x1: A of PROD 11
x2: A of PROD 12

Variable	Mean	Standard Deviation	Distribution
x1	1.	0.04	Lognormal
x2	2.	0.04	Lognormal



Responses

r1: Mass
r2: Stress in element 11, subcase 1
r3: Stress in element 13, subcase 2

Quantities of interest

r1: Mean and standard deviation (2 quantities)

r2: Mean, standard deviation, and probabilities of exceeding the bounds

r3: Mean, standard deviation, and probabilities of exceeding the bounds

Optimization Under Uncertainty (OUU) Problem Statement for Part A

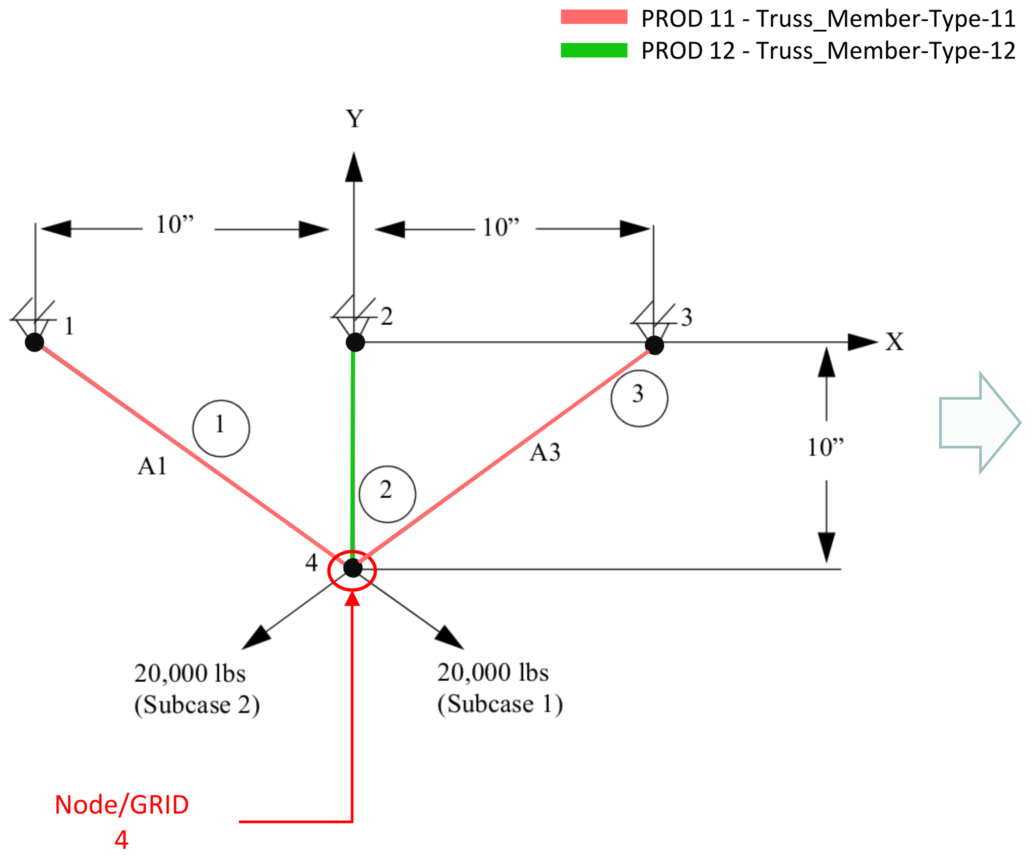
Design Variables

x1_mean: Mean of x1 (A of PROD 11)
x2_mean: Mean of x2 (A of PROD 12)

0.01 < x1_mean < 2.0

0.01 < x2_mean < 2.0

Variable	Initial Value	Lower Bound	Upper Bound
x1_mean	0.83724	0.01	2.0
x2_mean	0.32830	0.01	2.0



Objective

Minimize mean of r1 (mass)

Design Constraints

Constraints on probability of failure

~~g1: P(-15000 < r2 < 20000)~~
~~g2: P(-15000 < r3 < 20000)~~

g1: P(-14250 < r2 < 19000)
g2: P(-14250 < r3 < 19000)

g1, g2 < 0.95

95% probability of survival
or 5% probability of failure

Optimization Under Uncertainty (OUU) Problem Statement for Part B

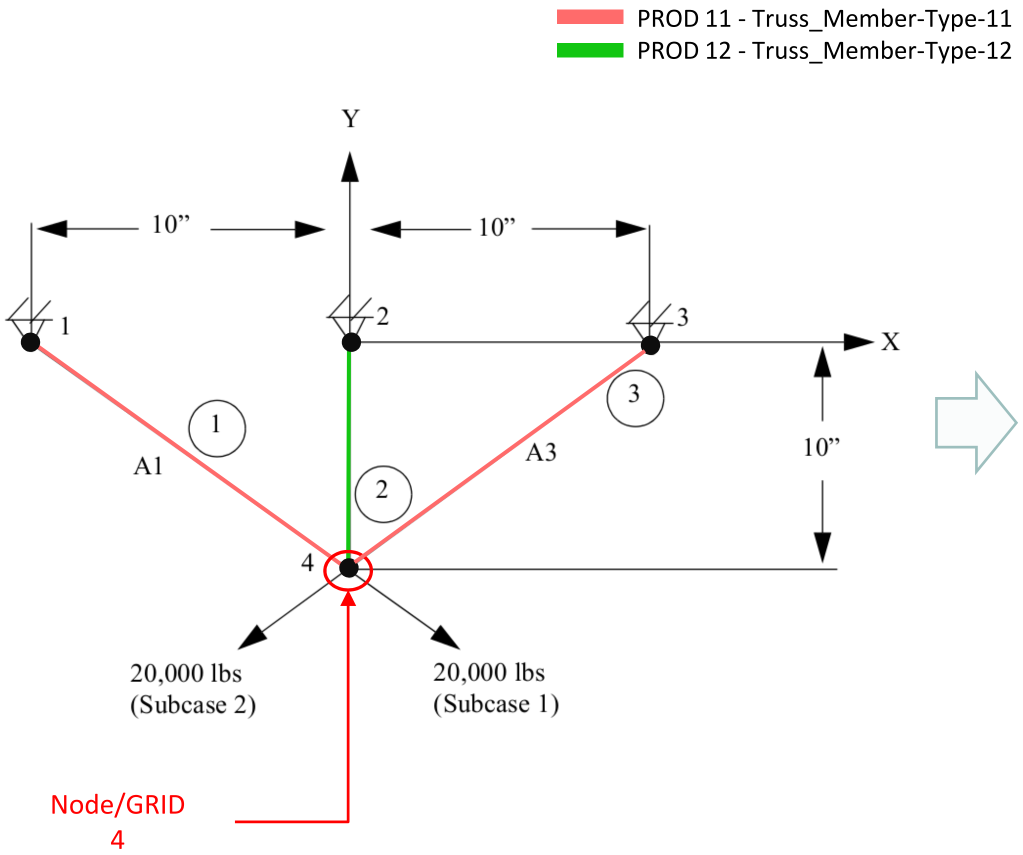
Design Variables

x1_mean: Mean of x1 (A of PROD 11)
x2_mean: Mean of x2 (A of PROD 12)

0.01 < x1_mean < 2.0

0.01 < x2_mean < 2.0

Variable	Initial Value	Lower Bound	Upper Bound
x1_mean	0.83724	0.01	2.0
x2_mean	0.32830	0.01	2.0



Objective

Minimize mean of r1 (mass)

Design Constraints

Constraints on probability of failure

~~r2_pl: P(r2 < -15000)~~
~~r2_pu: P(20000 < r2)~~
~~r3_pl : P(r3 < -15000)~~
~~r3_pu: P(20000 < r3)~~

Constraints on Reliability Indices

$\beta_{r2,-14250} < -1.644854$

$1.644854 < \beta_{r2,19000}$

$\beta_{r3,-14250} < -1.644854$

$1.644854 < \beta_{r3,19000}$

Alternative to directly constraining probabilities of failure pf < 0.05 (5%)

Why are the bounds reduced by 5%?

When this exercise was performed with bounds of -15000 and 2000, the actual final probabilities of failure were in excess of 6% and exceeded the desired probability of failure of 5%.

This is due to the following reasons.

- To reduce computational cost, the probabilities were approximated via the MVFOSM method. 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, and this is due to convergence tolerances.

To ensure the final solution is feasible and the maximum probabilities of failure are well below the desired probability of failure, the bounds are reduced by 5%. In practice, the bounds may need to be reduced anywhere between 5-20%. For this exercise with 2 variables, a 1% reduction of the bounds yielded acceptable results. A 1% reduction will not work in a majority of cases. For a 50 variable problem, a reduction up to 20% may be needed. The ideal reduction of the bounds requires trial and error, but from experience, reduction of 5, 10, 15 or 20% are worth trying. <u>The bound reduction is recommended when using the MVFOSM method.</u>

$$-15000 - -15000(0.05) = -14250$$

$$20000 - 20000(0.05) = 19000$$

Design Constraints

Constraints on probability of failure

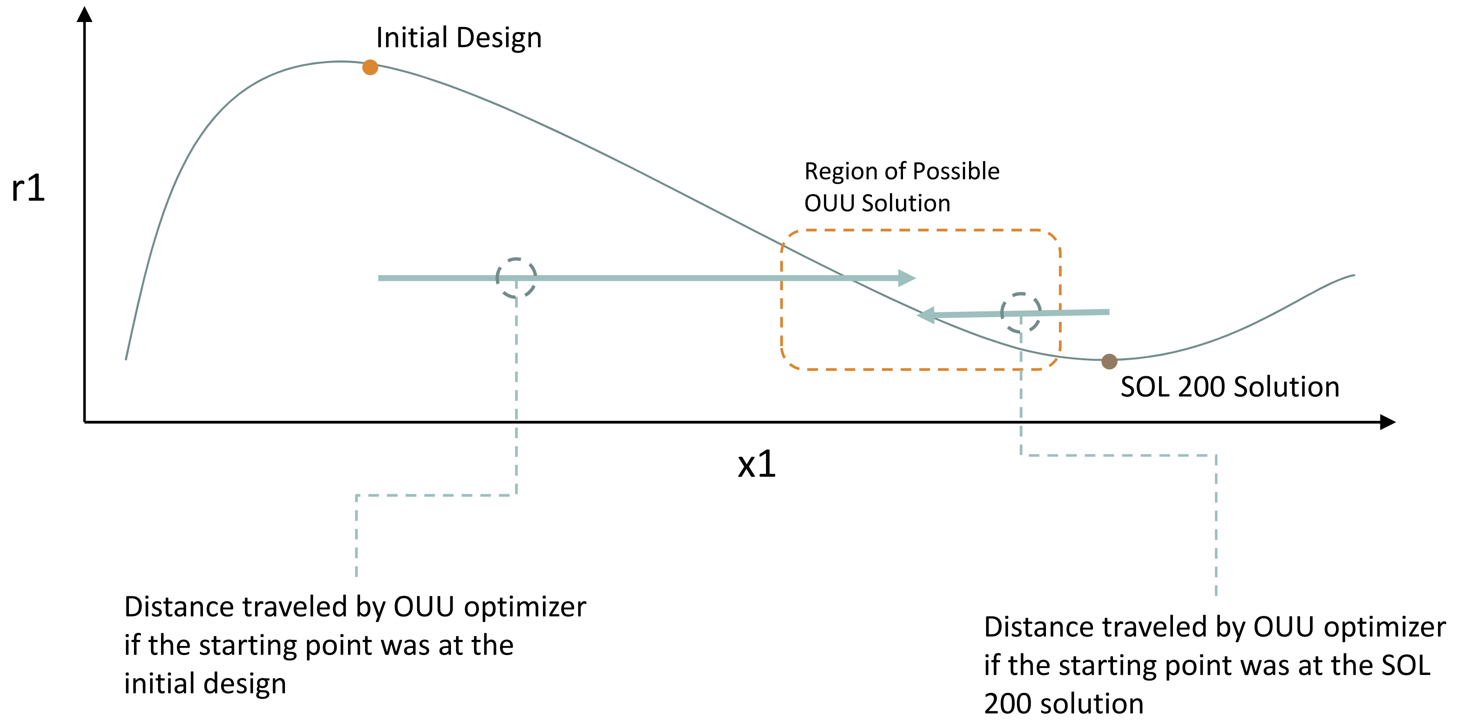
~~g1: P(-15000 < r2 < 20000)~~
~~g2: P(-15000 < r3 < 20000)~~

g1: P(-14250 < r2 < 19000)
g2: P(-14250 < r3 < 19000)

How were the initial values of the OUU variables determined?

When the input variables are certain, or deterministic, a traditional optimization with MSC Nastran SOL 200 may be performed. The solution from this optimization is termed the *SOL 200 solution*.

From experience, it is found the OUU solution is often near the SOL 200 solution. If the OUU starts at the initial design, the optimizer has to travel further and takes longer to converge. If the OUU starts at or near the SOL 200 solution, the optimizer travels less and converges faster to the OUU solution. Starting the OUU from the SOL 200 solution helps reduce the computational cost associated with OUU.



How were the initial values of the OUU variables determined?

Prior to this exercise, an MSC Nastran SOL 200 optimization was performed to yield an optimal solution (x1, x2) = (0.83724, 0.32830). The original analysis model values were (x1, x2) = (1.0, 2.0).

The OUU was configured in 2 separate ways. In trial 1, the initial values of the OUU variables were equal to the original analysis model values (1.0, 2.0). In trial 2, the initial values of the OUU variables were equal to the optimal solution values (0.83724, 0.32830).

When the initial values from a SOL 200 optimization are used, see trial 2, the optimizer converges faster during OUU than trial 1. Why? It is reasoned that in trial 1, the optimizer has to travel further to reach the optimal solution. The SOL 200 solution is likely to be close to the OUU solution, so the optimizer during OUU would have to travel a smaller distance to the optimal solution and would require fewer MSC Nastran runs.

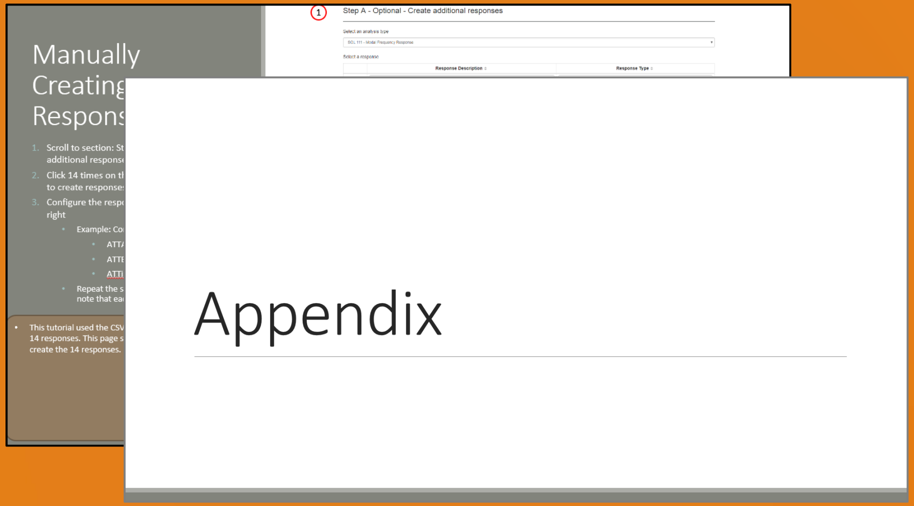
Prior to OUU, it is recommended to perform a local optimization or global optimization to determine ideal initial values for OUU variables.

OUU Trial	Initial Values	Comments	Number of MSC Nastran Runs to Converge During OUU	OUU Solution
Trial 1	x1_mean = 1.0 x2_mean = 2.0	These were the original analysis model values with no prior optimization.	30	x1_mean = .91407 x2_mean = .32415 Objective = 2.9094
Trial 2	x1_mean = 0.83724 x2_mean = 0.32830	The initial values are based on the optimal solution after an MSC Nastran SOL 200 optimization.	17	x1_mean = .95131 x2_mean = .33670 Objective = 3.0276

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

Optimization Under Uncertainty - Traditional optimization assumes both the inputs and outputs of black box functions, such as FEA solvers, are deterministic and certain. When inputs or variables of a finite element analysis models are uncertain, the responses are stochastic or random. The uncertainty in inputs and outputs poses challenges to traditional optimization. This exercise details the process of configuring an optimization under uncertainty, which addresses variables and responses that are uncertain.

SOL 200 Web App Capabilities

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

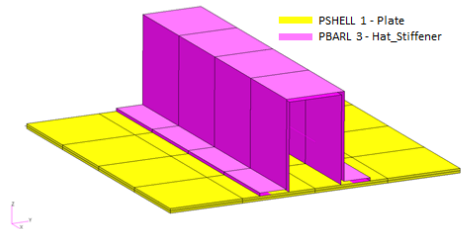
Compatibility

- 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.

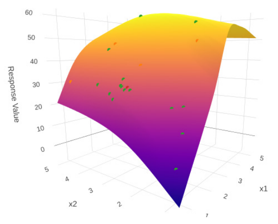
Benefits

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

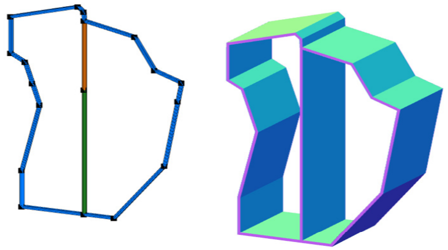
Web Apps



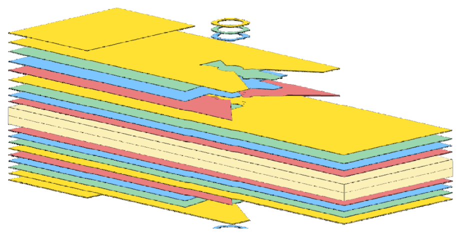
Web Apps for MSC Nastran SOL 200
Pre/post for MSC Nastran SOL 200. Support for size, topology, topometry, topography, multi-model optimization.



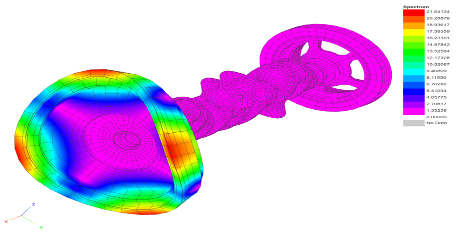
Machine Learning Web App
Bayesian Optimization for nonlinear response optimization (SOL 400)



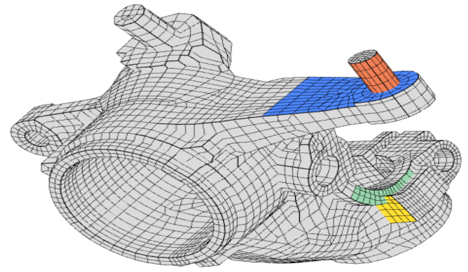
PBMSECT Web App
Generate PBMSECT and PBRSECT entries graphically



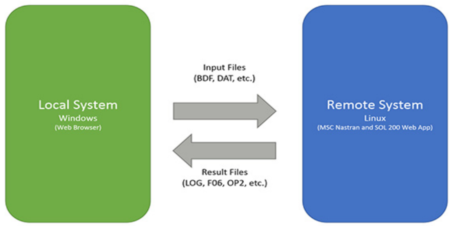
Ply Shape Optimization Web App
Optimize composite ply drop-off locations, and generate new PCOMPG entries



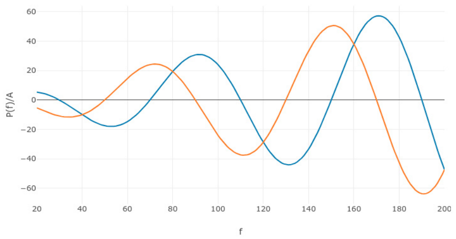
Post-processor Web App
View MSC Nastran results in a web browser on Windows and Linux



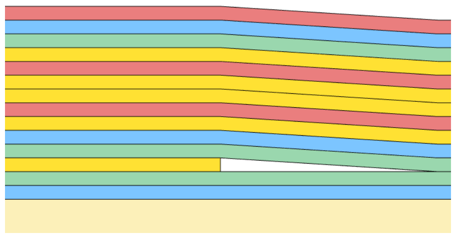
Shape Optimization Web App
Use a web application to configure and perform shape optimization.



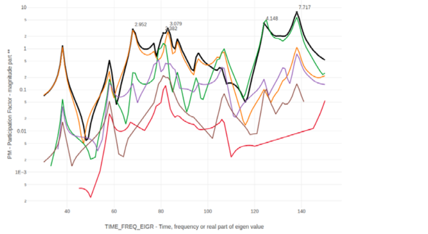
Remote Execution Web App
Run MSC Nastran jobs on remote Linux or Windows systems available on the local network



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



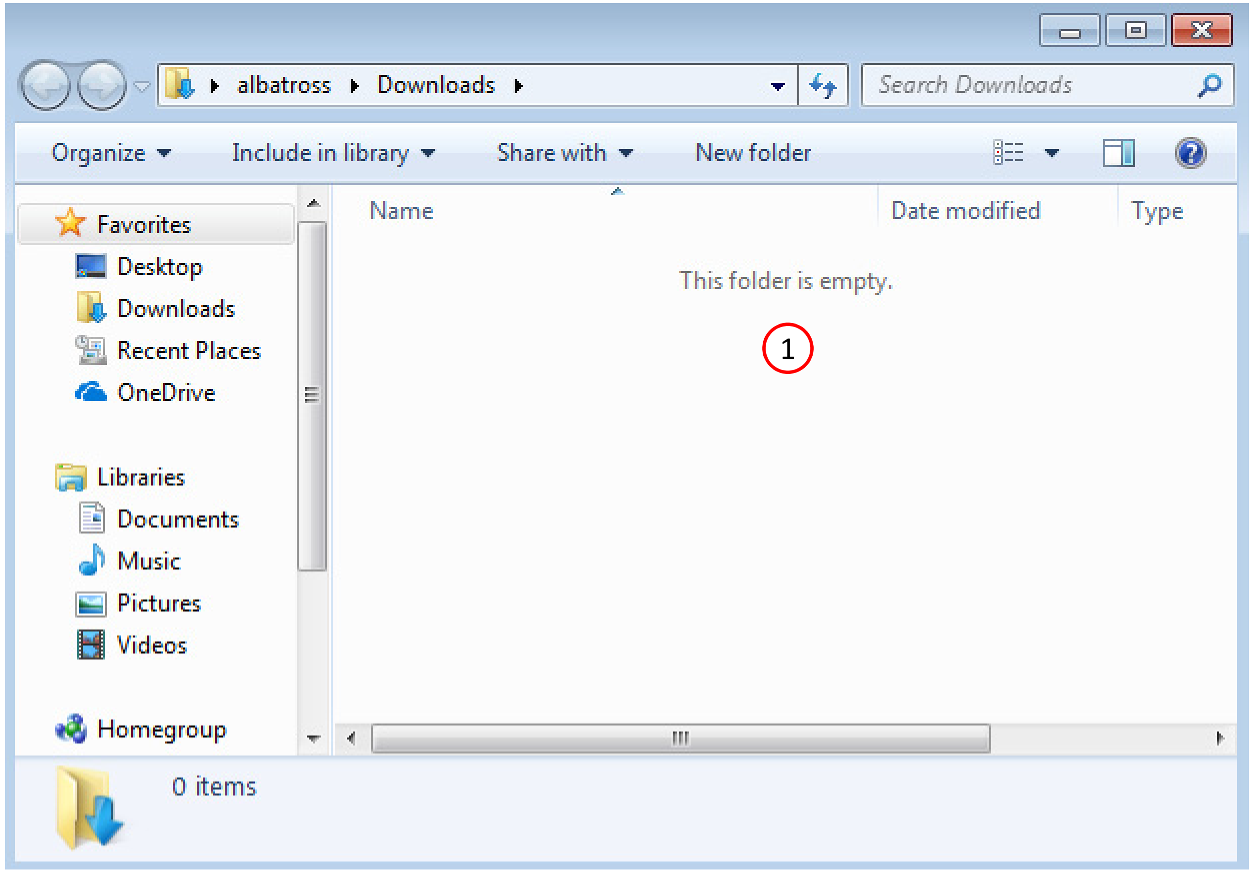
Stacking Sequence Web App
Optimize the stacking sequence of composite laminate plies



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

Before Starting

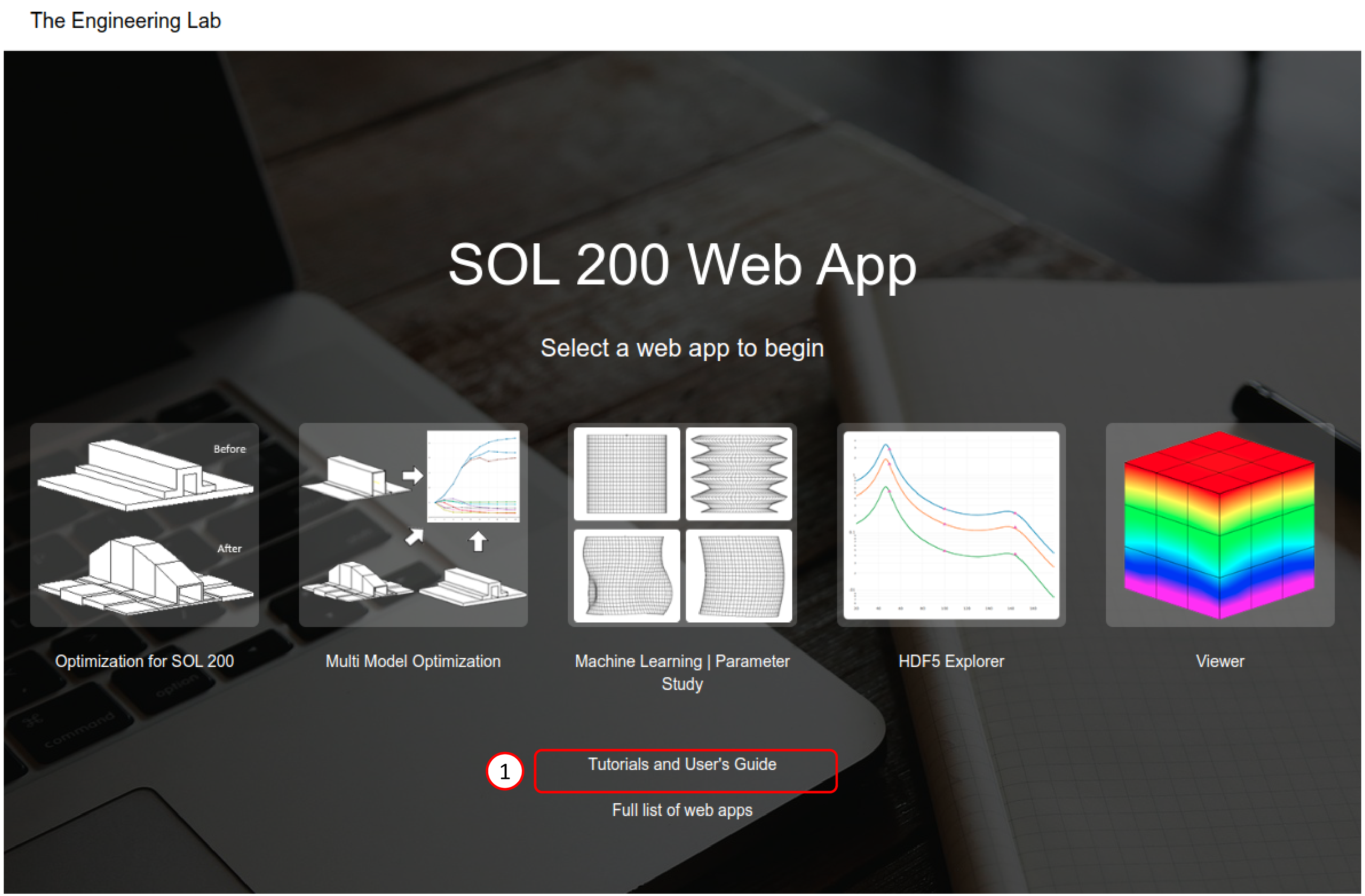
1. Ensure the Downloads directory is empty in order to prevent confusion with other files



Go to the User's Guide

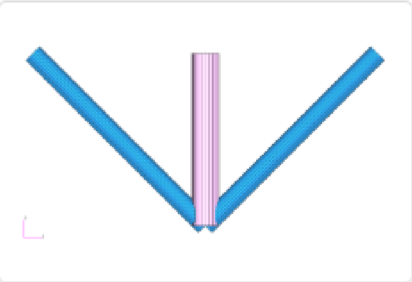
1. Click on the indicated link

- 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



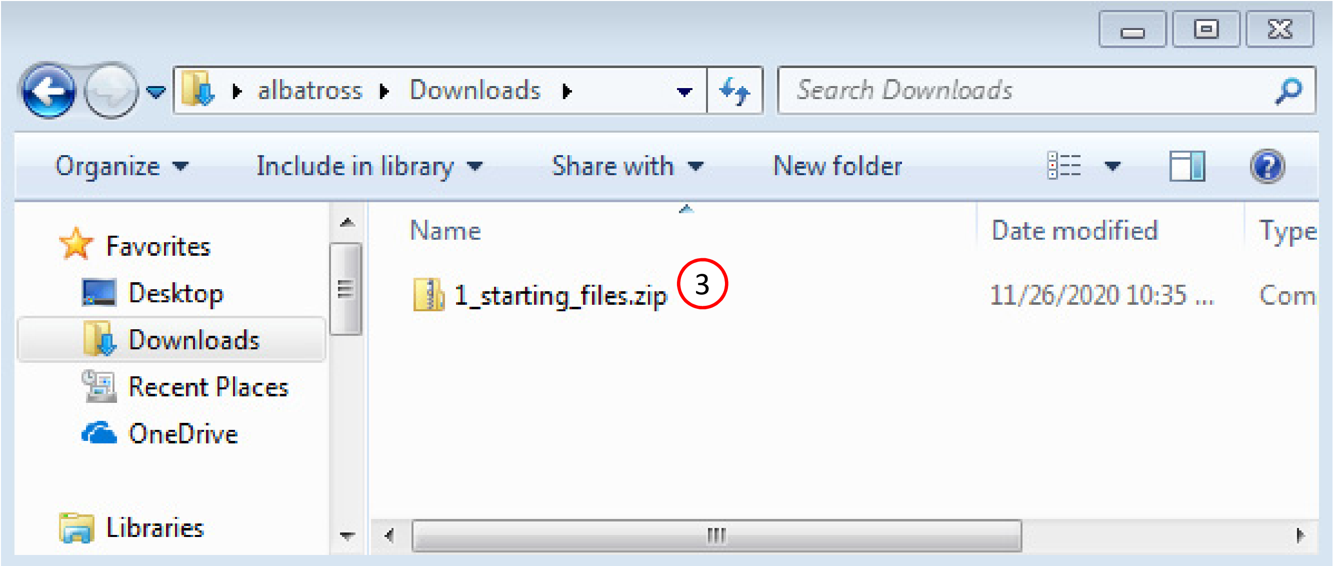
Optimization Under Uncertainty - 3 Bar Truss, Part 2 of 2 (1)

This example details the process to configure and perform an optimization under uncertainty (OUU) for a 3-bar truss. Also, details are included regarding how to interpret the OUU results and final probabilities of failure. The uncertainty quantification (UQ) is performed via the mean value first-order second-moment (MVFOSM) method. This is part 2 of a 2-part tutorial.

The optimization problem statement is to minimize the mean mass while ensuring the stress constraints have a probability of failure no greater than 5%.

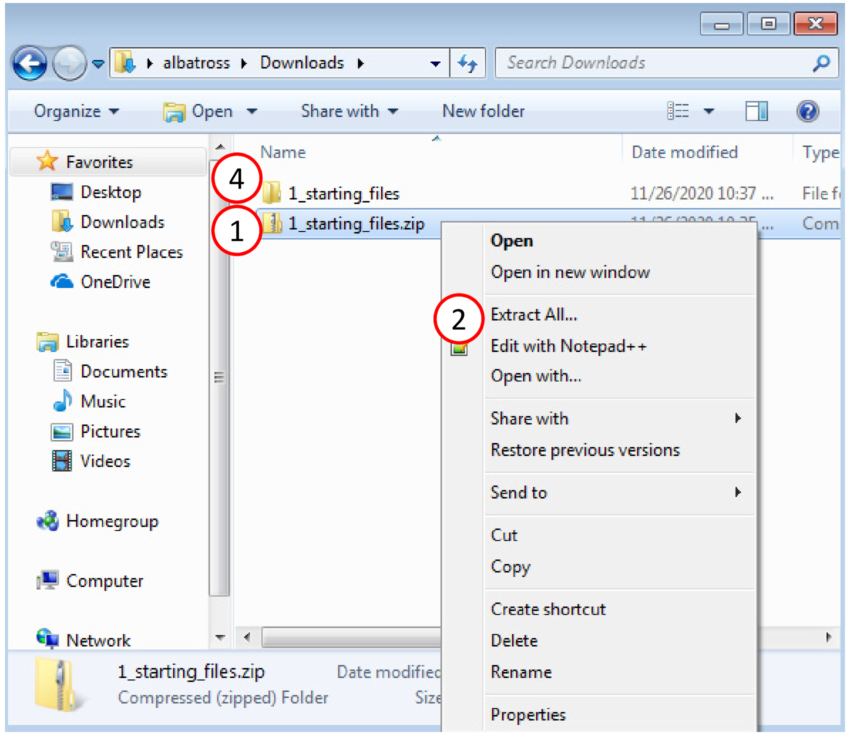
MSC Nastran is used to perform the finite element analysis and acquire responses and gradients. Sandia Dakota is used to perform the UQ and OUU. The SOL 200 Web App is used to configure the OUU and will automatically exchange responses and gradients between MSC Nastran and Sandia Dakota.

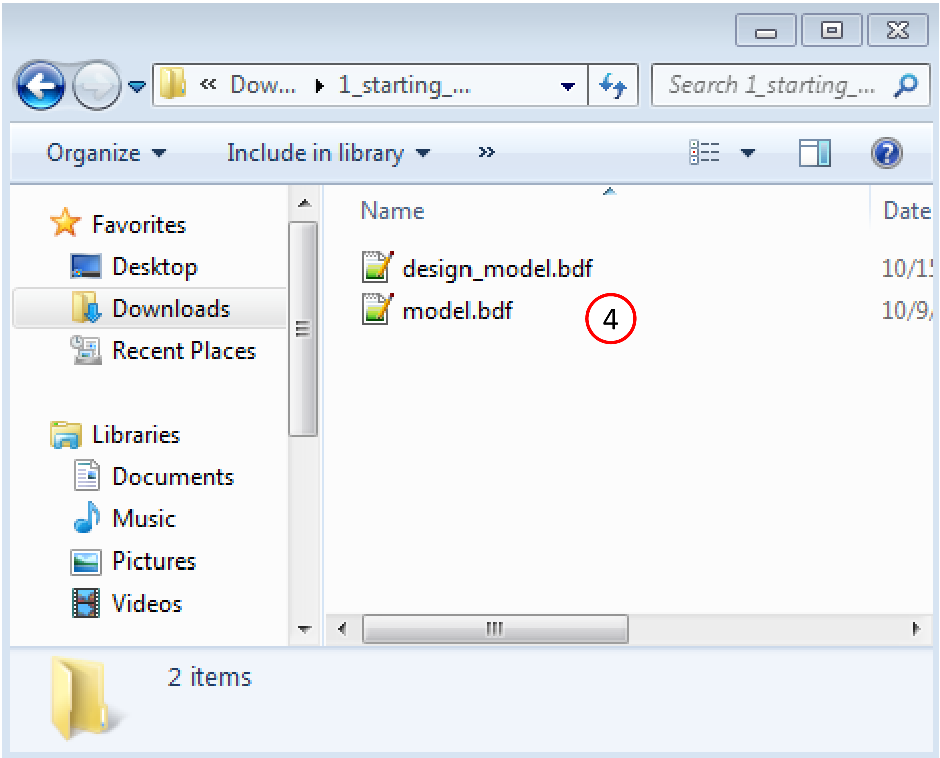
Starting BDF Files: Link (2)
Solution BDF Files: Link

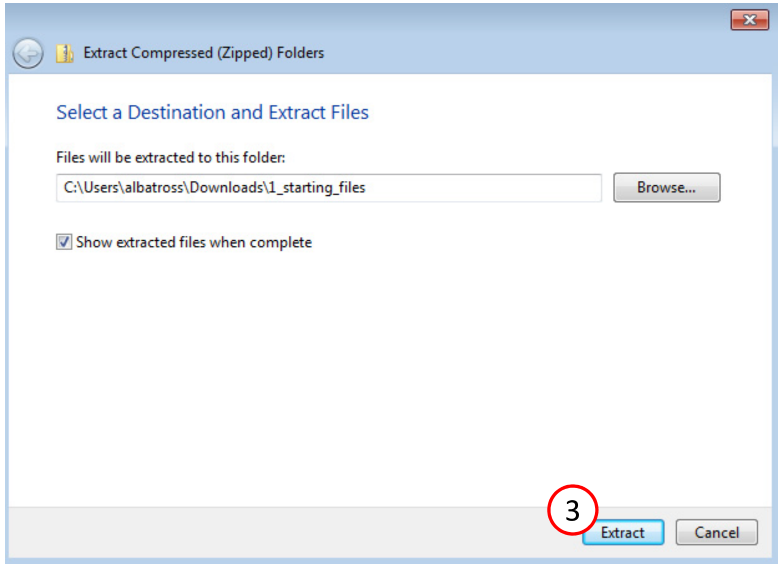


Obtain Starting Files

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



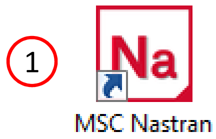


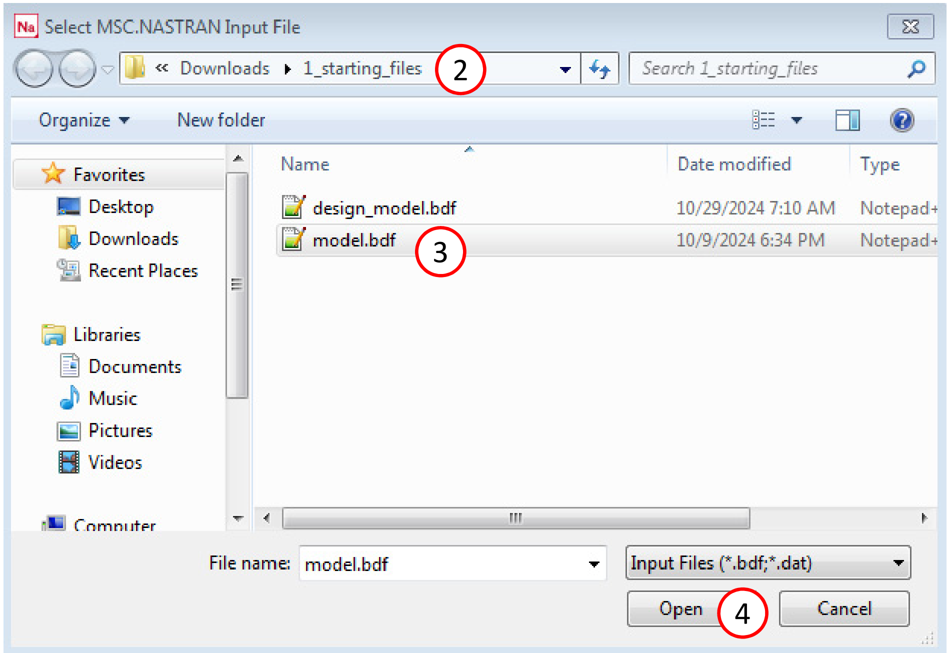


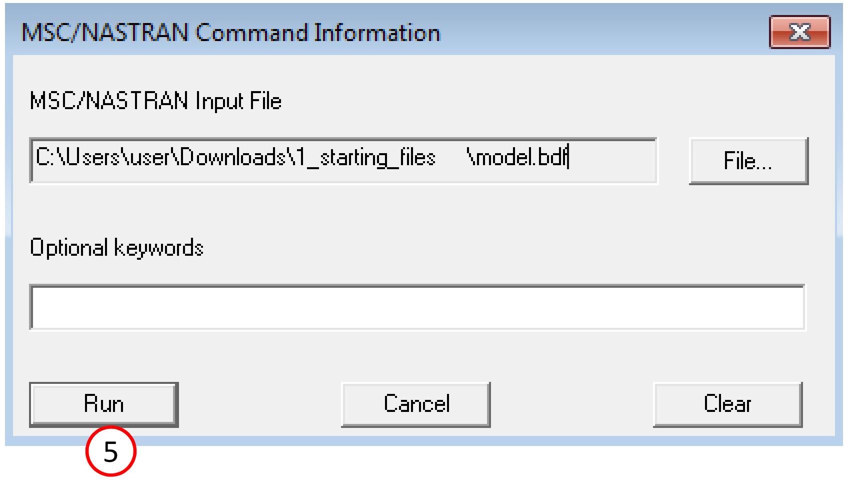
Create the Starting H5 File

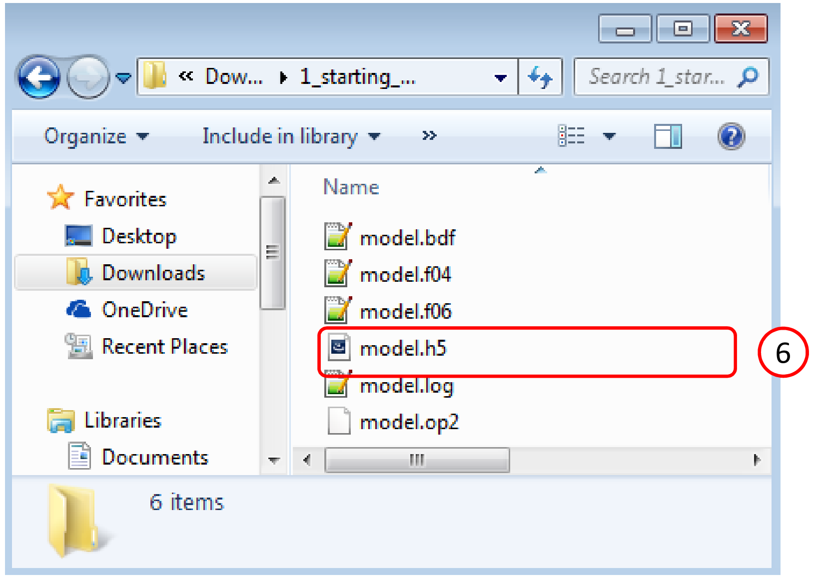
A starting H5 file must be created. This H5 file will be used to configure the responses later on.

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 – Optimization Under Certainty, Constraining Probaiblities

Open the Correct Page

1. Click on the indicated link

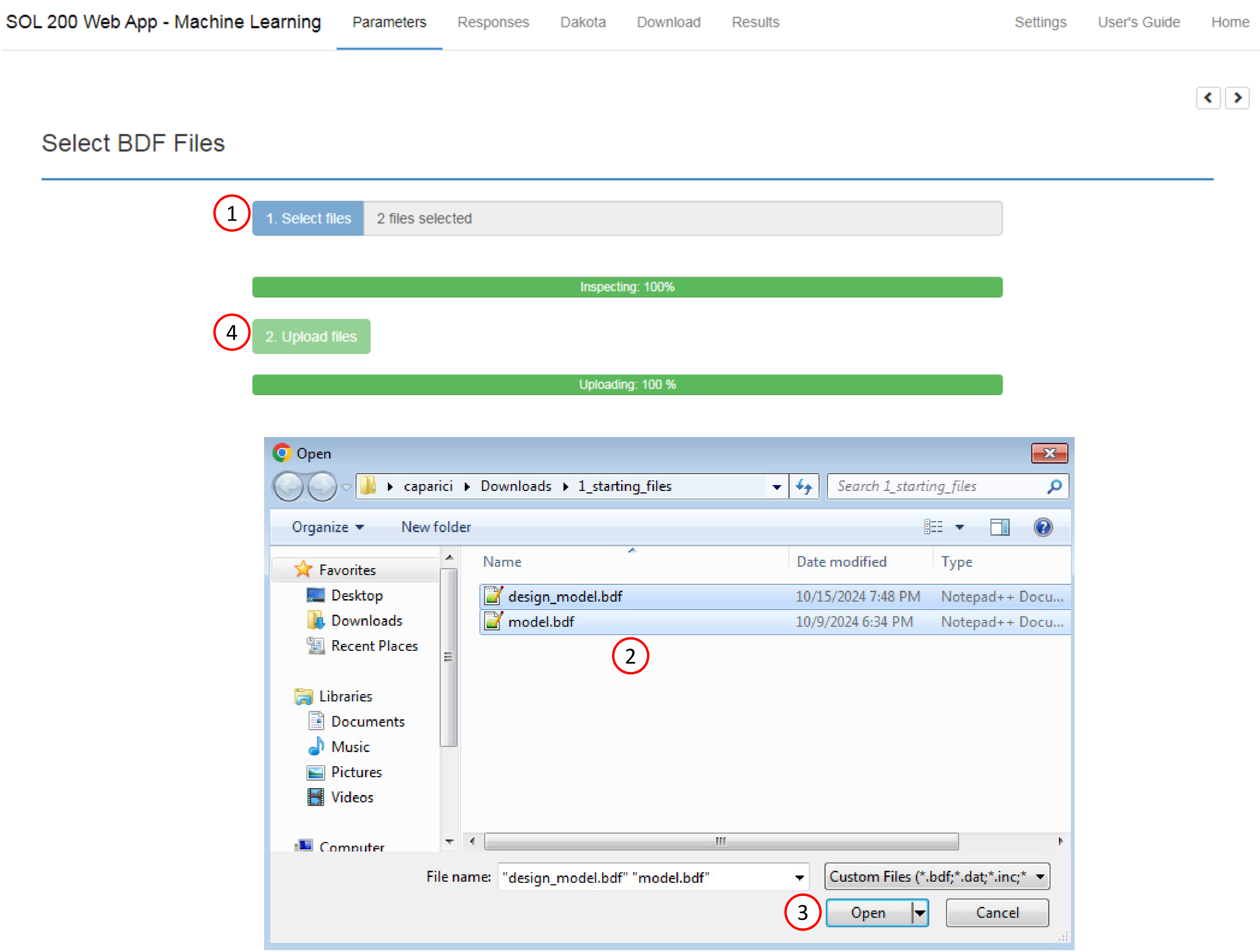
- MSC Nastran can perform many optimization types. The SOL 200 Web App includes dedicated web apps for the following:
    - Optimization for SOL 200 (Size, Topology, Topometry, Topography, Local Optimization, Sensitivity Analysis and Global Optimization)
    - Multi Model Optimization
    - Machine Learning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

The Engineering Lab

Select BDF Files

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

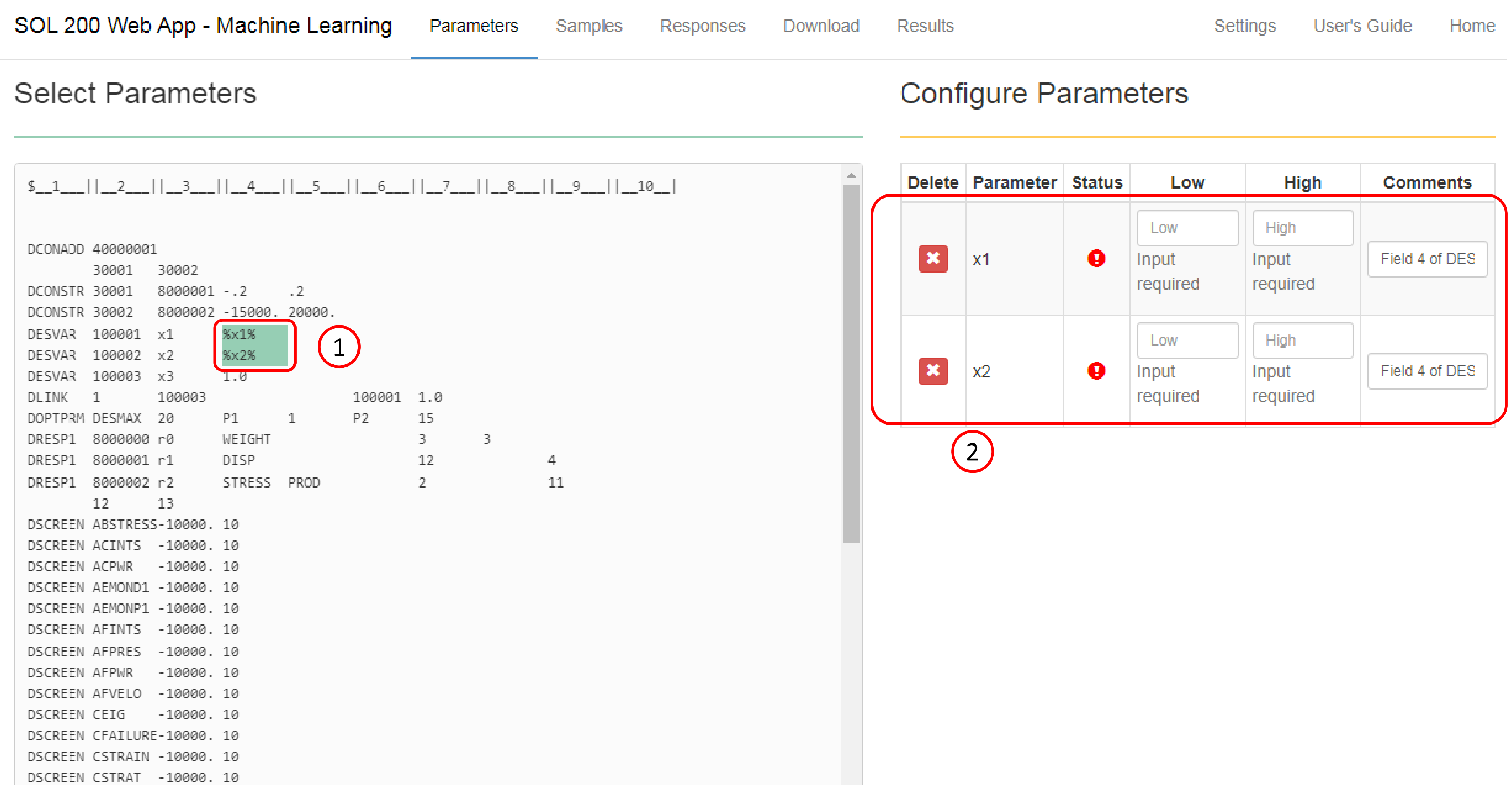
- When starting the procedure, all the necessary BDF, or DAT, files must be collected and uploaded together. Relevant INCLUDE files must also be collected and uploaded.



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

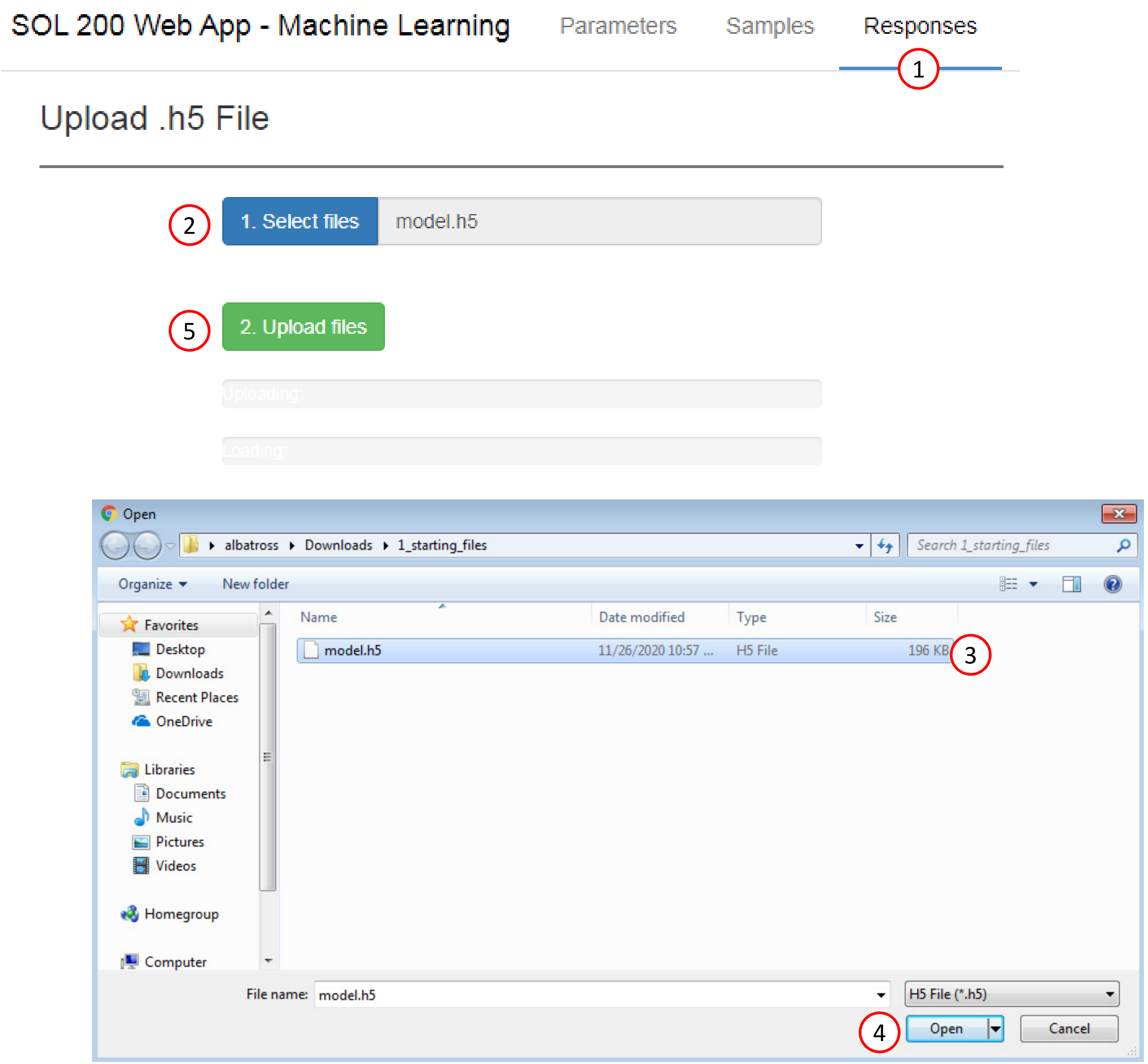
- If gradients are expected to be provided to Dakota, select only the initial values of DESVAR entries.
  - When the initial values of DESVASR entries are selected, only the independent DESVAR entries should be selected. In this example, DESVAR x3 is dependent on DESVAR x1. Variables x1 and x2 are both independent and are selected.
- If gradients are not expected, any other field with real values may be selected.



Responses

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

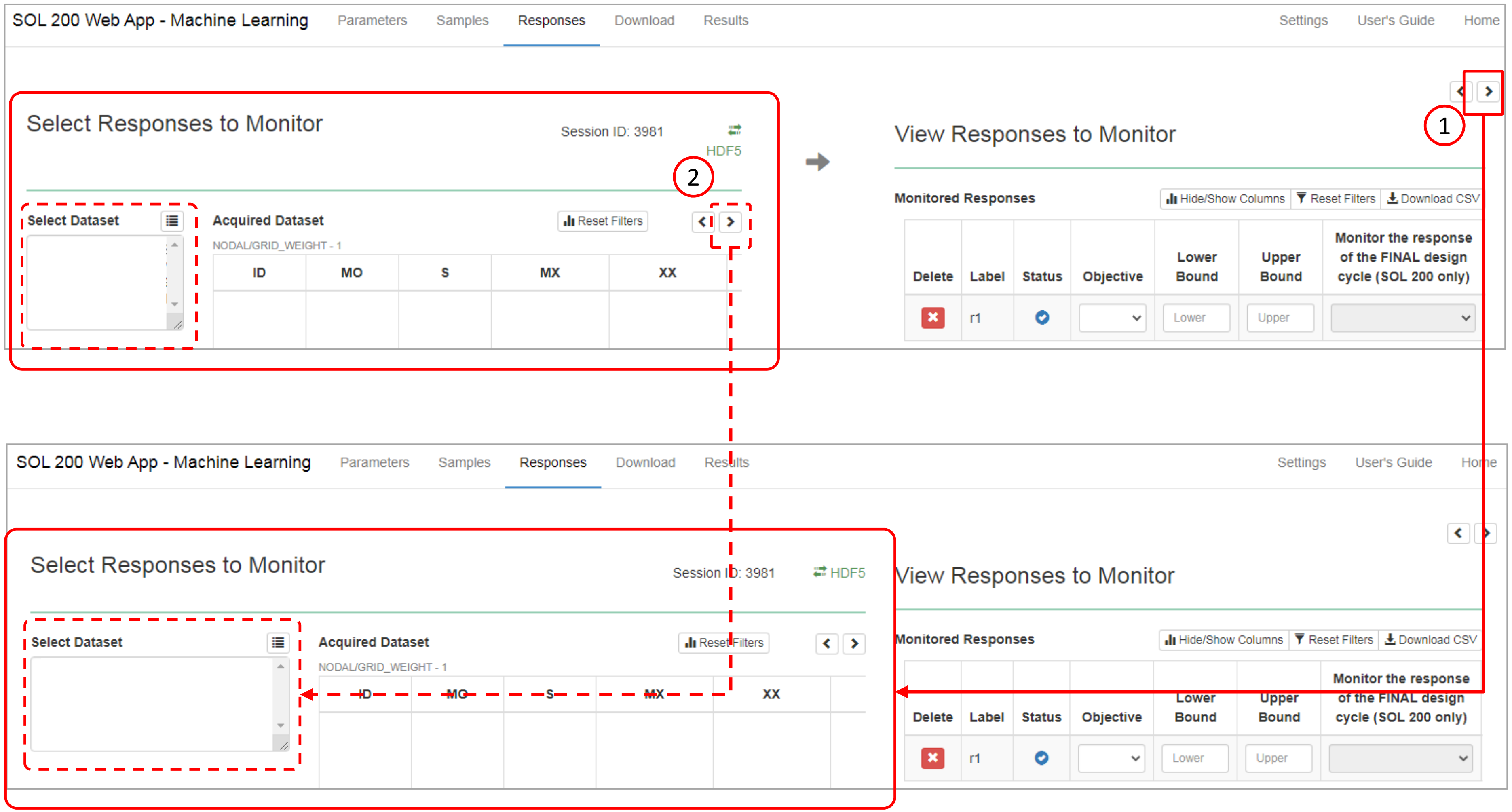
- On this page, the H5 file is uploaded to the web app.



Adjust the Column Width

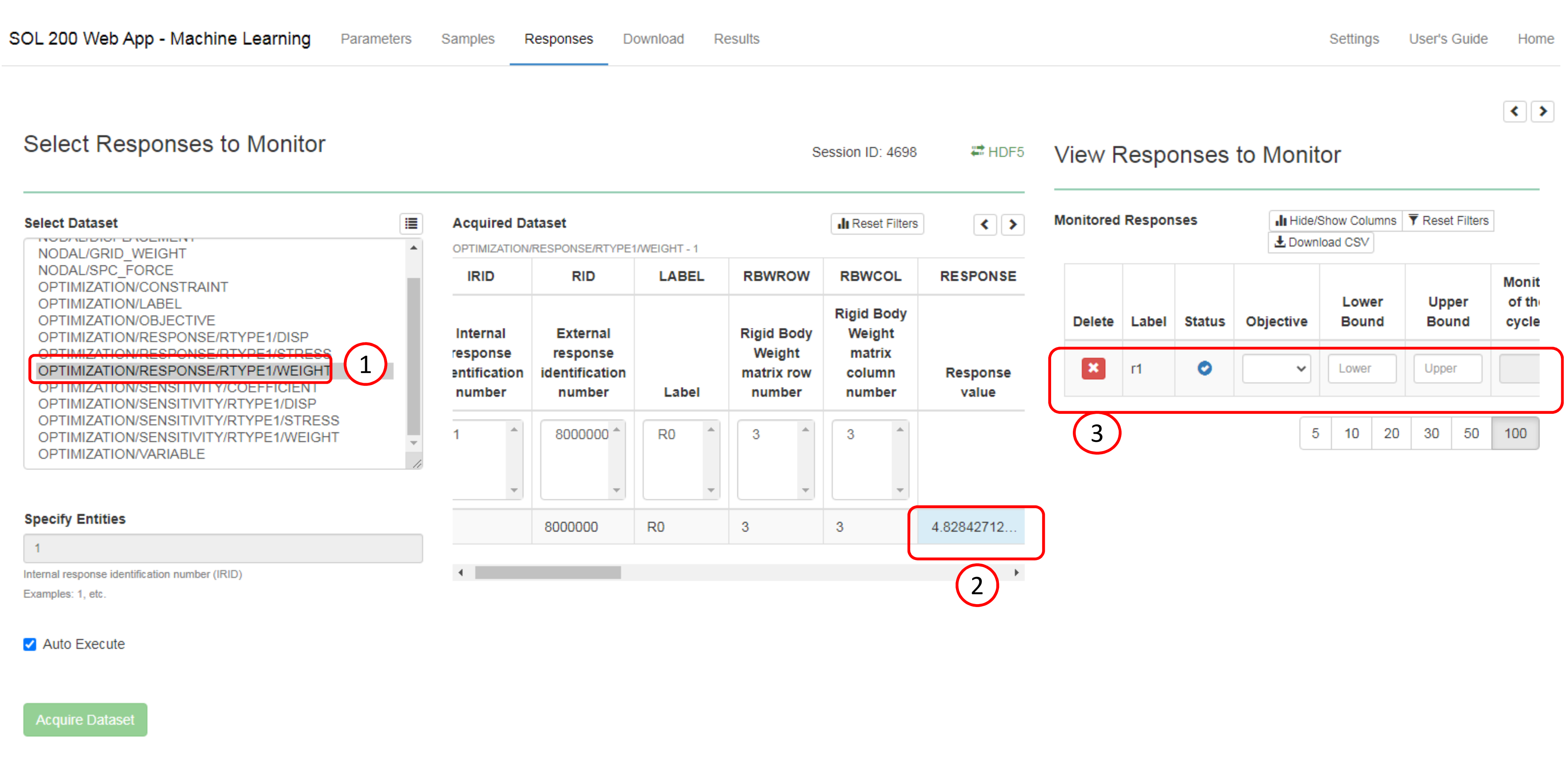
1. Optional - Use at your liking the buttons at the top right hand corner to adjust the width of the left and right columns
2. Optional – Use the indicated buttons to adjust the width of the column Select Dataset

- 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/RTYPE1/WEIGHT
2. Select the indicated cell
3. The newly created Response to Monitor is listed as r1



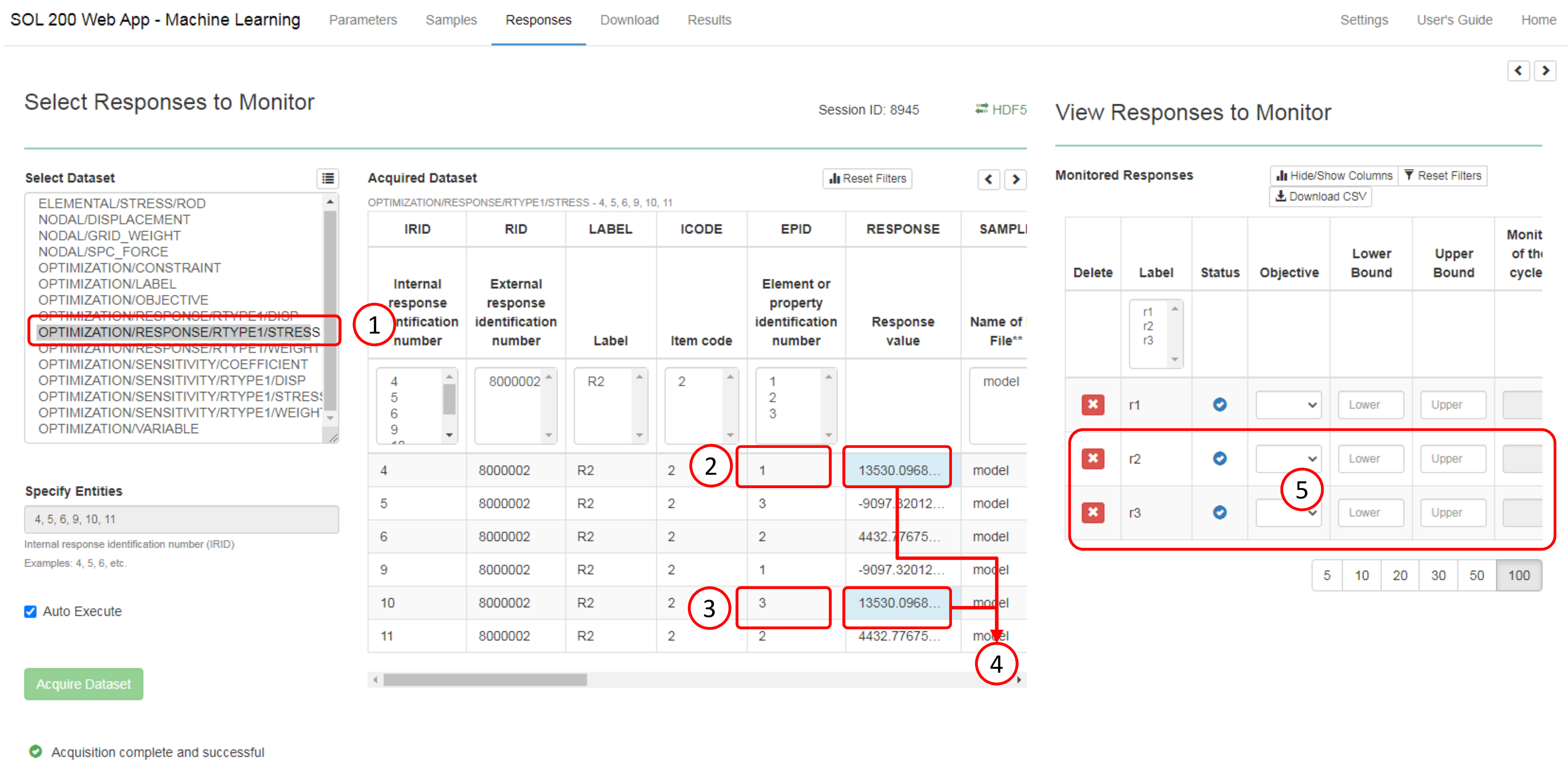
If Dakota expects gradients to be provided, ONLY responses from the following datasets may be selected:

- OPTIMIZATION/RESPONSE/RTYPE1/*
- OPTIMIZATION/RESPONSE/RTYPE2/*
- OPTIMIZATION/RESPONSE/RTYPE3/*

If gradients are not expected to be provided, responses from other datasets may be selected.

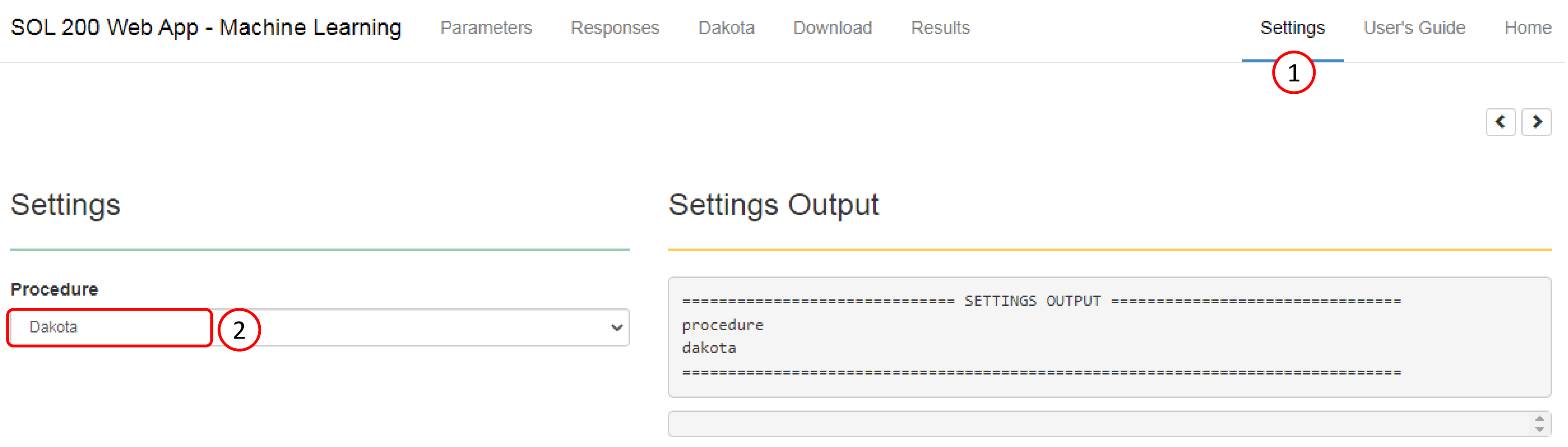
Select Responses

1. Select the following dataset: ELEMENTAL/RESPONSE/RTYPE1/STRESS
2. Notice that an axial stress for element 1 and subcase 1 is available
3. Notice that an axial stress for element 3 and subcase 2 is available
4. Select the indicated cells
5. New responses r2 and r3 have been created



Settings

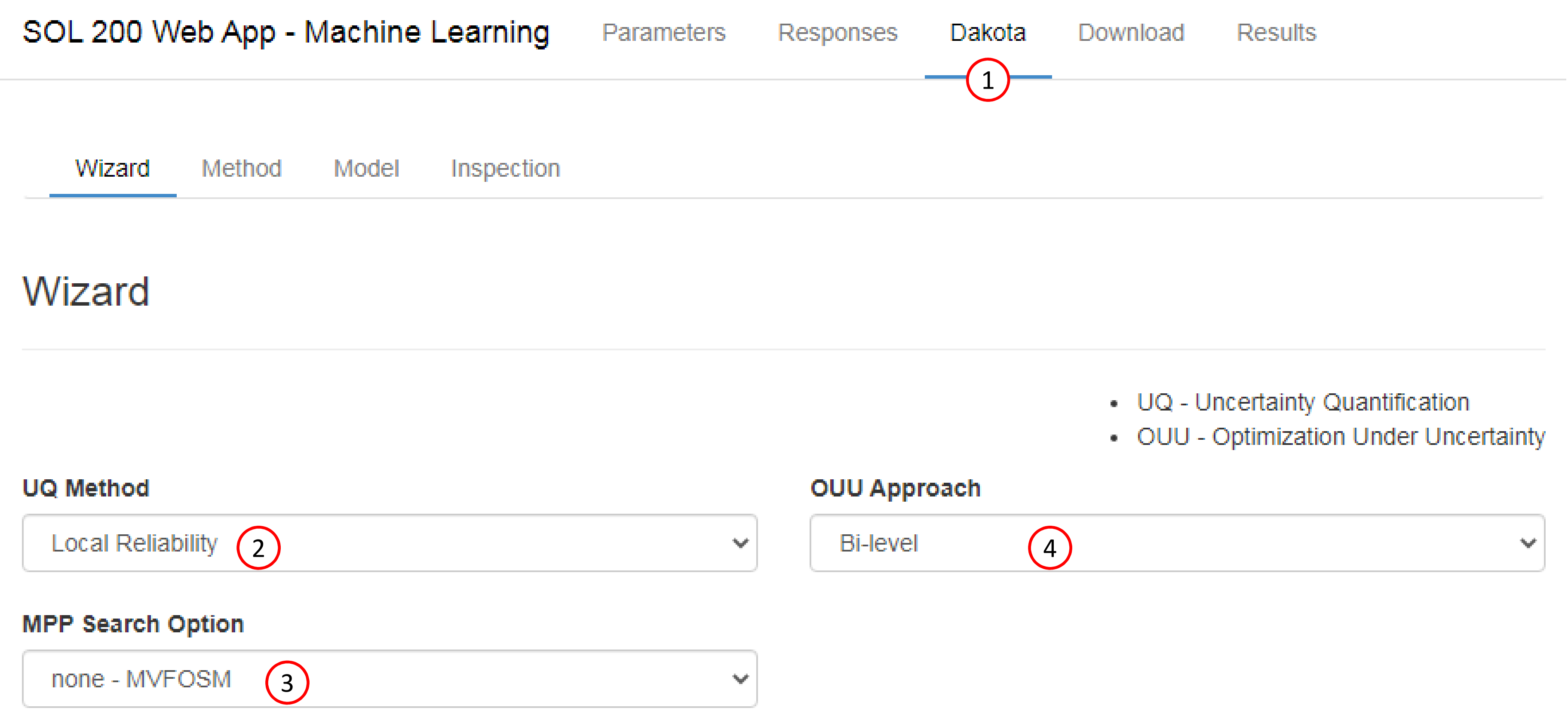
1. Click Settings
2. Set Procedure to Dakota



Dakota

1. Click Dakota
2. Set UQ Method to Local Reliability
3. Set MPP Search Option to none - MVFOSM
4. Set OUU Approach to Bi-level

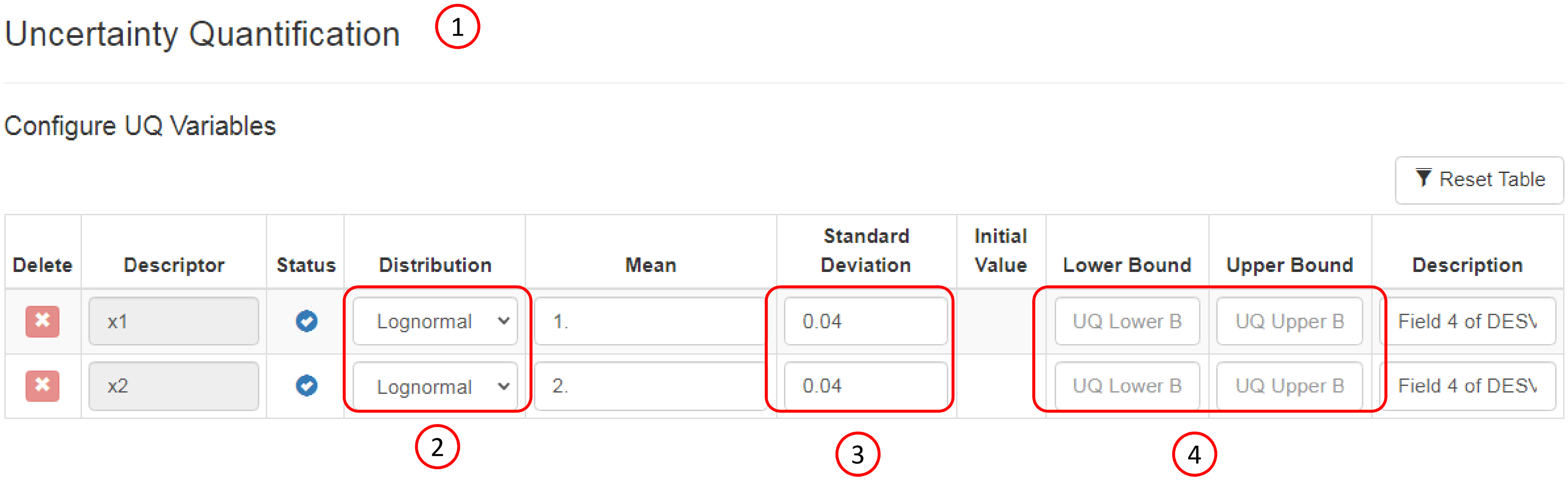
- Reliability, or local reliability, methods refers to a group of techniques to determine the tail probabilities of normally distributed responses and requires the availability of gradients for the responses. Reliability methods can employ the MVFOSM method to approximate the tail probabilities or can employ MPP search methods to determine the tail probabilities. The OUU approach refers to how often the optimizer runs the black box function. Readers are referred to the Dakota User's Manual and Theory Manual for more information.
- Outside of this exercise, a previous LHS of size 40 (40 MSC Nastran runs) was performed and revealed the responses were normally distributed. MSC Nastran SOL 200 can also output gradients/sensitivities for the responses. Reliability methods may be used during the OUU. Since the standard deviations of the UQ variables are small, the Mean Value First-Order Second-Moment Method (MVFOSM) yields an acceptable level of approximating the tail probabilities.



Uncertainty Quantification (UQ)

1. Scroll to section Uncertainty Quantification
2. Set both distributions to Lognormal Uncertain
3. Set both standard deviations to 0.04
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*.



Delete	Descriptor	Status	Distribution	Mean	Standard Deviation	Initial Value	Lower Bound	Upper Bound	Description
✖	x1	✔	Lognormal	1.	0.04		UQ Lower B	UQ Upper B	Field 4 of DES\
✖	x2	✔	Lognormal	2.	0.04		UQ Lower B	UQ Upper B	Field 4 of DES\

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: 8.3724E-1
   - Lower Bound: .01
   - Upper Bound: 2.
4. For x2_mean, set the following:
   - Initial Value: 3.2830E-1
   - Lower Bound: .01
   - Upper Bound: 2.

Optimization Under Uncertainty (1)

Select OUU Variables

Descriptor	Continuous	Mean	Description
x1		+ Mean	Field 4 of DESVAR 100001
x2		+ Mean	Field 4 of DESVAR 100002

(2)

Configure OUU Variables

Reset Table

Delete	Descriptor	Status	Initial Value	Lower Bound	Upper Bound	Description
✖	x1_mean	✔	8.3724E-1	.01	2.	Mean - Field 4 of DESVAR 100001
✖	x2_mean	✔	3.2830E-1	.01	2.	Mean - Field 4 of DESVAR 100002

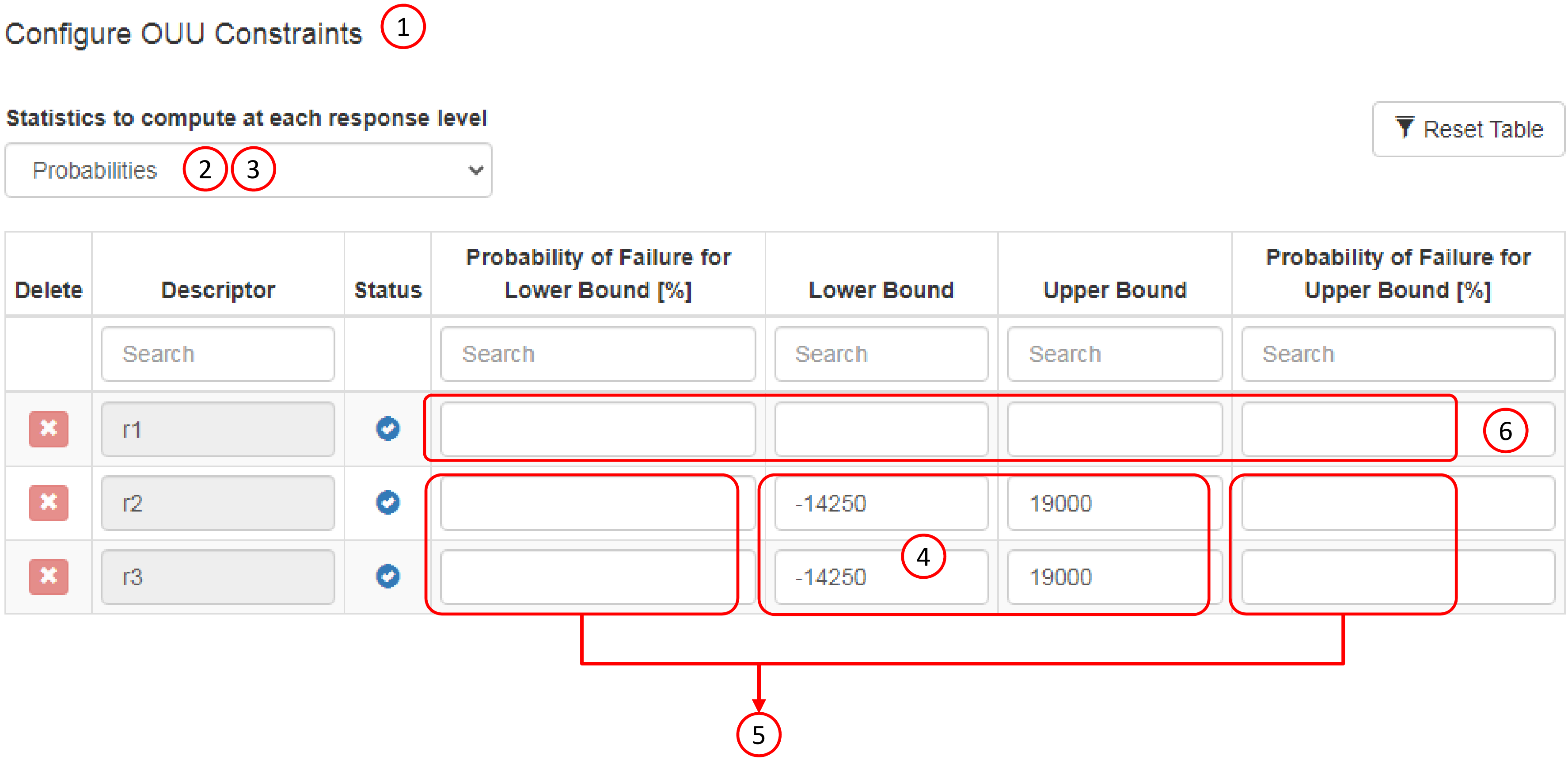
(3) (4)

Optimization Under Uncertainty (OUU)

1. Scroll to section Configure OUU Constraints
2. Set Statistics to compute at each response level to Probabilities
3. Ensure Probabilities has been selected. This is an often overlooked step.
4. Set the following bounds on the stress responses
   - Lower Bound: -14250
   - Upper Bound: 19000
5. Do NOT provide any constraint information for the probabilities of failure. The constraints on probability of failure are defined on the next page.
6. Do NOT provide any constraint information for response r1, i.e. do NOT constrain the weight response.

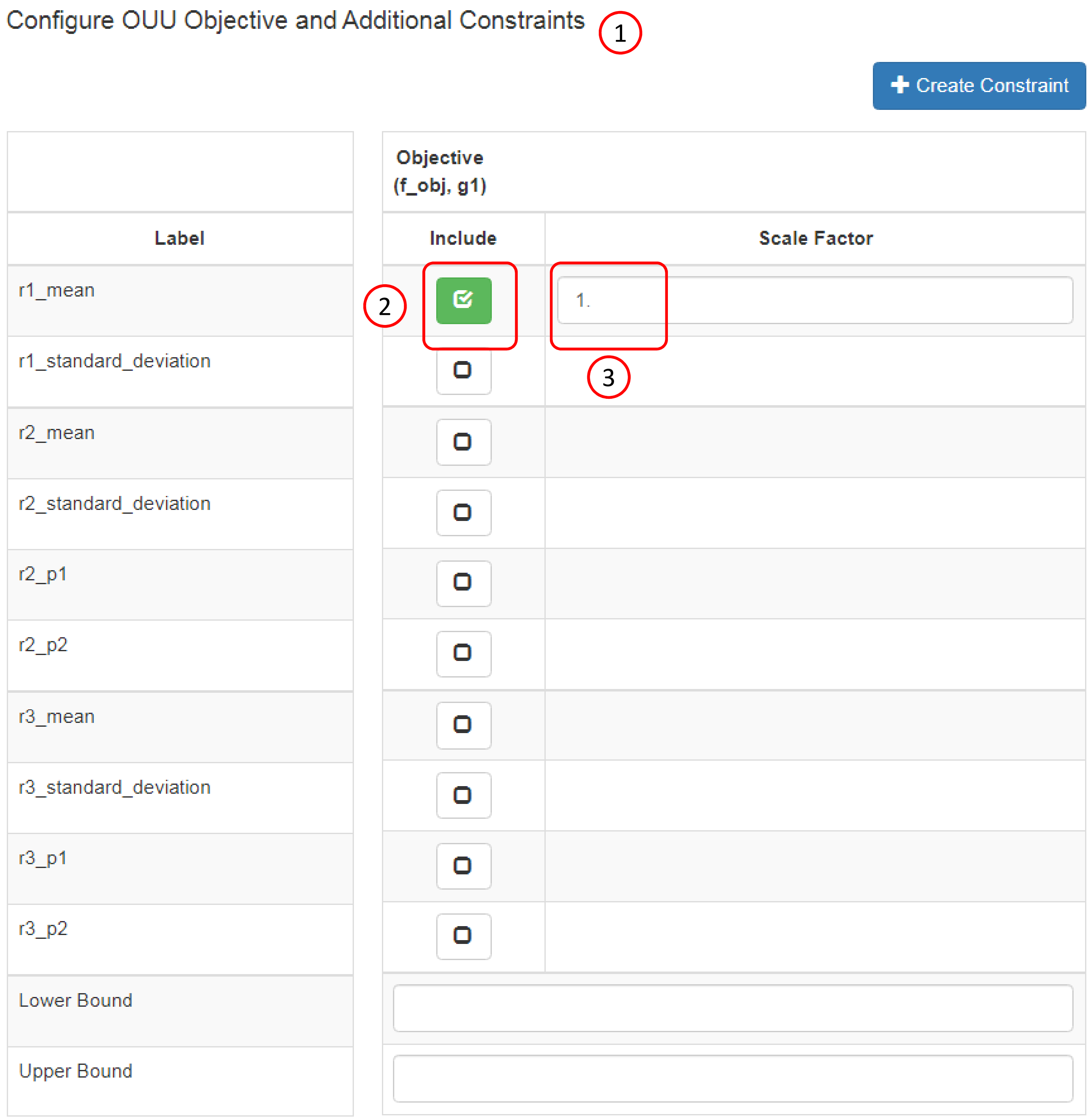
- After a Dakota uncertainty quantification, a mean and standard deviation (STD) is output for each response. If a lower and upper bound is provided, probabilities of exceeding the bounds are also output after the uncertainty quantification.

Response	Number of Statistics Output
r1	2 (Mean, STD)
r2	4 (Mean, STD, 2 probabilities)
r3	4 (Mean, STD, 2 probabilities)



Dakota - Optimization Under Uncertainty (OUU)

1. Scroll to section Configure OUU Objective and Additional Constraints
2. Click the indicated button to include the mean of response r1 (weight) in the objective.
3. Ensure the scale factor is 1.0 or +1.0, i.e. minimize the weight. If there is a desire to maximize the objective, a negative scale factor should be used.



Output Cumulative Distribution Function Values

1. Click Method
2. Click Display Selected Keywords
3. Find the distribution keyword
4. Set the value to cumulative
5. Click Wizard

The goal is to define constraints of this form, which requires CDF values.

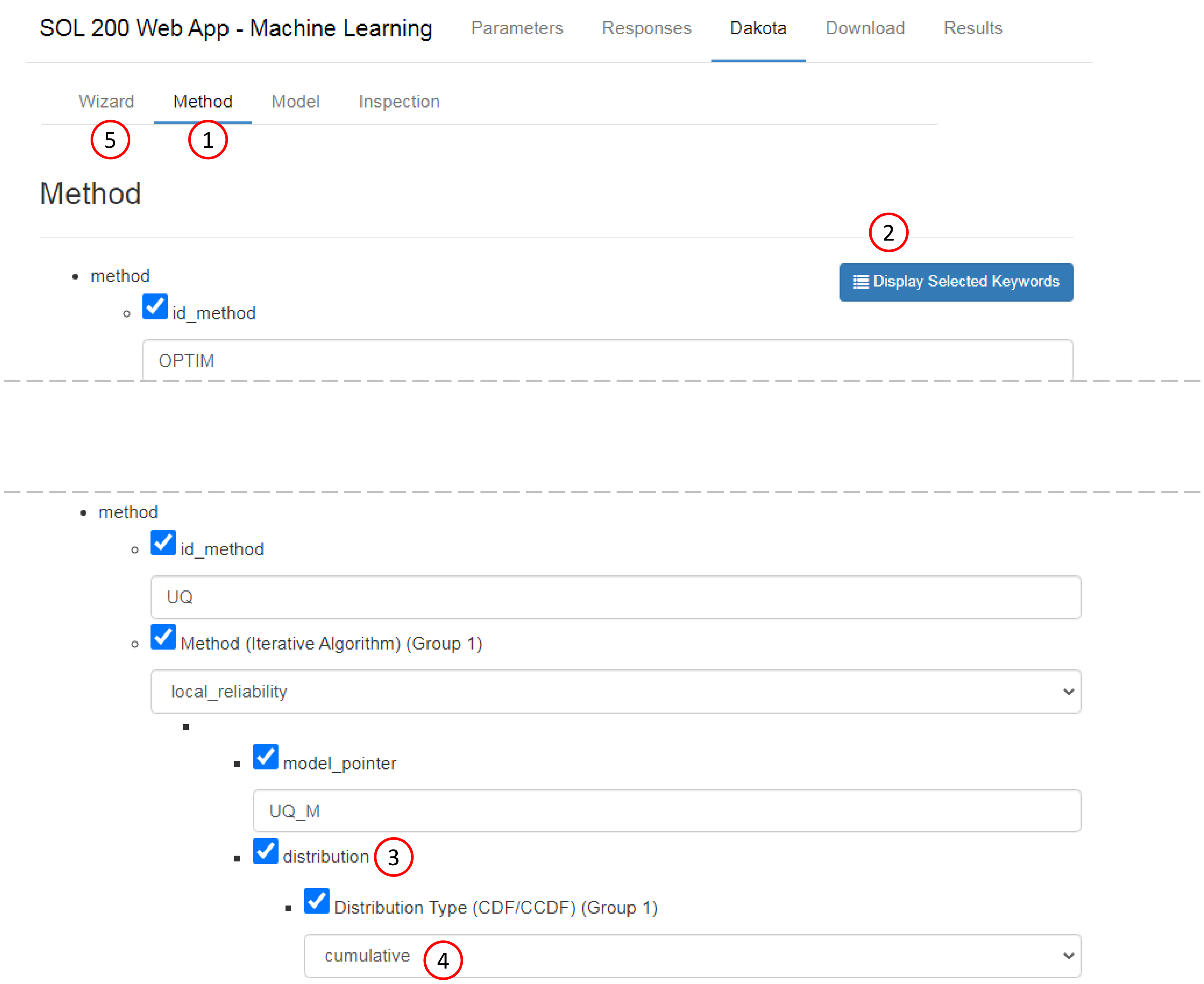
$$P(-a < X \leq b)$$
$$= F_X(b) - F_X(a)$$

When the cumulative option is used, Dakota will output probabilities or reliability indices of the form below, which correspond to CDF values:

$$P(X \leq b)$$
$$= F_X(b)$$

When the complementary option is used, Dakota will output probabilities or reliability indices of the form below, which correspond to CCDF values:

$$P(b < X)$$
$$= \bar{F}_X(b)$$



Probabilities Output by Sandia Dakota

1. Navigate to section Configure OUU Objective and Additional Constraints

A. Note in the Preview, the distribution is set to cumulative values.

B. Since the cumulative option is used, Sandia Dakota will output CDF values for response levels -14250, 19000, -14250, 19000 and are labeled r2_p1, r2_p2, r3_p1 and p3_p2, respectively.

For response r2, the probabilities for the lower and upper bound are as follows.

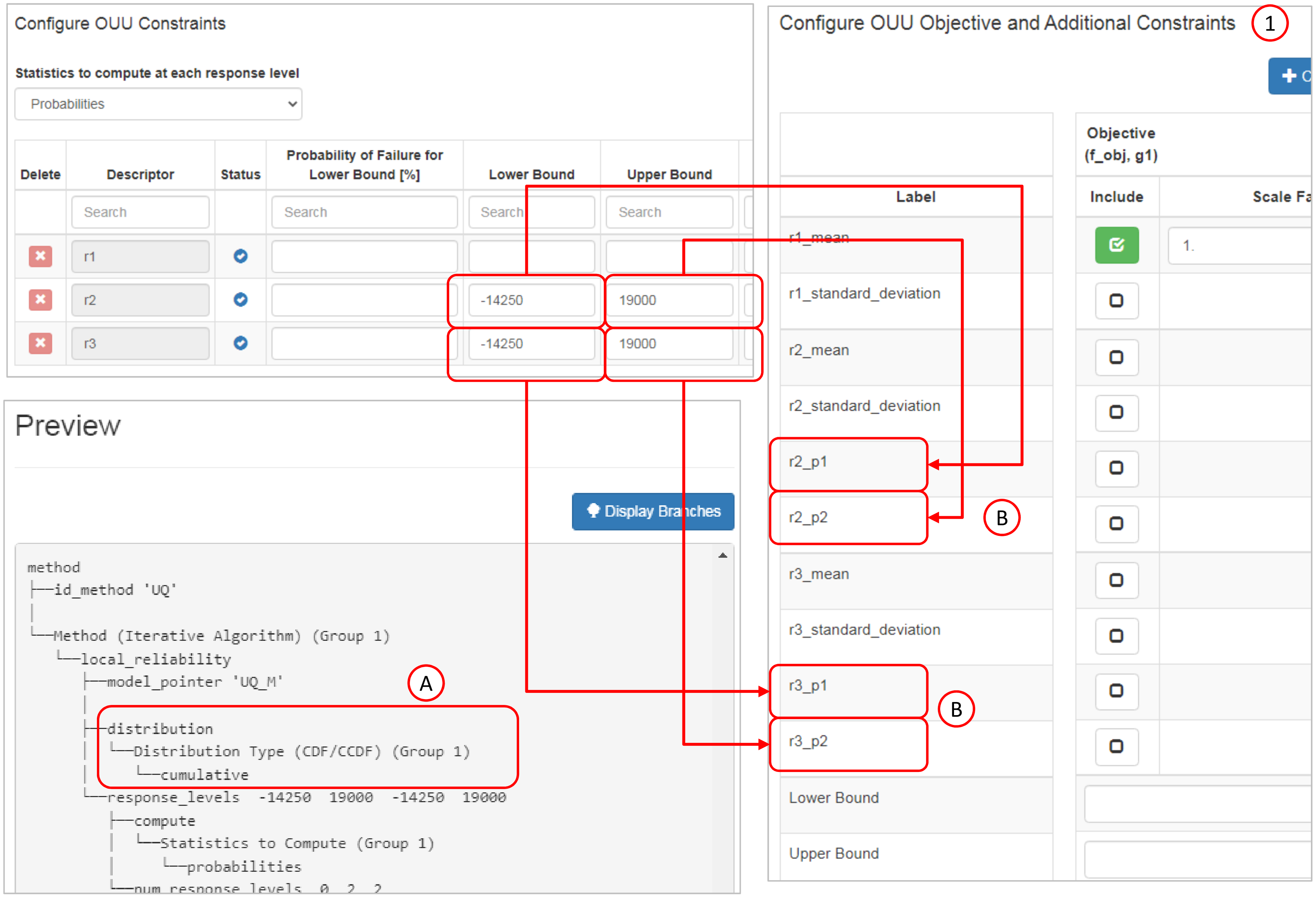
$$r2\_p1 = P(X \leq -14250) = F_X(-14250)$$

$$r2\_p2 = P(X \leq 19000) = F_X(19000)$$

For response r3, the probabilities for the lower and upper bound are as follows.

$$r3\_p1 = P(X \leq -14250) = F_X(-14250)$$

$$r3\_p2 = P(X \leq 19000) = F_X(19000)$$



Dakota - Optimization Under Uncertainty (OUU)

Constraints of the following form may be manually created:

gi = label_1 * scale_factor_1 + … +
label_2 * scale_factor_2 + … +
label_i * scale_factor_i

Constraints are created of this form,

P(-a < X ≤ b)

= F_X (b) - F_X (a)

= 1.0 * ri_p2 - 1.0 * ri_p1

where a and b are the lower and upper bound, respectively. This probability P(-a < X ≤ b) is limited to 95% probability of survival or (5% probability of failure)

Dakota - Optimization Under Uncertainty (OUU)

Create a constraint defines this expression:

$P(-14250 < X \leq 19000)$

$= F_X(19000) - F_X(-14250)$
$= 1.0 * r2\_p2 - 1.0 * r2\_p1$

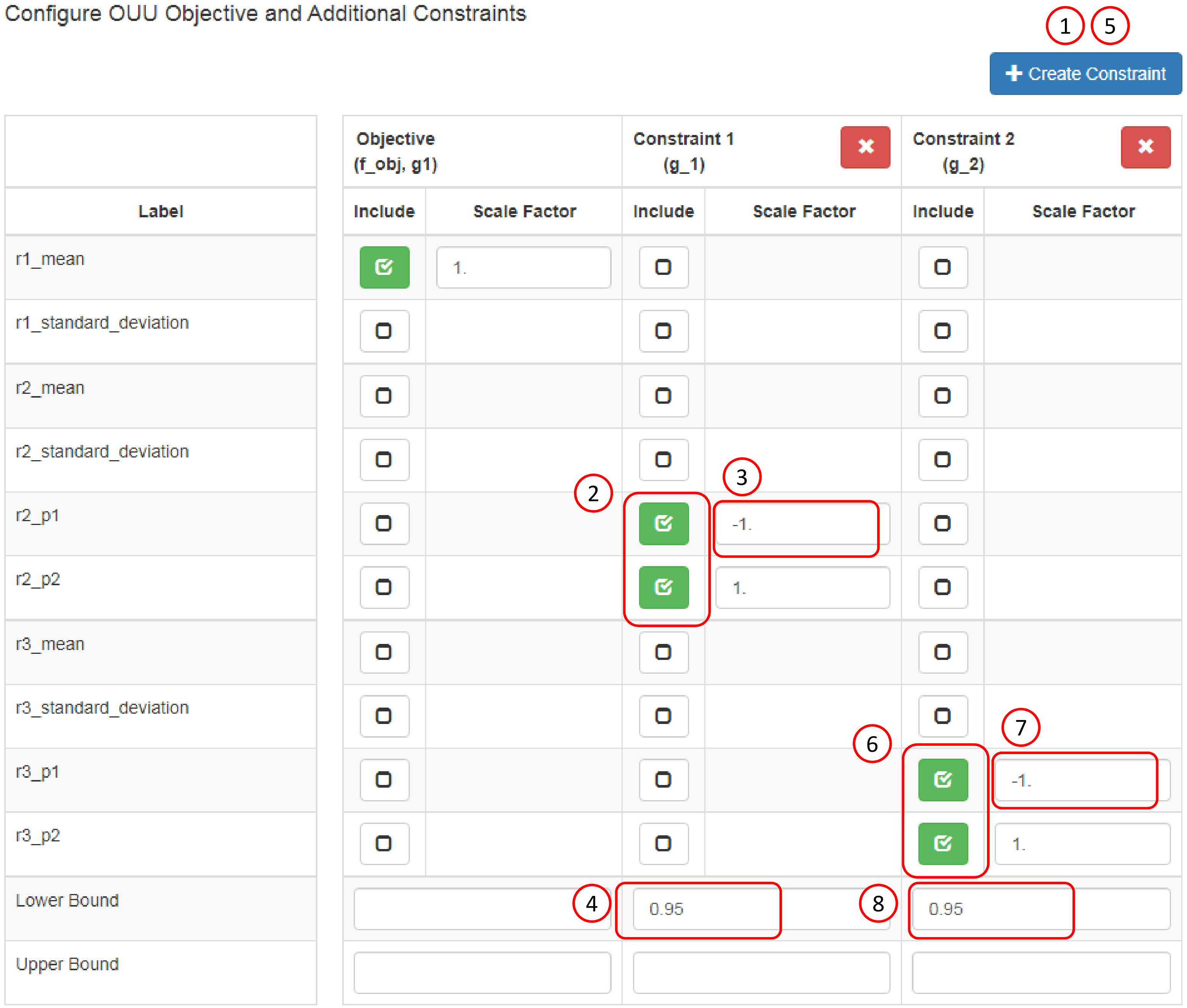
1. Click the indicated button to create a custom constraint.
2. Click the indicated buttons
3. Set the scale factor to -1.
4. Set the lower bound to 0.95

Create a constraint defines this expression:

$P(-14250 < X \leq 19000)$

$= F_X(19000) - F_X(-14250)$
$= 1.0 * r3\_p2 - 1.0 * r3\_p1$

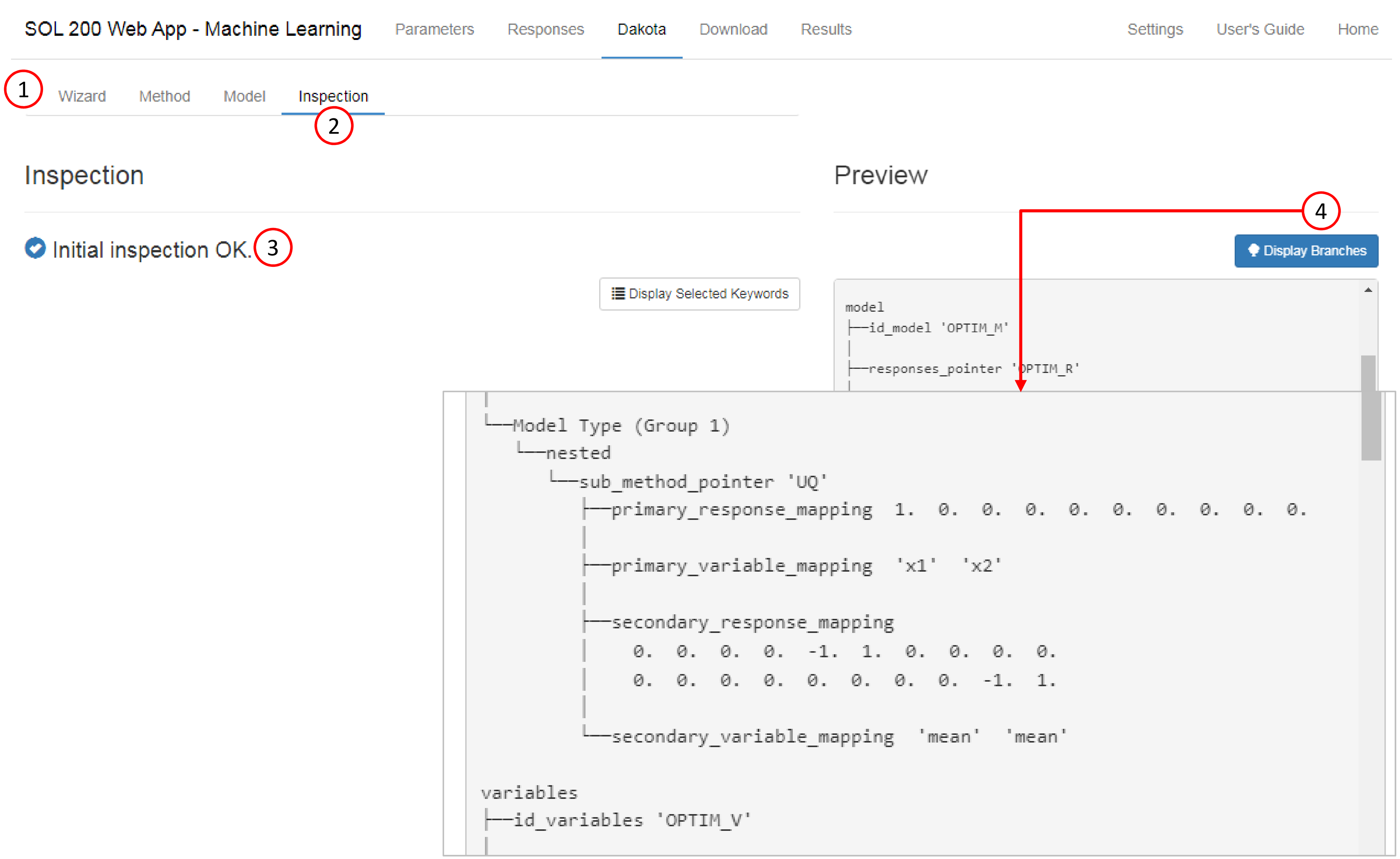
5. Click the indicated button to create a custom constraint.
6. Click the indicated buttons
7. Set the scale factor to -1.
8. Set the lower bound to 0.95



HEXAGON Technology Partner

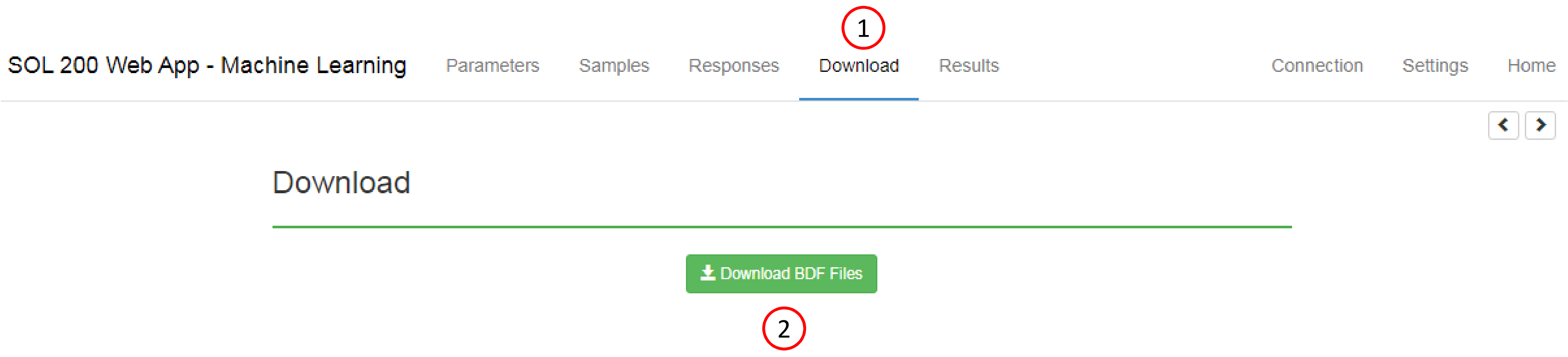
Inspection

1. Scroll to the navigation bar listing Wizard, Method, Model and Inspection
2. Click inspection
3. Ensure there are no error and the message reads OK
4. Click Display Branches
   - This will display branch lines that help communicate how each keyword in the Dakota input file is associated



Download

1. Click Download
2. Click Download BDF Files

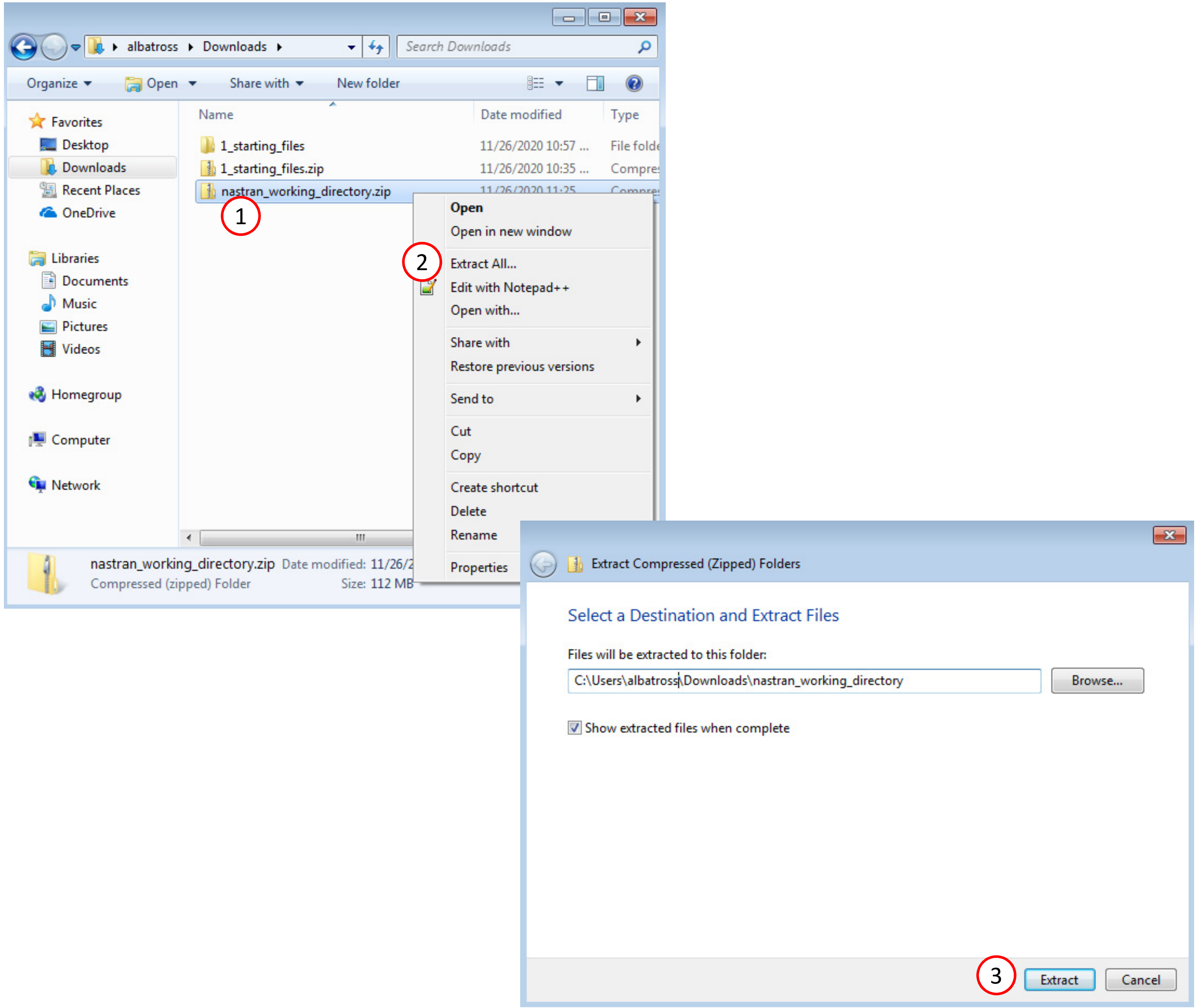


Start MSC Nastran

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

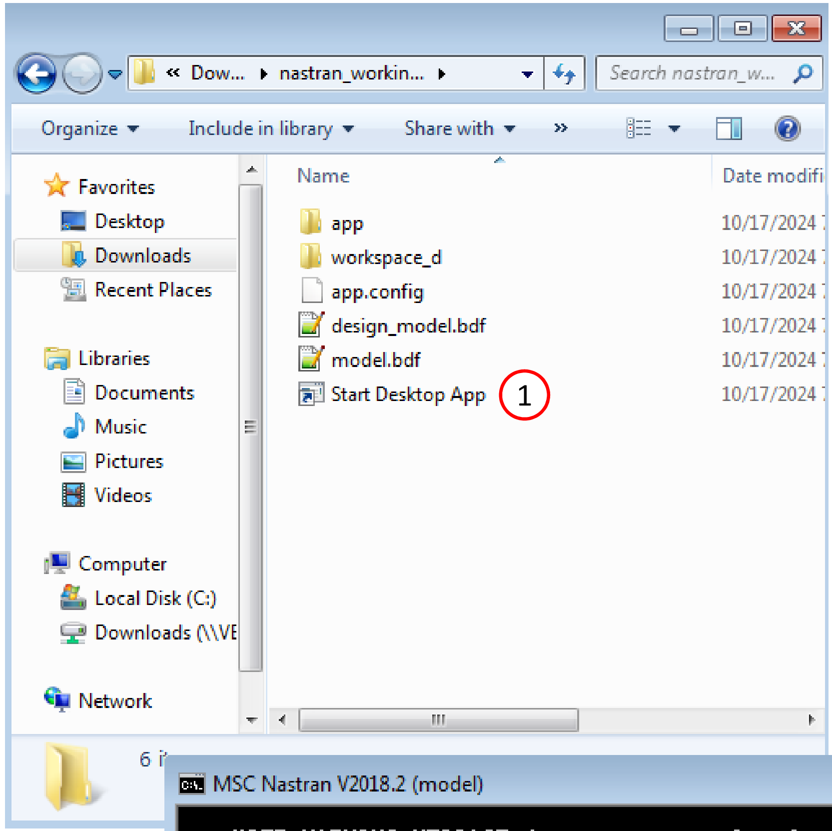
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click "Start Desktop App" to display the results.

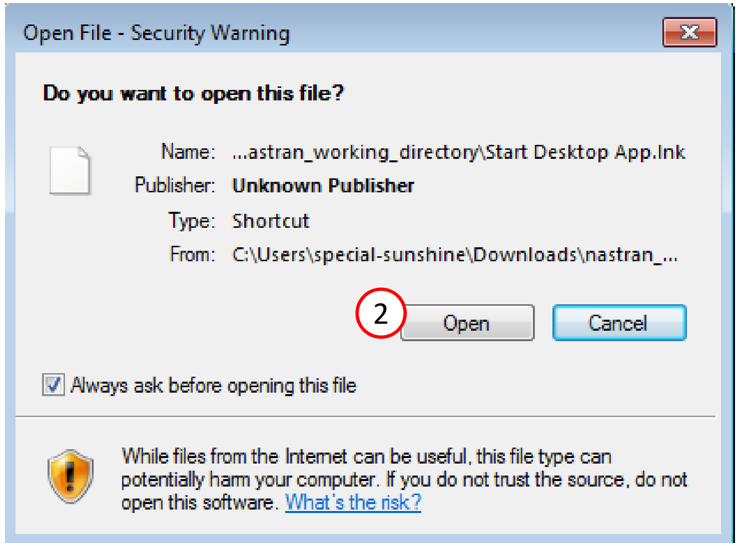
Using Linux?

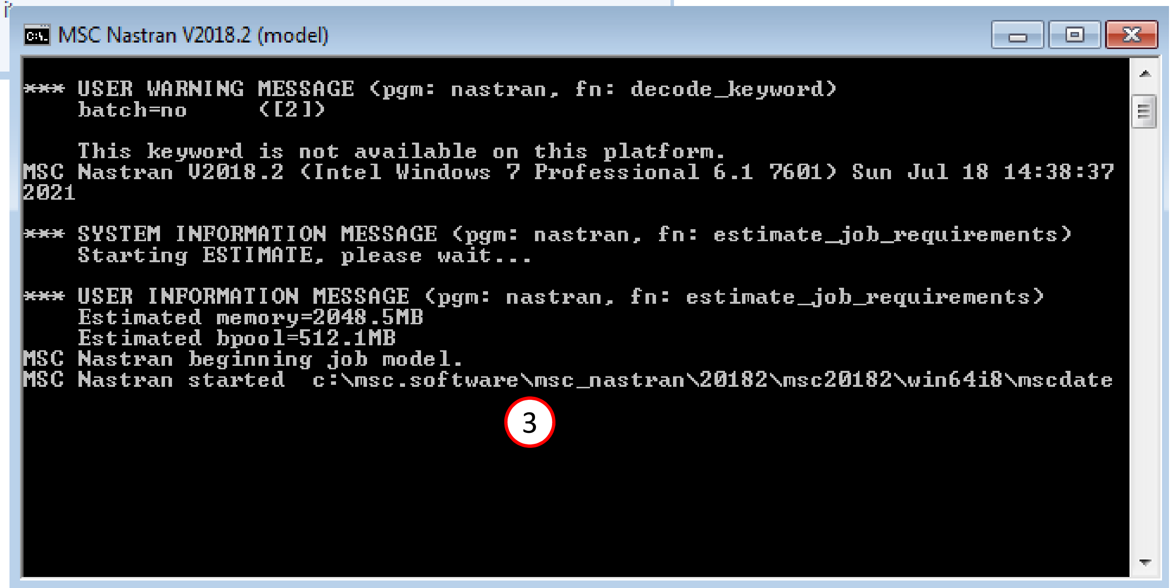
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

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

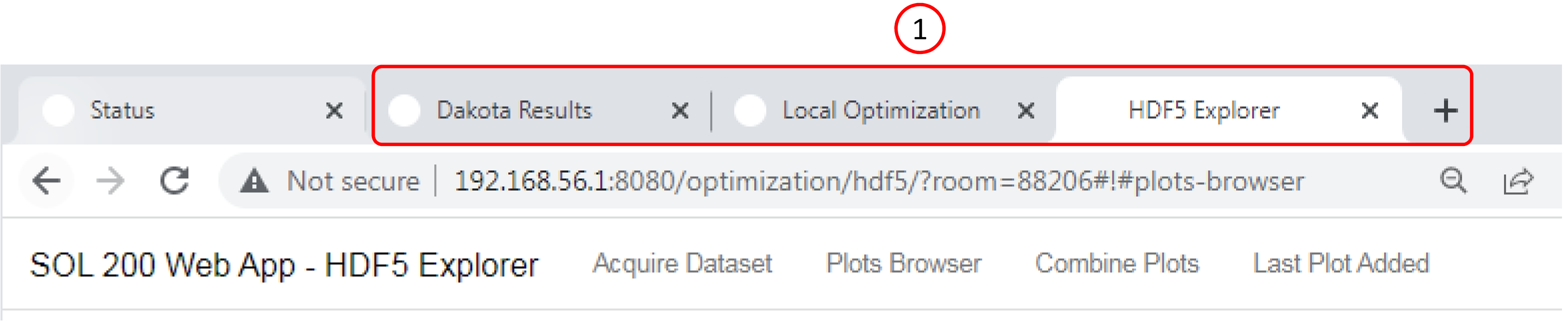
Python MSC Nastran

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.
3. At the end of the optimization, the normalized constraint is very small and close to zero. Negative or near zero normalized constraint values indicate a feasible design. This optimization has converged to a feasible design.

OUU Results

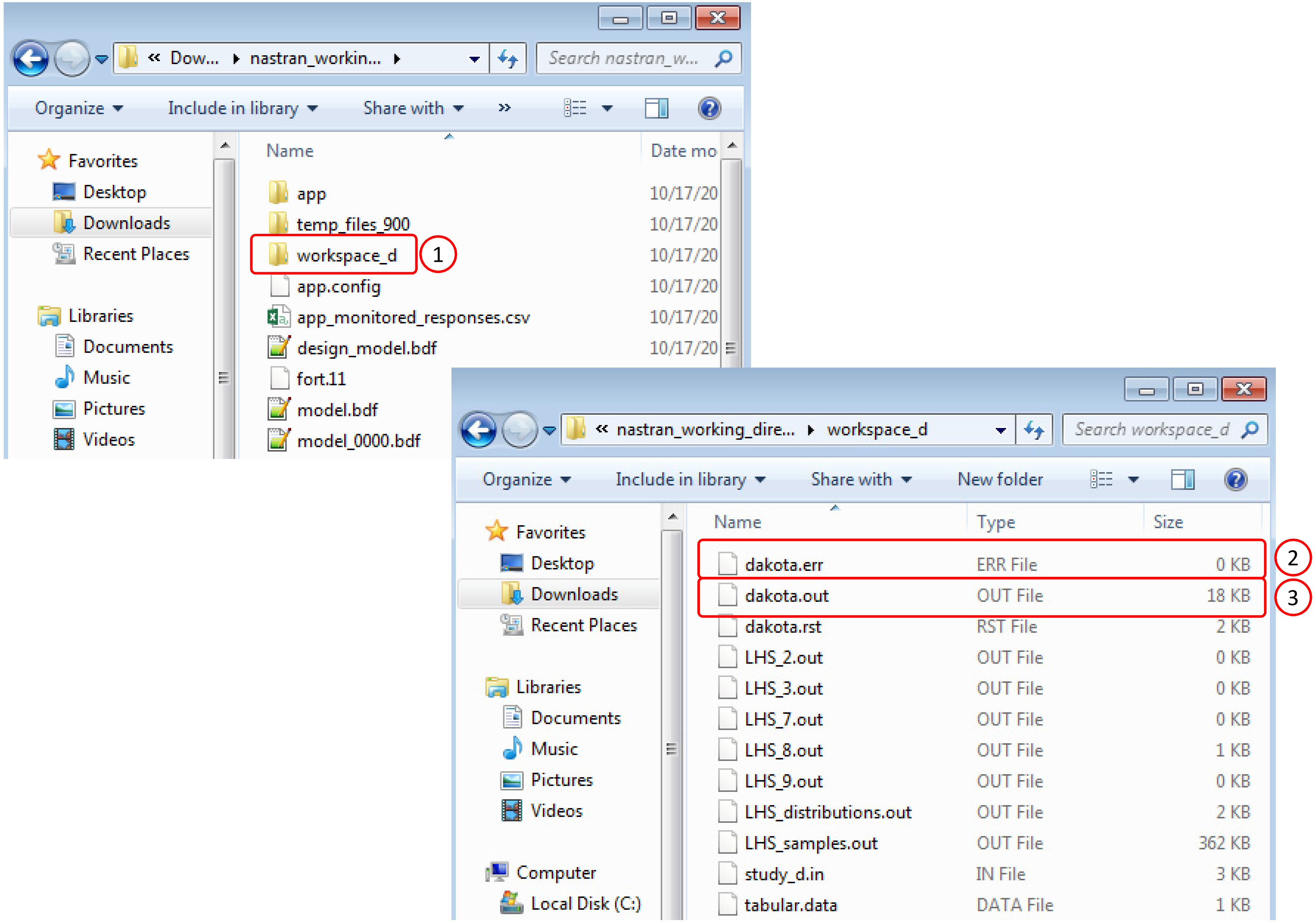
1. Navigate to section Design Variables
2. Since the initial design was infeasible, i.e. one or more constraints on probabilities of failure were violated, the optimizer has had to increase the mean of variable x1 (x1_mean).
3. The mean of variable x2 (x2_mean) has also varied, but the net change was small.

Questions? Email: christian@ the-engineering-lab.com

HEXAGON Technology Partner

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. Recall he objective was to minimize the mean of r1, i.e. minimize the mean weight.
4. This exercise was configured to constrain probabilities of survival P(a < X < b). The reported constraint values are probabilities of survival.
5. During the OUU, the optimizer has acquired response or gradients for 45 designs or variable values. 40 of these evaluations were unique, while 5 evaluations was non-unique. MSC Nastran was run 40 times at only the unique designs. Each evaluation was to acquire responses or gradients.
6. Lastly, recall the following initial configurations were made to reduce the cost of OUU.
   - Local reliability with MVFOSM was used.
   - A SOL 200 optimization was performed to determine ideal initial values for the OUU variables.

   The total wall clock time was ~176 seconds, which is approximately 2 minutes.

1 5

```
<<<<< Function evaluation summary (UQ_I): 45 total (40 new, 5 duplicate)
<<<<< Best parameters          =
                      9.5655447473e-01 x1_mean
                      3.2127255631e-01 x2_mean
<<<<< Best objective function  =
                      3.0268171789e+00
<<<<< Best constraint values   =
                      9.4922214748e-01
                      9.4922214748e-01
<<<<< 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        =    176.952 [parent =    176.952, child =          0]
  Total wall clock =    176.381
```

2 3 4 6

Discussion of Final Probabilities of Failure

1. The constraint values for P(-14250 < r2 ≤ 19000) and P(-14250 < r3 ≤ 19000) are listed. The probability of survival 94.922% is slightly violating the lower bound of 95%, but a small level of violation is expected when using optimizers. The constraint is deemed satisfied. The probability of failure is 5.077% (1.0 – 9.49222E-1)

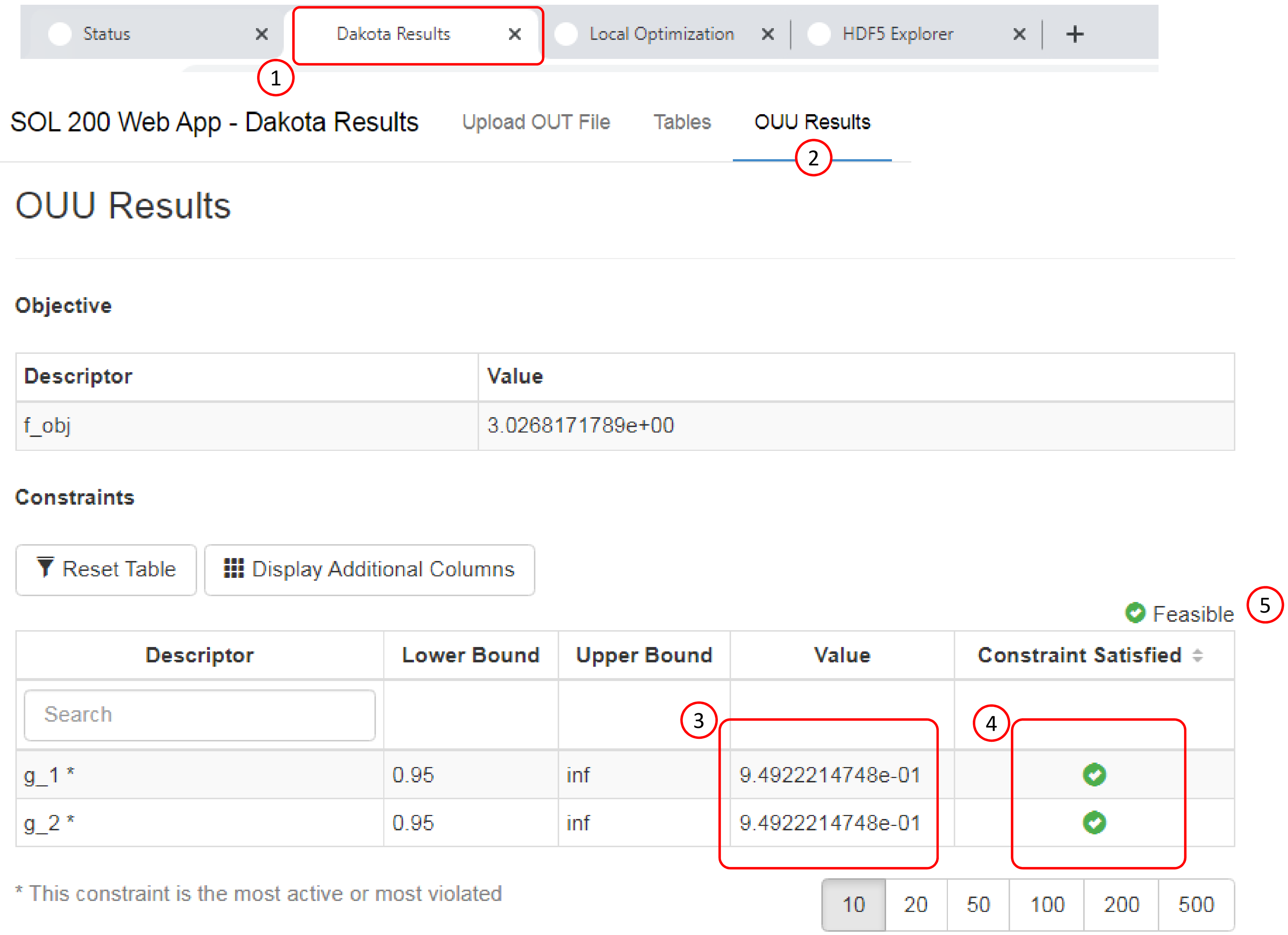
```
<<<<< Function evaluation summary (UQ_I): 45 total (40 new, 5 duplicate)
<<<<< Best parameters          =
                      9.5655447473e-01 x1_mean
                      3.2127255631e-01 x2_mean
<<<<< Best objective function  =
                      3.0268171789e+00
<<<<< Best constraint values   =
                      9.4922214748e-01
                      9.4922214748e-01
<<<<< 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        =    176.952 [parent =    176.952, child =          0]
  Total wall clock =    176.381
```

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. The values of the constraints are visible
4. Each constraint has an associated icon indicating if the constraint is satisfied or violated. Upon inspection, all the individual constraints are satisfied.
5. Alternatively, the indicated icon represents if all the design constraints are satisfied or violated. In this case, the design feasible, indicating all the constraints are satisfied.



Part B – Optimization Under Certainty, Constraining Reliability Indices

Reducing the Cost of OUU

In part A, multiple strategies were implemented to reduce the cost of OUU and included:

1. the use of the MVFOSM method,
2. performing a SOL 200 optimization to determine ideal initial values for the OUU variables,
3. reducing the number of constraints.

In part A, probabilities were directly constrained and resulted in an OUU that required 40 MSC Nastran runs to converge.

There is another strategy to further reduce the number of FEA runs needed for OUU. The additional strategy involves constraining equivalent reliability indices, instead of directly constraining probabilities.

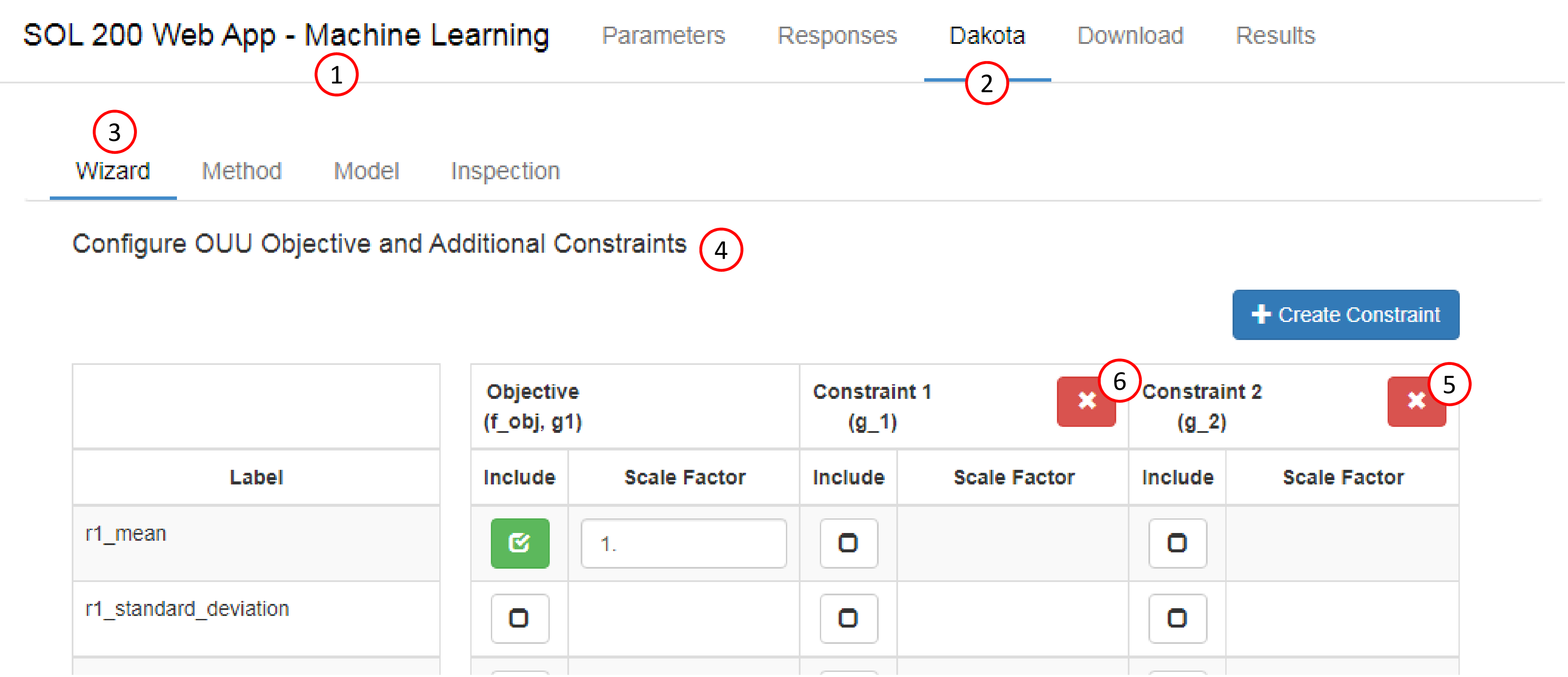
To the right is a table comparing the cost of OUUs that constrain probabilities and reliability indices. When constraining reliability indices, the OUU requires 17 FEA runs, and is a significant reduction of runs. It should be noted that constraining reliability indices yields faster OUU when local reliability methods, e.g. MVFOSM, are used for UQ. For other UQ methods, e.g. polynomial chaos, etc., constraining probabilities or reliability indices results in a similar number of runs. Constraining reliability indices should be considered when using the MVFOSM method, or other local reliability methods, during OUU.

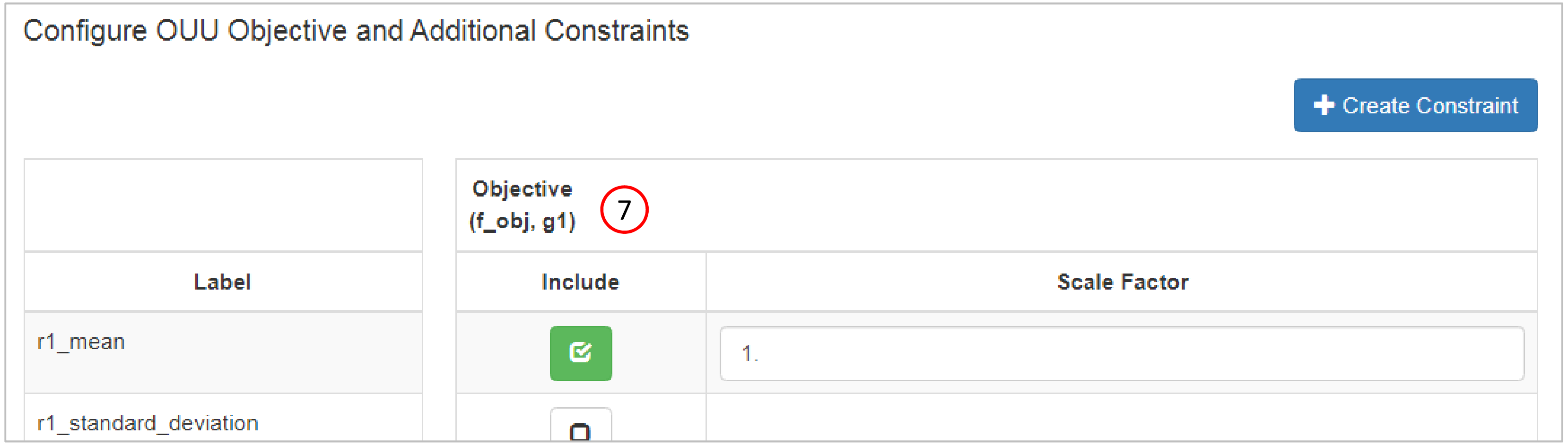
Part B discusses the process of constraining reliability indices and results in a less expensive OUU.

	Part A	Part B
Comments	Probabilities are constrained	Equivalent reliability indices are constrained
Objective	3.0268171789e+00	3.0274056041e+00
Number of MSC Nastran Runs	40 runs	17 Runs

OUU Configuration

1. Return to the Machine Learning web app
2. Click Dakota
3. Click Wizard
4. Scroll to section Configure OUU Objective and Additional Constraints
5. Click the indicated button to delete constraint g_2
6. Click the indicated button to delete constraint g_1
7. Only the Objective column should remain





Dakota - Optimization Under Uncertainty (OUU)

1. Navigate to section Configure OUU Constraints
2. Set Statistics to compute at each response level to Reliabilities
3. Set the Probability of Failure for Lower Bound to 5. This constrains the probability of the form P($X \leq$ a).
4. Set the Probability of Failure for Upper Bound to 5. This constrains the probability of the form P(b < X).

- Dakota has options to calculate the probabilities, reliabilities or generalized reliabilities for each response level. Per the reference below, it has been documented that the optimization converges faster when constraining reliabilities. While on the user interface, limits on probabilities of failure have been specified, internally equivalent limits on reliabilities have been defined.

Eldred, M. S., Agarwal, H., Perez, V. M., Wojtkiewicz, S. F., & Renaud, J. E. (2007). Investigation of reliability method formulations in DAKOTA/UQ. *Structure and Infrastructure Engineering*, 3(3), 199–213. https://doi.org/10.1080/15732470500254618

SOL 200 Web App - Machine Learning | Parameters | Responses | Dakota | Download | Results

Wizard | Method | Model | Inspection

Configure OUU Constraints (1)

Statistics to compute at each response level

Reliabilities (2)

Reset Table

Delete	Descriptor	Status	Probability of Failure for Lower Bound [%]	Lower Bound	Upper Bound	Probability of Failure for Upper Bound [%]
	Search		Search	Search	Search	Search
✖	r1	✔				
✖	r2	✔	5.	-14250	19000	5.
✖	r3	✔	5.	-14250	19000	5.

(3) (4)

Dakota - Optimization Under Uncertainty (OUU)

1. Scroll to section Configure OUU Objective and Additional Constraints
2. Notice the necessary constraints have been automatically created and managed.

Since reliability indices are being output, r2_p1 correspond to a reliability index for -14250 and r2_p2 corresponds to a reliability index for 19000. The same may be said for r3_p1 and r3_p2.

Statistics to compute at each response level

Reliabilities

Configure OUU Objective and Additional Constraints (1)

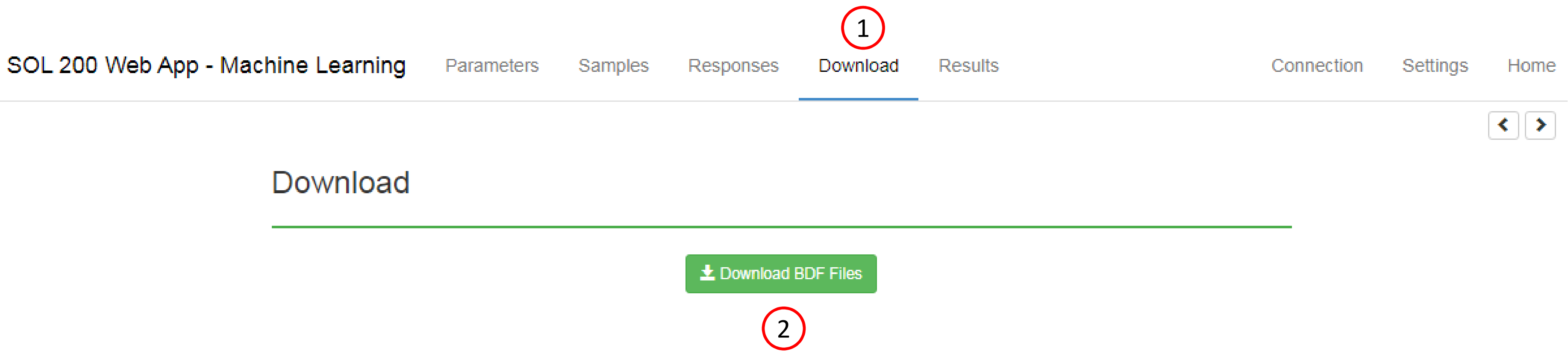
(2)

Create Constraint

	Objective (f_obj, g1)		Constraint 1 (r2_pl)		Constraint 2 (r2_pu)		Constraint 3 (r3_pl)	
Label	Include	Scale Factor	Include	Scale Factor	Include	Scale Factor	Include	Scale Factor
r1_mean	☑	1.	☐		☐		☐	
r1_standard_deviation	☐		☐		☐		☐	
r2_mean	☐		☐		☐		☐	
r2_standard_deviation	☐		☐		☐		☐	
r2_p1	☐		☑	1.	☐		☐	
r2_p2	☐		☐		☑	1.	☐	
r3_mean	☐		☐		☐		☐	
r3_standard_deviation	☐		☐		☐		☐	
r3_p1	☐		☐		☐		☑	1.
r3_p2	☐		☐		☐		☐	
Lower Bound					1.644854			
Upper Bound			-1.644854				-1.644854	

Download

1. Click Download
2. Click Download BDF Files

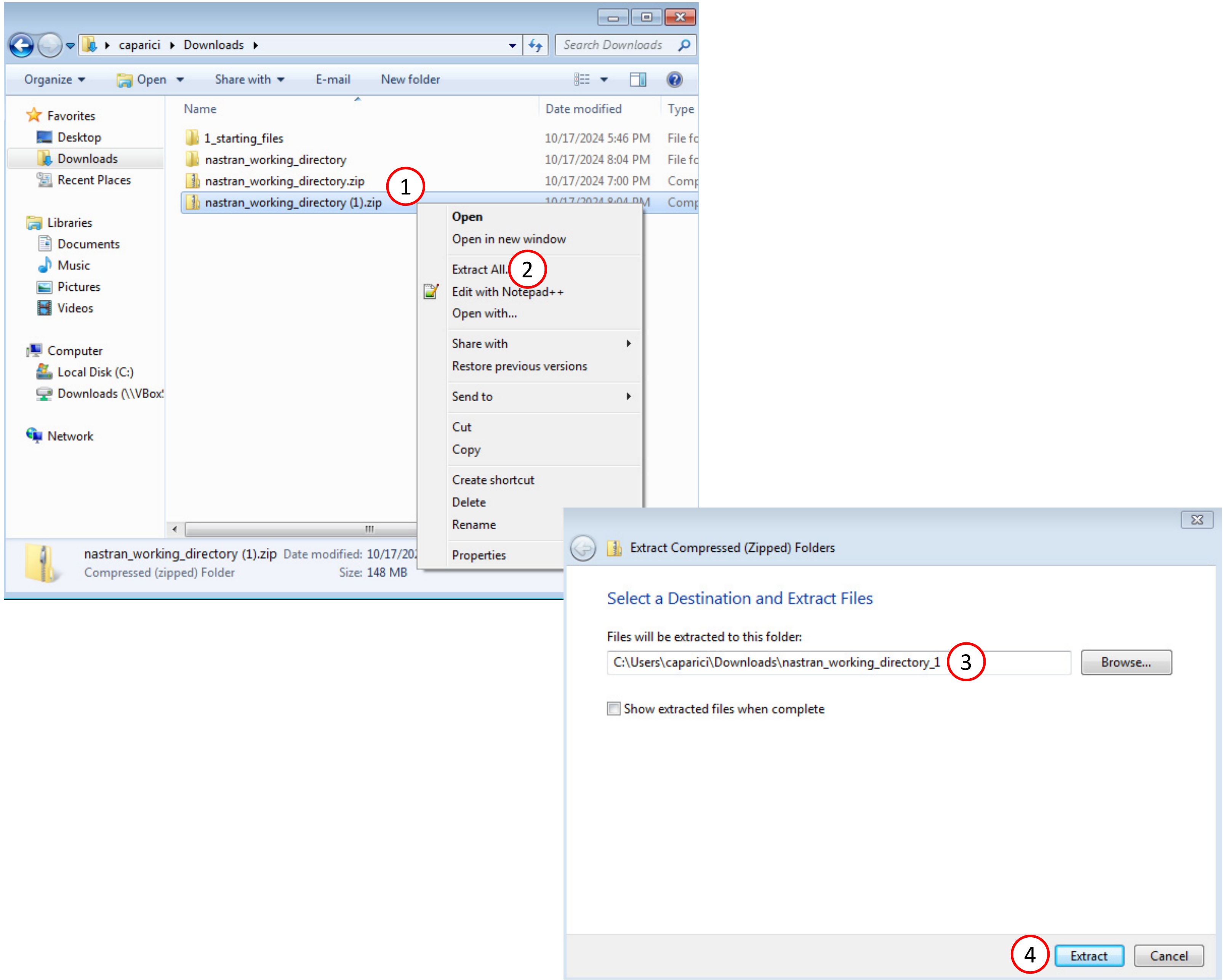


Start MSC Nastran

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

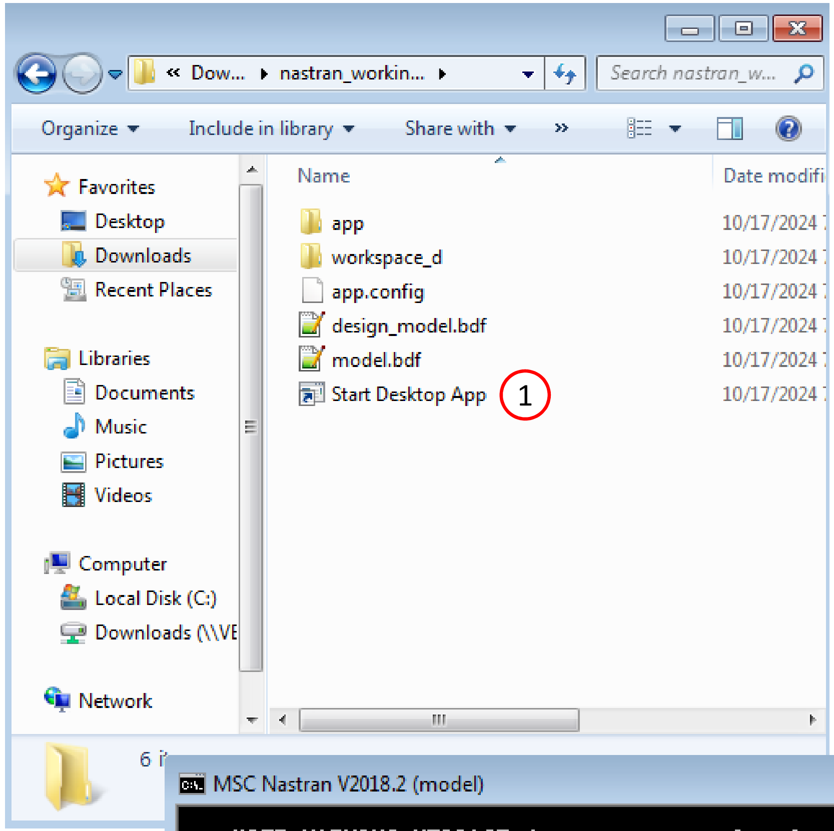
- One can run the Nastran job on a remote machine as follows: 1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click "Start Desktop App" to display the results.

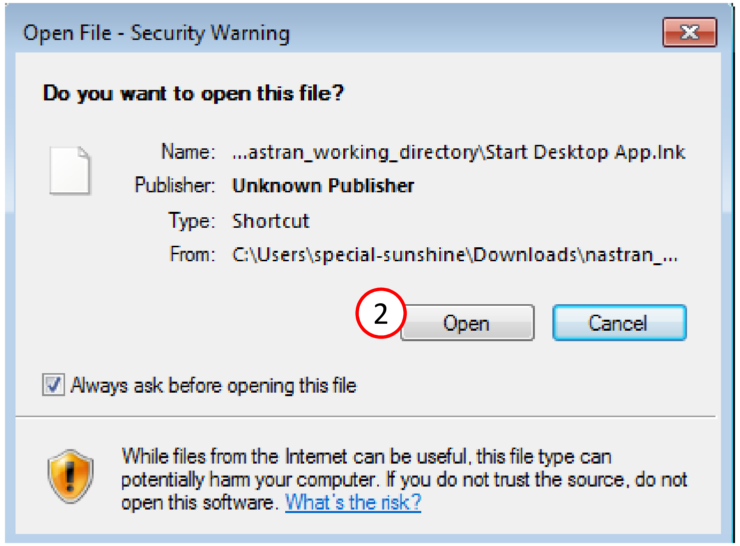
Using Linux?

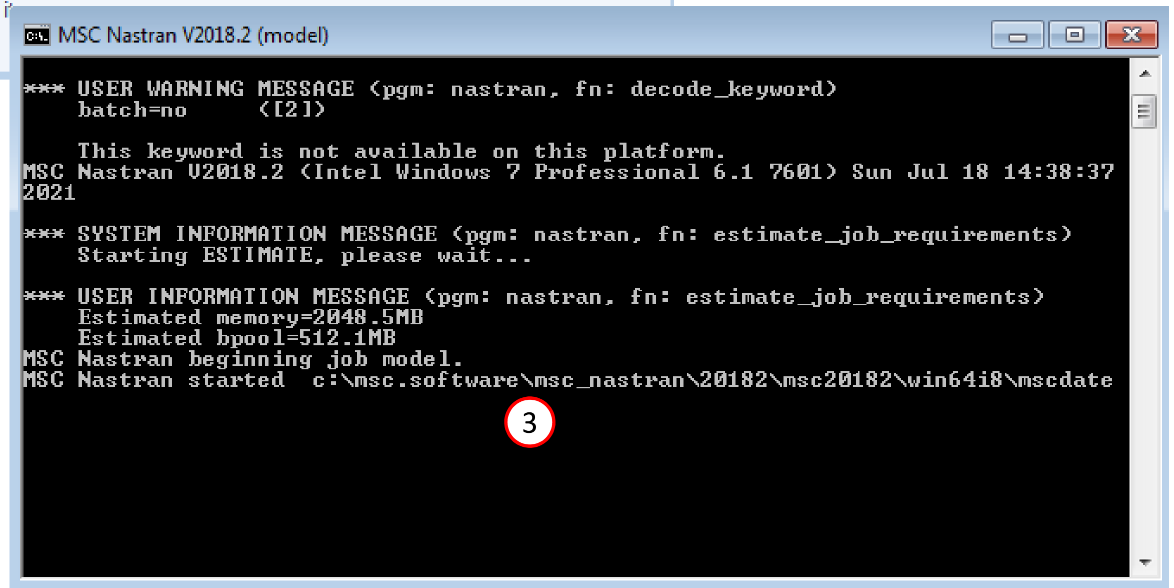
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

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

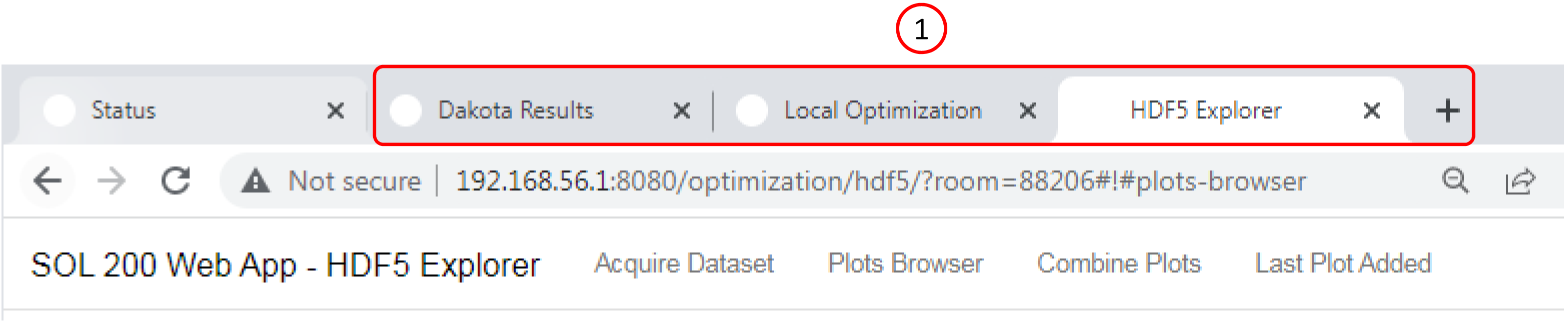
Python MSC Nastran

Status

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

OUU Completion

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

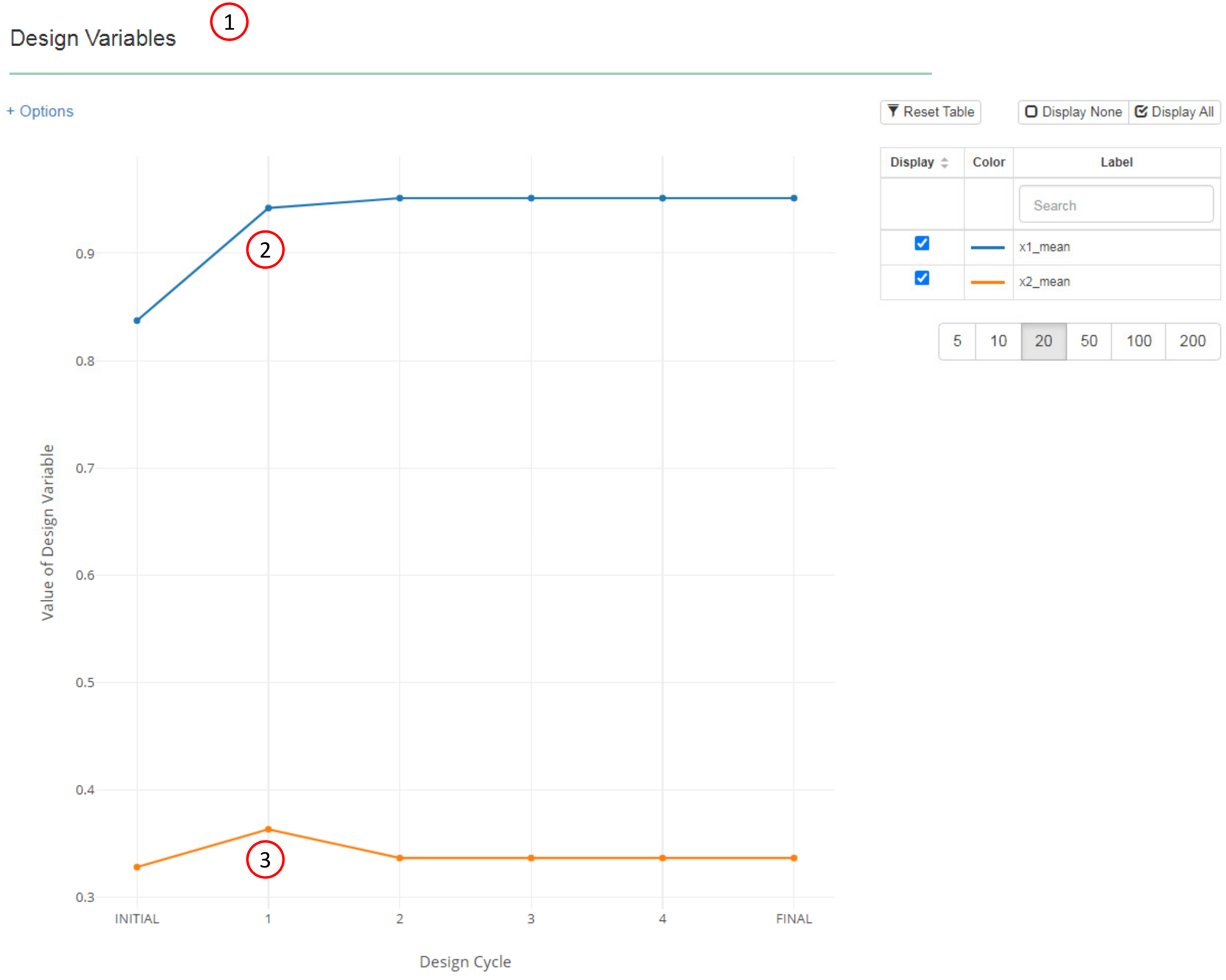


OUU Results

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.
3. At the end of the optimization, the normalized constraint is very small and close to zero. Negative or near zero constraint values indicate a feasible design. This optimization has converged to a feasible design.

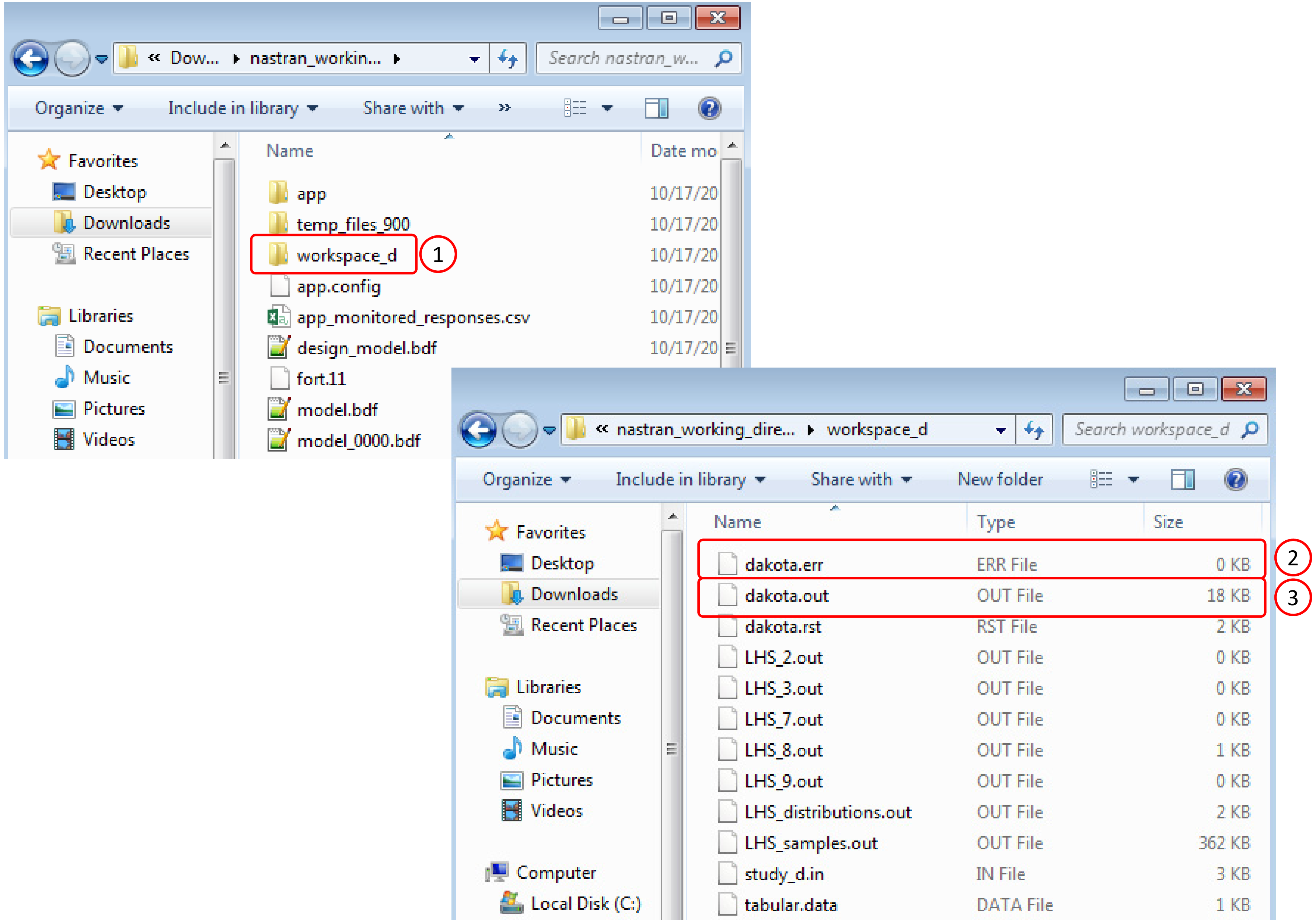
OUU Results

1. Navigate to section Design Variables
2. Since the initial design was infeasible, i.e. one or more constraints on probabilities of failure were violated, the optimizer has had to increase the mean of variable x1 (x1_mean).
3. The mean of variable x2 (x2_mean) has also varied, but the net change was small.



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. Recall he objective was to minimize the mean of r1, i.e. minimize the mean weight.
4. This exercise was configured to constrain probabilities of failure, but since we configured the Dakota input file to internally constrain equivalent reliability values, the reported constraint values are reliability values.
5. During the OUU, the optimizer has acquired response or gradients for 22 designs or variable values. 17 of these evaluations were unique, while 5 evaluations was non-unique. MSC Nastran was run 17 times at only the unique designs. Each evaluation was to acquire responses or gradients.
6. Lastly, the following initial configurations were made to reduce the cost of OUU. The total wall clock time was ~65 seconds, which is a little under 1 minute.
   - Local reliability with MVFOSM was used.
   - A SOL 200 optimization was performed to determine ideal initial values for the OUU variables.
   - Internally, reliabilities were constrained instead of probabilities.

1 5

```
<<<<< Function evaluation summary (UQ_I): 22 total (17 new, 5 duplicate)
<<<<< Best parameters          =
                      9.5130797754e-01 x1_mean
                      3.3670031639e-01 x2_mean
<<<<< Best objective function  =
                      3.0274056041e+00
<<<<< Best constraint values   =
                     -4.5383296250e+01
                      1.6448457983e+00
                     -4.5383296250e+01
                      1.6448457983e+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        =      63.777 [parent =      63.777, child =            0]
  Total wall clock =      65.896
```

2 3 4 6

Discussion of Final Probabilities of Failure

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 reliability, often known as the reliability index, is defined as:

$$\beta = \frac{\mu_{ri} - \mu_{ri\_upper\_bound}}{\sigma_{ri}}$$

The probability and reliability index are related via the following expression:

$$p_f = \Phi(-\beta)$$

1. The use of the distribution=complementary keyword in the Dakota input file study_d.in prompts Dakota to determine the probabilities, reliabilities or generalized reliabilities for when $P(x < X)$, where x is a response level corresponding to the lower or upper bound. If distribution=cumulative, or the distribution is absent and undefined, the values are calculated for when $P(X \le x)$.

2. The reliability index of 1.6448 yields a probability of failure $p_f = \Phi(-1.6448) = 0.050001$ (5.0001%) for the upper bound, i.e. P(19000 < r2) = 5.0001%. Note that 5.0001 is not below 5.0 and shows some violation of the constraint, which is expected due to optimizer tolerances. Even though the violation is small, the violation is within the tolerance to be deemed as a satisfied constraint. The constraints are satisfied for the upper bounds.

3. For the lower bound, the reliabilities correspond to the probability of being greater than the lower bound, i.e. probability of survival (p_s), , i.e. P(-14250 < r2) . For the lower bound, the probability of failure is equal to 1.0 - p_s. The reliability index of -45.38 yields $p_s = \Phi(- -45.38) = 1$ (100% p_s) for the lower bound. The p_f is 1.0 - p_s=0.0 (0% p_f). Since 0% < 5% p_f, the constraints are satisfied for the lower bounds.

```
method
  id_method 'UQ'
    local_reliability
      model_pointer 'UQ_M'
      distribution
          complementary        (1)
      response_levels -20000 20000 -20000 20000
        compute
            reliabilities
        num_response_levels 0 2 2
```

```
<<<<< Function evaluation summary (UQ_I): 22 total (17 new, 5 duplicate)
<<<<< Best parameters          =
                      9.5130797754e-01 x1_mean
                      3.3670031639e-01 x2_mean
<<<<< Best objective function  =
                      3.0274056041e+00
<<<<< Best constraint values   =
                     -4.5383296250e+01
                      1.6448457983e+00
                     -4.5383296250e+01
                      1.6448457983e+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        =     63.777 [parent =     63.777, child =           0]
  Total wall clock =     65.896
```

(2) 1.6448457983e+00 values; (3) -4.5383296250e+01 values

Discussion of Final Probabilities of Failure

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

What are generalized reliabilities?

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

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

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

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

The limit state function may be thought of as the normalized constraints, i.e. g(x) < 0.0.

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 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)

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. The values of the constraints are visible
4. Each constraint has an associated icon indicating if the constraint is satisfied or violated. Upon inspection, all the individual constraints are satisfied.
5. Alternatively, the indicated icon represents if all the design constraints are satisfied or violated. In this case, the design feasible, indicating all the constraints are satisfied.

Status ✕ | Dakota Results ✕ | Local Optimization ✕ | HDF5 Explorer ✕ | +

1

SOL 200 Web App - Dakota Results   Upload OUT File   Tables   OUU Results

2

OUU Results

Objective

Descriptor	Value
f_obj	3.0274056041e+00

Constraints

Reset Table   Display Additional Columns

5

Feasible

3   4

Descriptor	Lower Bound	Upper Bound	Value	Constraint Satisfied
Search				
r2_pl	-inf	-1.644854	-4.5383296250e+01	✔
r2_pu *	1.644854	inf	1.6448457983e+00	✔
r3_pl	-inf	-1.644854	-4.5383296250e+01	✔
r3_pu *	1.644854	inf	1.6448457983e+00	✔

* This constraint is the most active or most violated

10 20 50 100 200 500

Discussion of Final Probabilities of Failure

1. Click Display Additional Columns
2. The constraint values include probabilities of survival (p_s) and failure (p_f). The goal is to interpret all the values as probabilities of failure.
3. Click Display All as Probability of Failure
4. All p_s values has been updated to equivalent p_f values. This is done by taking the complement:

$$p_f = 1.0 - p_s \,.$$

Constraints

1

Reset Table | Display Additional Columns | Display All as Probability of Failure

Feasible

2

Descriptor	Lower Bound	Upper Bound	Value	pf/ps	Description	Probability	Reliability Index	Constraint Satisfied
Search								
r2_pl	-inf	-1.644854	-4.5383296250e+01	p_s	P(-1.4250e+4 < r2)	1.0000e+0	-4.5383296250e+01	✔
r2_pu *	1.644854	inf	1.6448457983e+00	p_f	P(1.9000e+4 < r2)	5.0001e-2	1.6448457983e+00	✔
r3_pl	-inf	-1.644854	-4.5383296250e+01	p_s	P(-1.4250e+4 < r3)	1.0000e+0	-4.5383296250e+01	✔
r3_pu *	1.644854	inf	1.6448457983e+00	p_f	P(1.9000e+4 < r3)	5.0001e-2	1.6448457983e+00	✔

Constraints

3

Reset Table | Display Additional Columns | Display All as Probability of Failure

Feasible

4

Descriptor	Lower Bound	Upper Bound	Value	pf/ps	Description	Probability	Reliability Index	Constraint Satisfied
Search								
r2_pl	-inf	-1.644854	-4.5383296250e+01	p_f	P(r2 < -1.4250e+4)	0.0000	-inf	✔
r2_pu *	1.644854	inf	1.6448457983e+00	p_f	P(1.9000e+4 < r2)	5.0001e-2	1.6448457983e+00	✔
r3_pl	-inf	-1.644854	-4.5383296250e+01	p_f	P(r3 < -1.4250e+4)	0.0000	-inf	✔
r3_pu *	1.644854	inf	1.6448457983e+00	p_f	P(1.9000e+4 < r3)	5.0001e-2	1.6448457983e+00	✔

Discussion of Final Probabilities of Failure

There is a goal to identify the most active or most violated constraints. Such constraints are marked with an asterisk (*).

1. The indicated constraints are the most active or most violated since an asterisk (*) is visible
2. In this exercise, reliability indices were constrained. Equivalent probability values are displayed in the column Probability.
3. The probabilities of failure are 5.0001%, which exceeds the limit of 5% specified in this exercise. Due to optimizer tolerances, a small amount of constraint violation is expected.
4. Since the violation is minor, the design is deemed feasible.

Status ✕ | Dakota Results ✕ | Local Optimization ✕ | HDF5 Explorer ✕ | +

SOL 200 Web App - Dakota Results | Upload OUT File | Tables | OUU Results

OUU Results

Objective

Descriptor	Value
f_obj	3.0274056041e+00

Constraints

Reset Table | Display Additional Columns | Display All as Probability of Failure

4 Feasible

Descriptor	Lower Bound	Upper Bound	Value	pf/ps	Description	Probability	Reliability Index	Constraint Satisfied
Search								
r2_pl	-inf	-1.644854	-4.5383296250e+01	p_f	P(r2 < -1.4250e+4)	0.0000	-inf	✔
r2_pu *	1.644854	inf	1.6448457983e+00	p_f	P(1.9000e+4 < r2)	5.0001e-2	1.6448457983e+00	✔
r3_pl	-inf	-1.644854	-4.5383296250e+01	p_f	P(r3 < -1.4250e+4)	0.0000	-inf	✔
r3_pu *	1.644854	inf	1.6448457983e+00	p_f	P(1.9000e+4 < r3)	5.0001e-2	1.6448457983e+00	✔

1 2 3

* This constraint is the most active or most violated

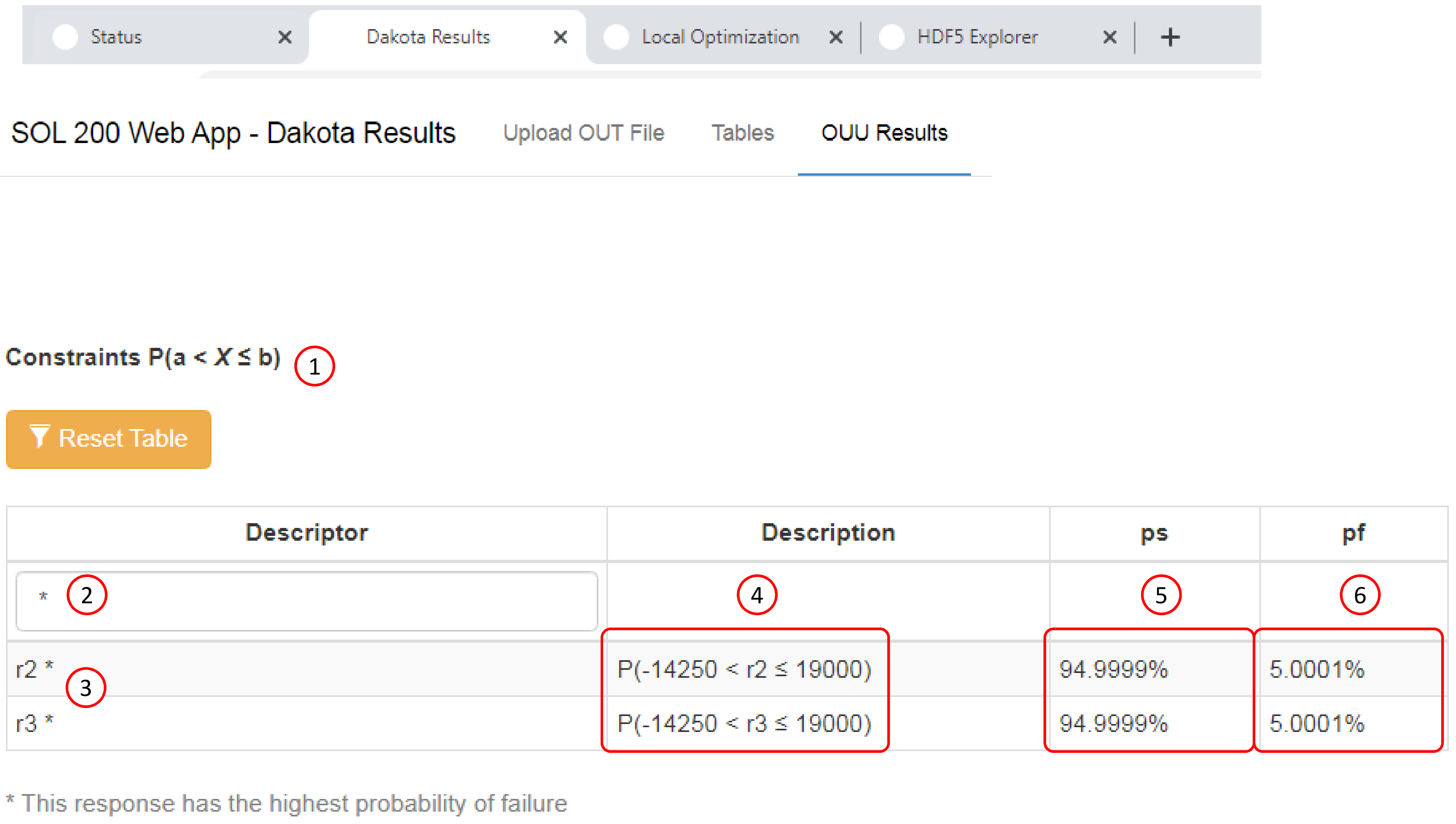
10 | 20 | 50 | 100 | 200 | 500

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 failure is displayed in column pf.

The highest probability of failure is 5.0001%.



Reducing the Cost of OUU

Constraining reliability indices has proven to be very effective in reducing the cost of OUU.

There are now 4 strategies to reducing the cost of OUU.

1. Use the MVFOSM method, if appropriate.
2. Perform a SOL 200 optimization to determine ideal initial values for the OUU variables.
3. Reduce the number of constraints.
4. Constrain reliability indices if using local reliability methods, e.g. MVFOSM.

	Part A	Part B
Comments	Probabilities are constrained	Equivalent reliability indices are constrained
Objective	3.0268171789e+00	3.0274056041e+00
Number of MSC Nastran Runs	40 runs	17 Runs

Part C – Verification of OUU Solution

Motivation

So far during the OUUs, the MVFOSM method has been used to approximate the statistical quantities such as mean, standard deviation and probabilities.

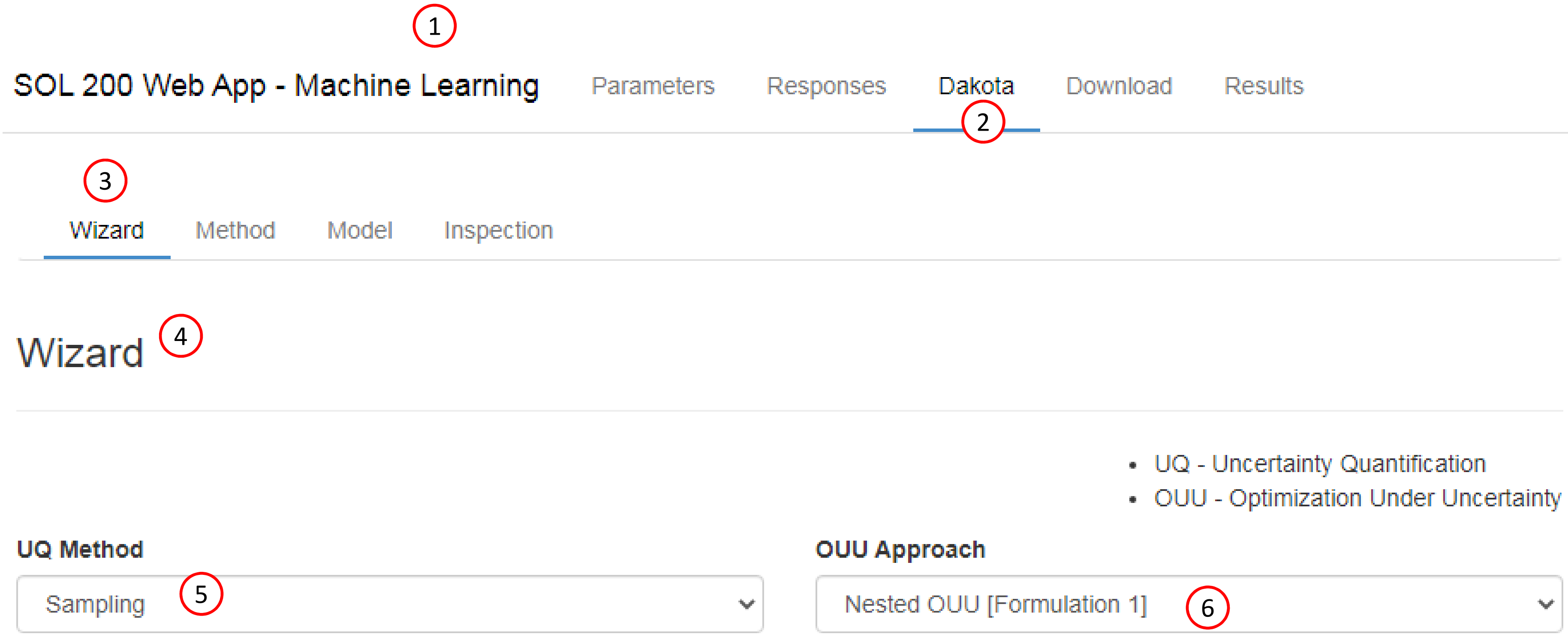
A confirmation must be performed to ensure the actual probabilities of failure are within the desired limits.

This part C discusses the verification process.

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. Ensure you are in section Configure OUU Variables
3. Take the optimal variable values from the previous OUU and replace the old initial values of the OUU variables.

The idea is to determine the new probabilities of failure at the optimal variable values. The UQ method is sampling and will consist of 80 MSC Nastran runs.

SOL 200 Web App - Machine Learning | Parameters | Responses | Dakota | Download | Results

Configure OUU Variables (1) (2)

Reset Table

Delete	Descriptor	Status	Initial Value	Lower Bound	Upper Bound	Description
✖	x1_mean	✔	9.5130797754E-01	.01	2.	Mean - Field 4 of DESVAR 100001
✖	x2_mean	✔	3.3670031639E-01	.01	2.	Mean - Field 4 of DESVAR 100002

(3)

```
<<<<< Function evaluation summary (UQ_I): 22 total (17 new, 5 duplicate)
<<<<< Best parameters          =
                      9.5130797754e-01 x1_mean
                      3.3670031639e-01 x2_mean
<<<<< Best objective function  =
                      3.0274056041e+00
<<<<< Best constraint values   =
                     -4.5383296250e+01
                      1.6448457983e+00
                     -4.5383296250e+01
                      1.6448457983e+00
```

Uncertainty Quantification

1. Navigate to section Configure OUU Constraints
2. Restore the lower bound to -15000
3. Restore the upper bound to 20000

The goal is to determine the actual probability P(-15000 < X < 20000)

Configure OUU Constraints (1)

Statistics to compute at each response level

Reliabilities

Reset Table

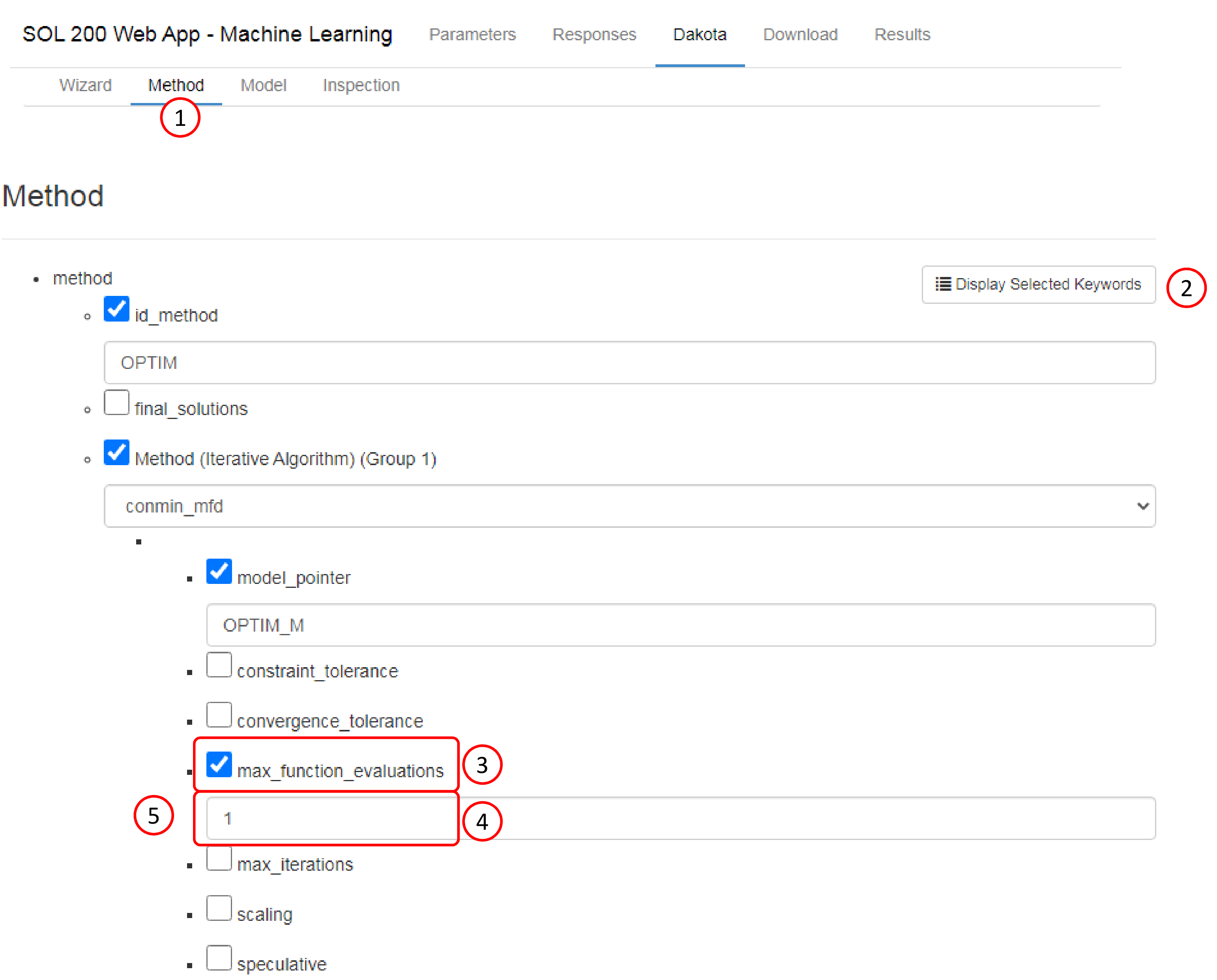
Delete	Descriptor	Status	Probability of Failure for Lower Bound [%]	Lower Bound	Upper Bound	Probability of Failure for Upper Bound [%]
	Search		Search	Search	Search	Search
✖	r1	✔				
✖	r2	✔	5.	-15000	20000	5.
✖	r3	✔	5.	-15000	20000	5.

(2) (3)

Uncertainty Quantification

1. Click Method
2. Click Display Selected Keywords to turn off the option. The button should be white when off.
3. Mark the indicated checkbox to turn on the keyword max_function_evaluations
4. Set the indicated input box to 1
5. 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.

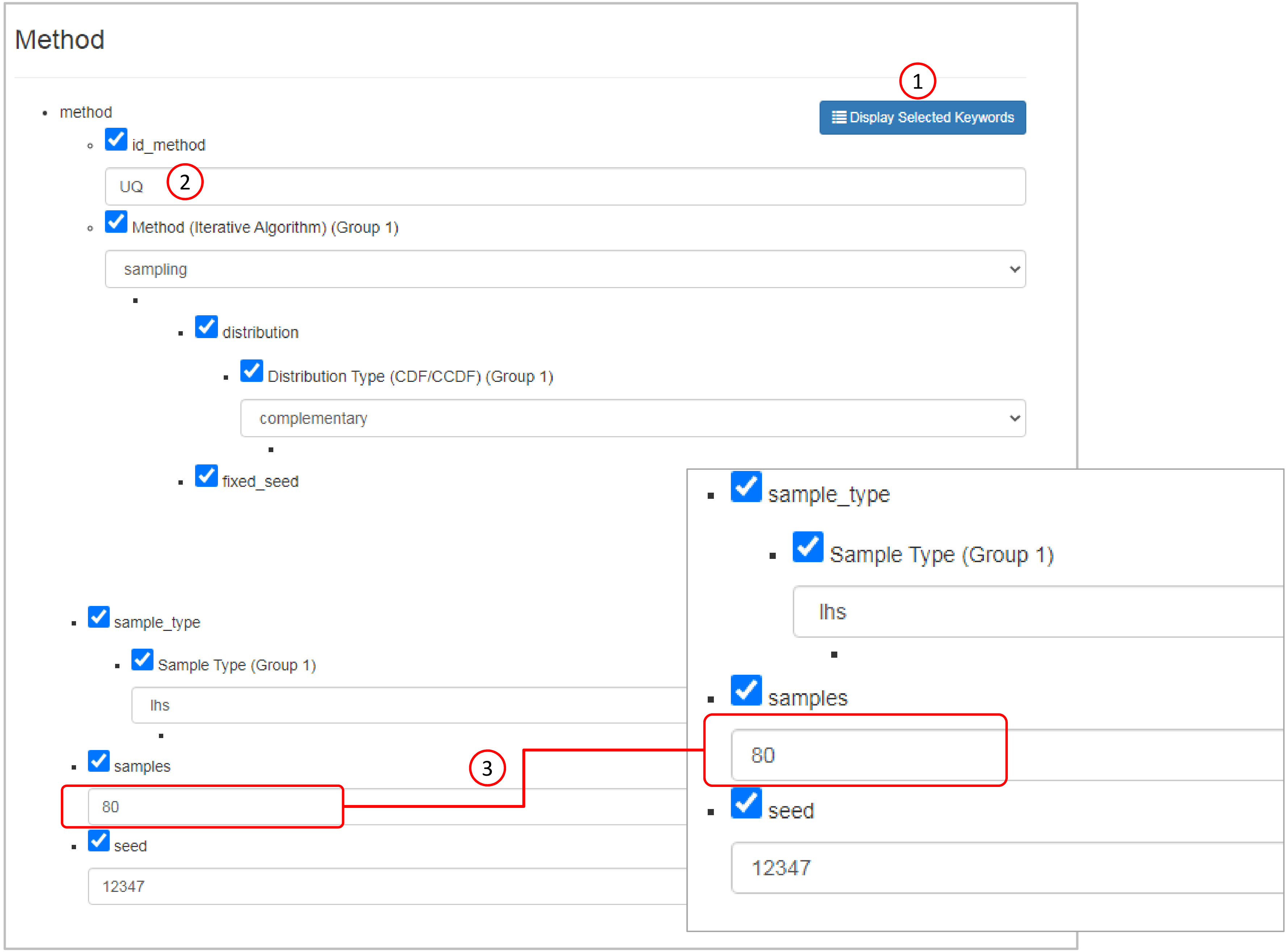


Uncertainty Quantification

1. Click Display Selected Keywords
2. Scroll to the method keyword block with id_method=UQ
3. Change the number of samples to 80.

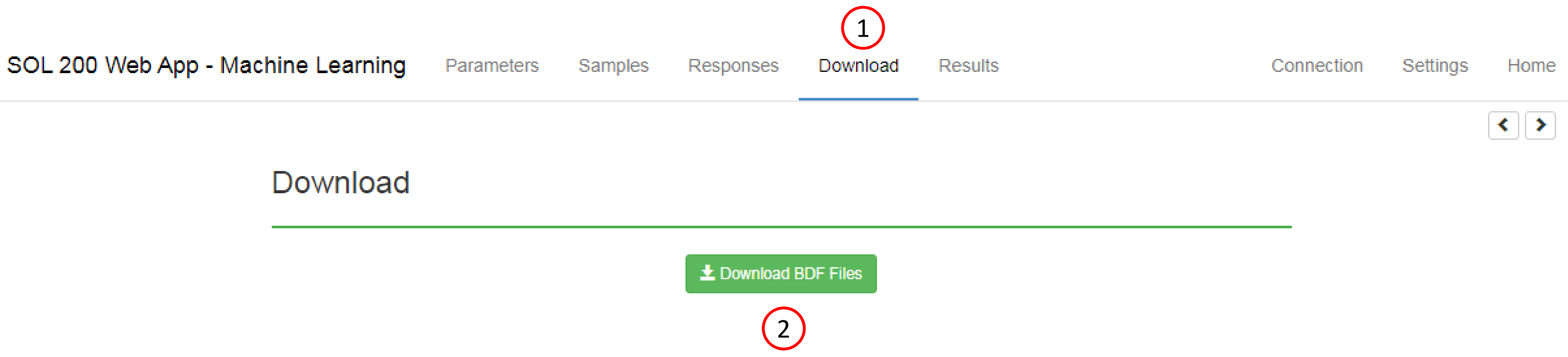
Note the an LHS of size 80 is used to determine the probabilities. This is a contrast to the first part of this tutorial where reliability methods were used to approximate the probabilities. The probabilities at the end of the OUU are approximate, so the goal is to confirm actual probabilities from the LHS are below the max probability of failure of 5%.

- Why 80 runs? Later on, it is shown that a convergence study revealed convergence after 40 runs. 80 runs is used since the cost of each FEA is very low, i.e. it takes less than 1 second for an FEA run, and a high confidence in the probabilities is desired.



Download

1. Click Download
2. Click Download BDF Files

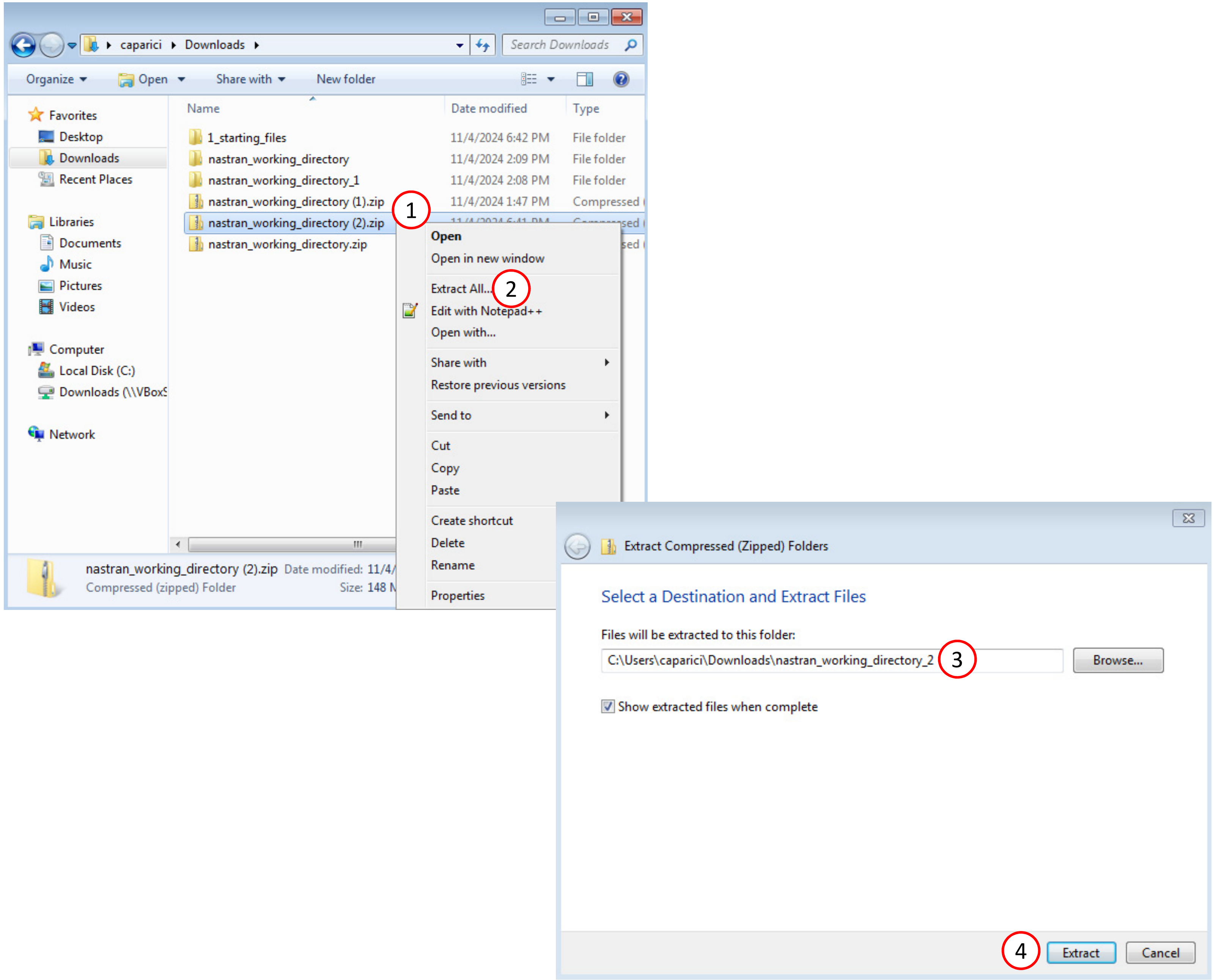


Start MSC Nastran

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_2.
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

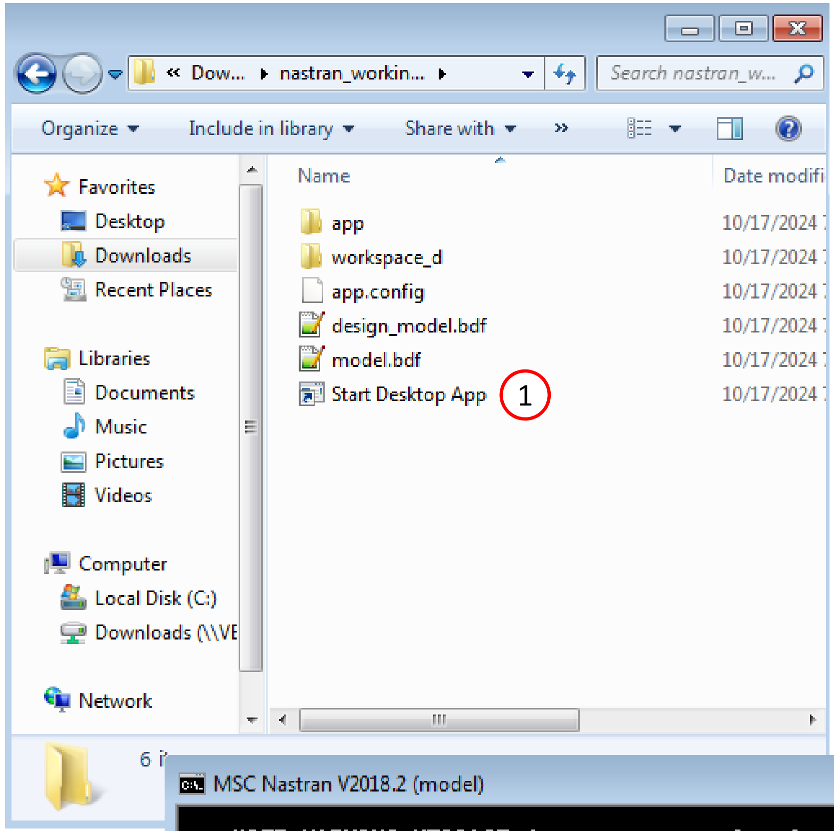
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click “Start Desktop App” to display the results.

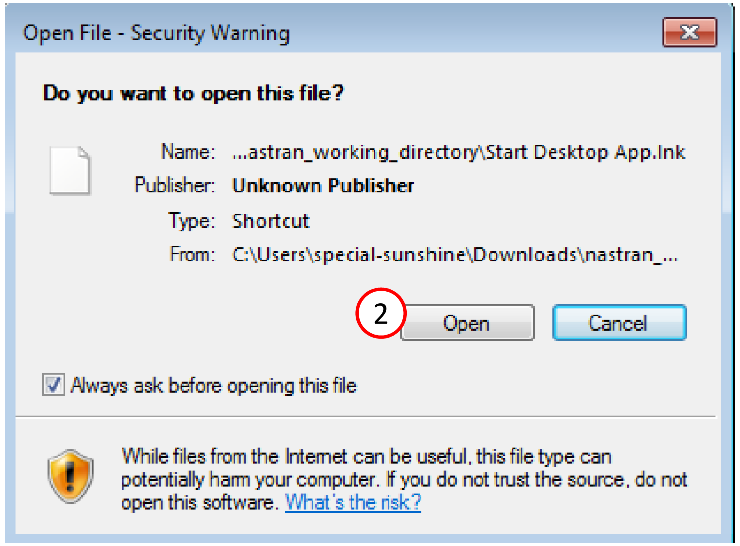
Using Linux?

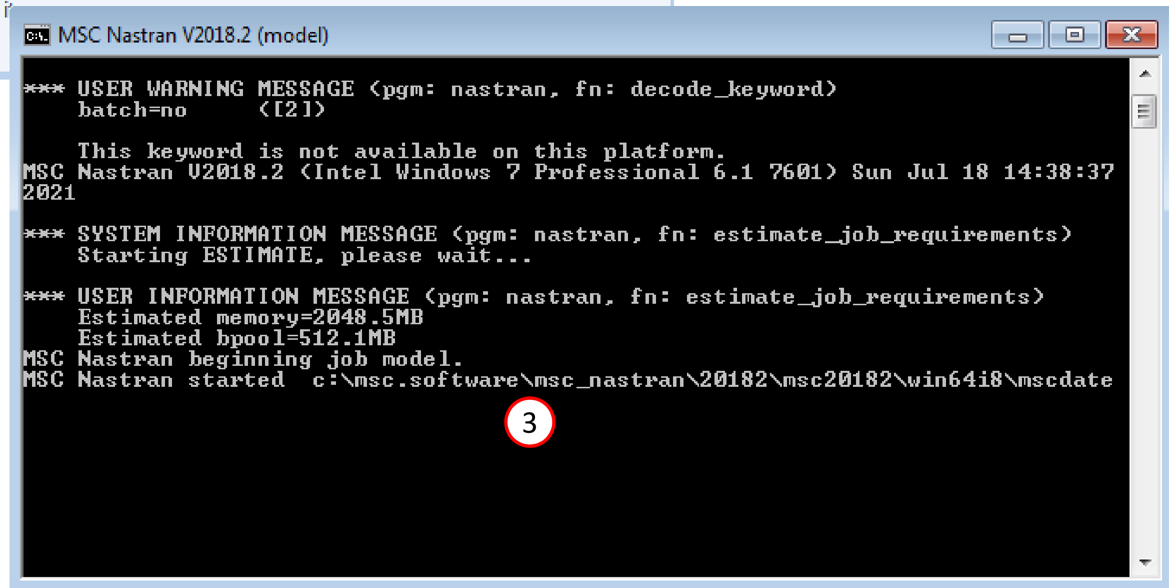
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

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

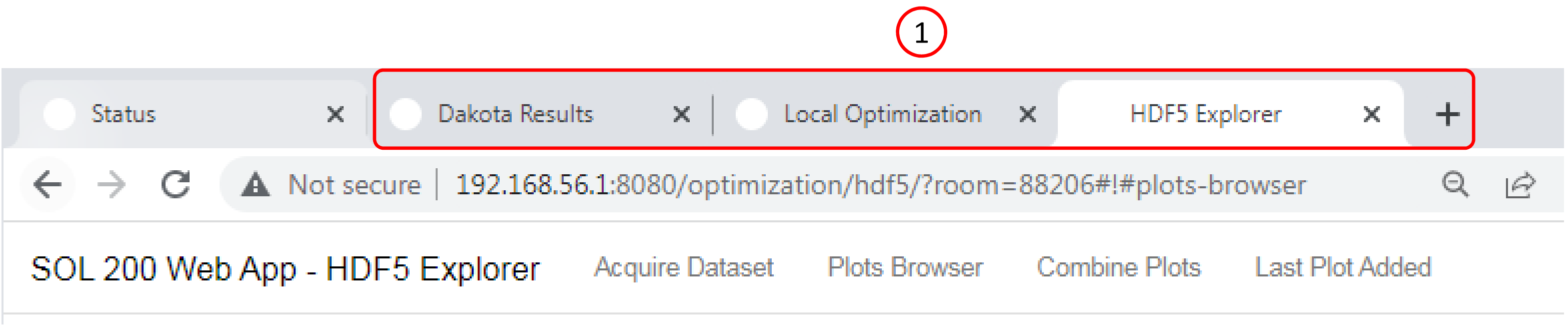
Python MSC Nastran

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 80 (80 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 80 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   80
------------------------------
Parameters for evaluation 80:
                      9.3601787070e-01 x1
                      3.0795175384e-01 x2

blocking fork: ./app/desktop_app_a --analysis_driver_Dakota

Active response data for UQ_I evaluation 80:
Active set vector = { 1 1 1 }
                      2.9554100886e+00 r1
                      1.8273803055e+04 r2
                      1.8273803055e+04 r3
Active response data from sub_iterator:
Active set vector = { 1 0 0 0 1 1 0 0 1 1 }
                      3.0275633258e+00 mean_r1
                     -4.6148586651e+01 ccdf_blev1_r2
                      2.9884090953e+00 ccdf_blev2_r2
                     -4.6148586651e+01 ccdf_blev1_r3
                      2.9884090953e+00 ccdf_blev2_r3
---------------------------
NestedModel Evaluation    1 results:
---------------------------
Active response data from nested mapping:
Active set vector = { 1 1 1 1 1 }
                      3.0275633258e+00 f_obj
                     -4.6148586651e+01 r2_pl
                      2.9884090953e+00 r2_pu
                     -4.6148586651e+01 r3_pl
                      2.9884090953e+00 r3_pu


Iteration terminated: max_function_evaluations limit has been met.
<<<<< Function evaluation summary (UQ_I): 80 total (80 new, 0 duplicate)
<<<<< Best parameters          =
                      9.5130797754e-01 x1_mean
                      3.3670031639e-01 x2_mean
<<<<< Best objective function  =
                      3.0275633258e+00
<<<<< Best constraint values   =
                     -4.6148586651e+01
                      2.9884090953e+00
                     -4.6148586651e+01
                      2.9884090953e+00
```

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
4. Click Display Additional Columns
5. Click Display All as Probability of Failure

There is a goal to identify the most active or most violated constraints. Such constraints are marked with an asterisk (*).

6. The indicated constraints are the most active or most violated
7. The respective probabilities of failure are displayed

Status ✕ | Dakota Results ✕ | Local Optimization ✕ | HDF5 Explorer ✕ | +

1

SOL 200 Web App - Dakota Results   Upload OUT File   Tables   OUU Results

2

OUU Results

Objective

Descriptor	Value
f_obj	3.0275633258e+00

Constraints

4   5

Reset Table   Display Additional Columns   Display All as Probability of Failure

3

Feasible

Descriptor	Lower Bound	Upper Bound	Value	pf/ps	Description	Probability	Reliability Index	Constraint Satisfied
Search								
r2_pl	-inf	-1.644854	-4.6148586651e+01	p_f	P(r2 < -1.5000e+4)	0.0000	-inf	✔
r2_pu *	1.644854	inf	2.9884090953e+00	p_f	P(2.0000e+4 < r2)	1.4022e-3	2.9884090953e+00	✔
r3_pl	-inf	-1.644854	-4.6148586651e+01	p_f	P(r3 < -1.5000e+4)	0.0000	-inf	✔
r3_pu *	1.644854	inf	2.9884090953e+00	p_f	P(2.0000e+4 < r3)	1.4022e-3	2.9884090953e+00	✔

6   7

* This constraint is the most active or most violated

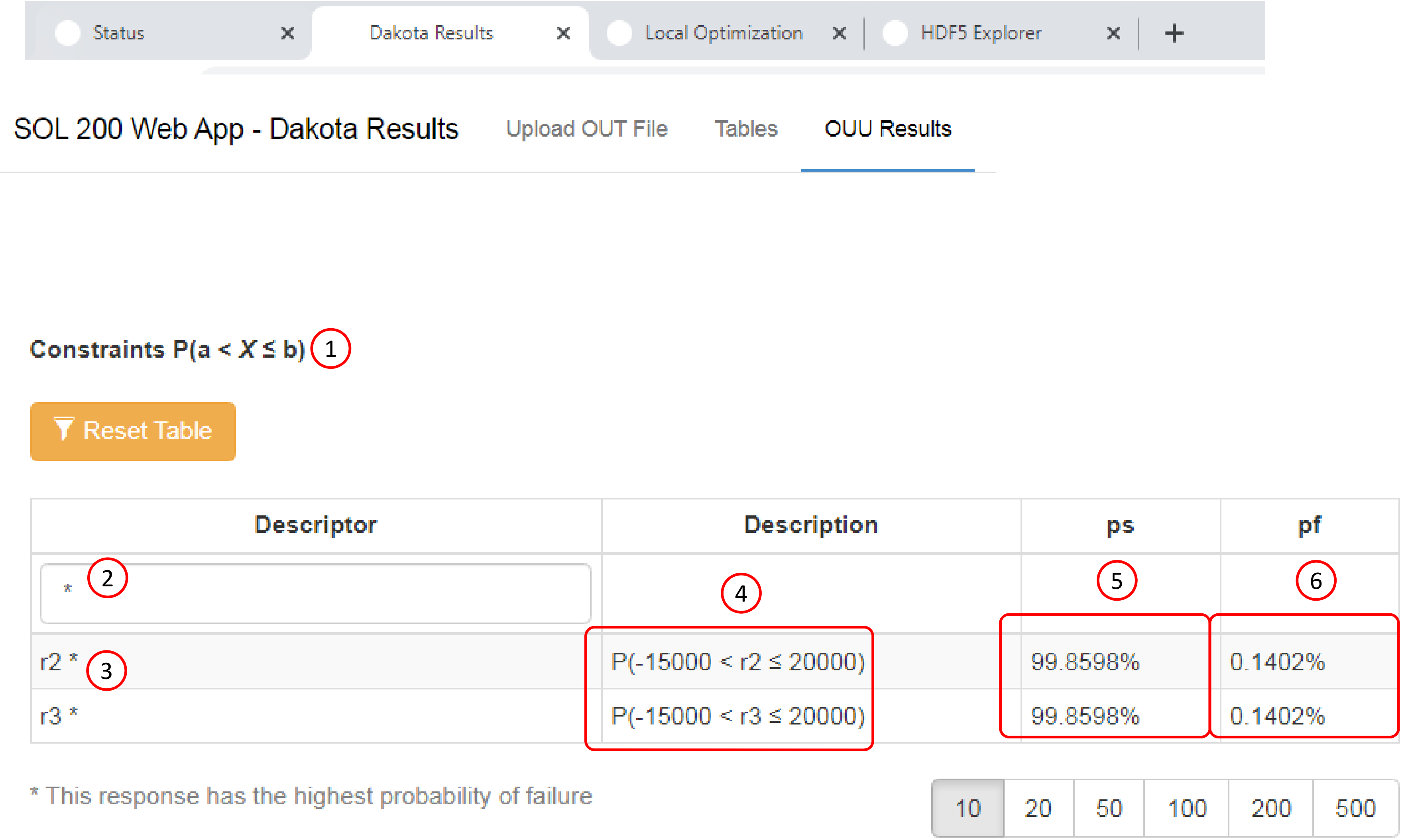
10 | 20 | 50 | 100 | 200 | 500

Discussion of Final Probabilities of Failure

The table on the previous page displayed separate probabilities of failure for the bounds, i.e. P(X < a) and P(b < X). 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 0.1402%, which is well under the desired 5%



Descriptor	Description	ps	pf
*			
r2 *	P(-15000 < r2 ≤ 20000)	99.8598%	0.1402%
r3 *	P(-15000 < r3 ≤ 20000)	99.8598%	0.1402%

Comparison of Approximate and Actual p_f

- It is seen the actual and worst case p_f of 0.1402% is well under the desired 5%.
- The OUU in part B and the verification of probabilities in part C has been a success.

Response	Part B - OUU Approximated Reliability Indices (Equivalent Probability of Failure)	Part C - Verification (Actual Probability of Failure)
Comments	The OUU considered and reported reliability indices	These are the probabilities after an LHS of size 80
r2, lower bound	-4.5383296250e+01 (pf=0.0%)	(pf=0.0%)
r2, upper bound	1.6448457983e+00 (pf=5%)	(pf=0.14022%)
r3, lower bound	-4.5383296250e+01 (pf=0.0%)	(pf=0.0%)
r3, upper bound	1.6448457983e+00 (pf=5%)	(pf=0.14022%)

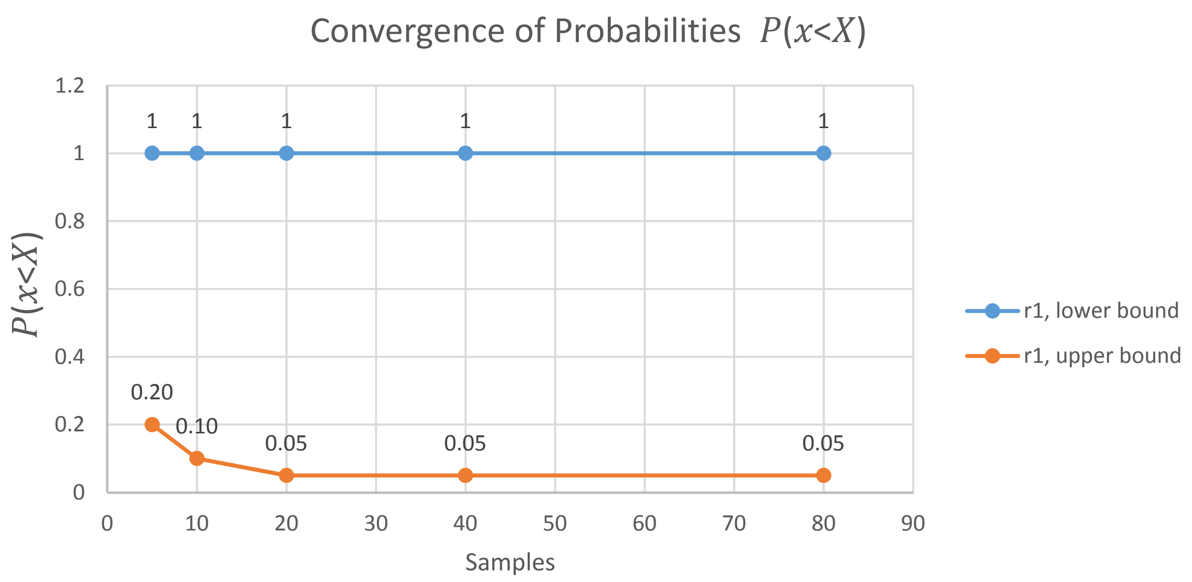
Do the tail probabilities converge?

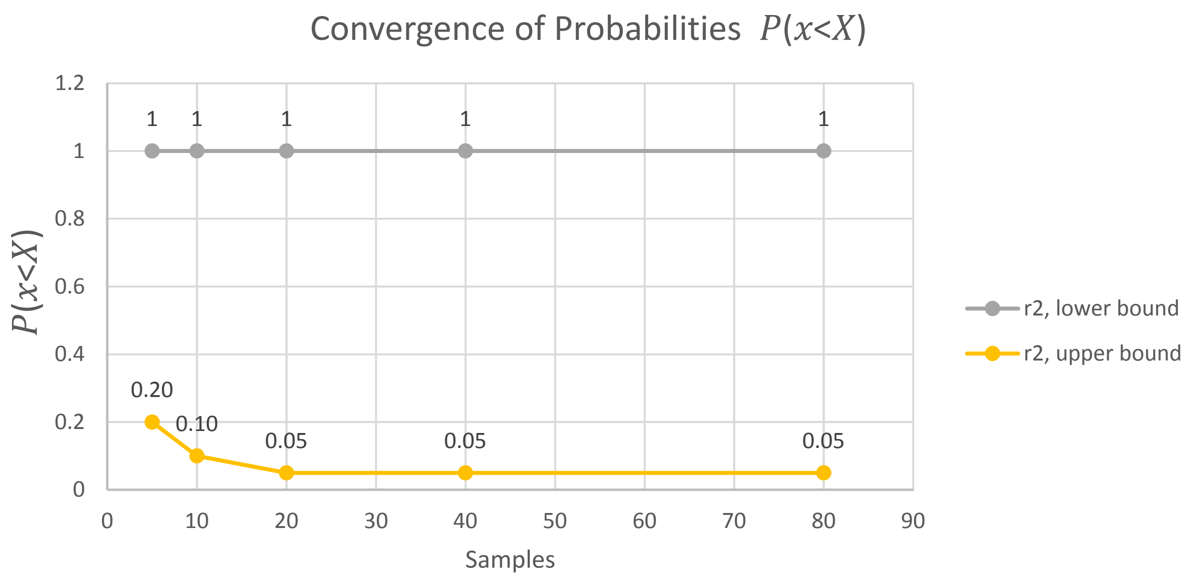
Before this exercise, a UQ involved an incremental sampling approach of 5, 10, 20, 40 and 80 runs. Incremental sampling is triggered with the keyword refinement_samples. The sampling type was LHS.

Do the tail probabilities converge?

A plot of the probabilities indicates convergence. The probabilities after 40 runs are labeled as *actual* probabilities. The probabilities after 40 runs are not significantly different from the probabilities after 80 runs. To reduce computational cost, UQ with 40 runs may be considered instead. Lastly, these runs were configured with a random seed of 1337. It is recommended to repeat the UQ with different seed values to confirm the probabilities are consistent after repeated randomization of the sampling points.

```
method
   id_method 'UQ'
      sampling
         distribution
               complementary
         fixed_seed
         refinement_samples  5  10  20
         response_levels  -20000
            compute
                  probabilities
            num_response_levels
         sample_type
               lhs
         samples 5
         seed 1337
```





Clarification On Reliability Indices

Some readers may have noticed the following difference in part A and B, and this difference is discussed here.

In part A, the following probability was defined and constrained,

$$P(a < X \leq b)$$

$$= P(X \leq b) - P(X \leq a)$$

In part B, each bound was assigned its own reliability index and was internally constrained.

$$\beta(a) < -1.644854$$

$$1.644854 < \beta(b)$$

Some readers may ask, “why not combine the reliability indices and constrain the combined value?”, e.g.

$$\beta(b) - \beta(a)$$

Configure OUU Constraints

Statistics to compute at each response level

Reliabilities

Reset Table

Delete	Descriptor	Status	Probability of Failure for Lower Bound [%]	Lower Bound	Upper Bound	Probability of Failure for Upper Bound [%]
	Search		Search	Search	Search	Search
✖	r1	✔				
✖	r2	✔	5.	-14250	19000	5.
✖	r3	✔	5.	-14250	19000	5.

$P(X < a)$ $P(b < X)$

Clarification On Reliability Indices

The CDF or CCDF values measure the area under the PDF curve, so the area between two values a and b is simply

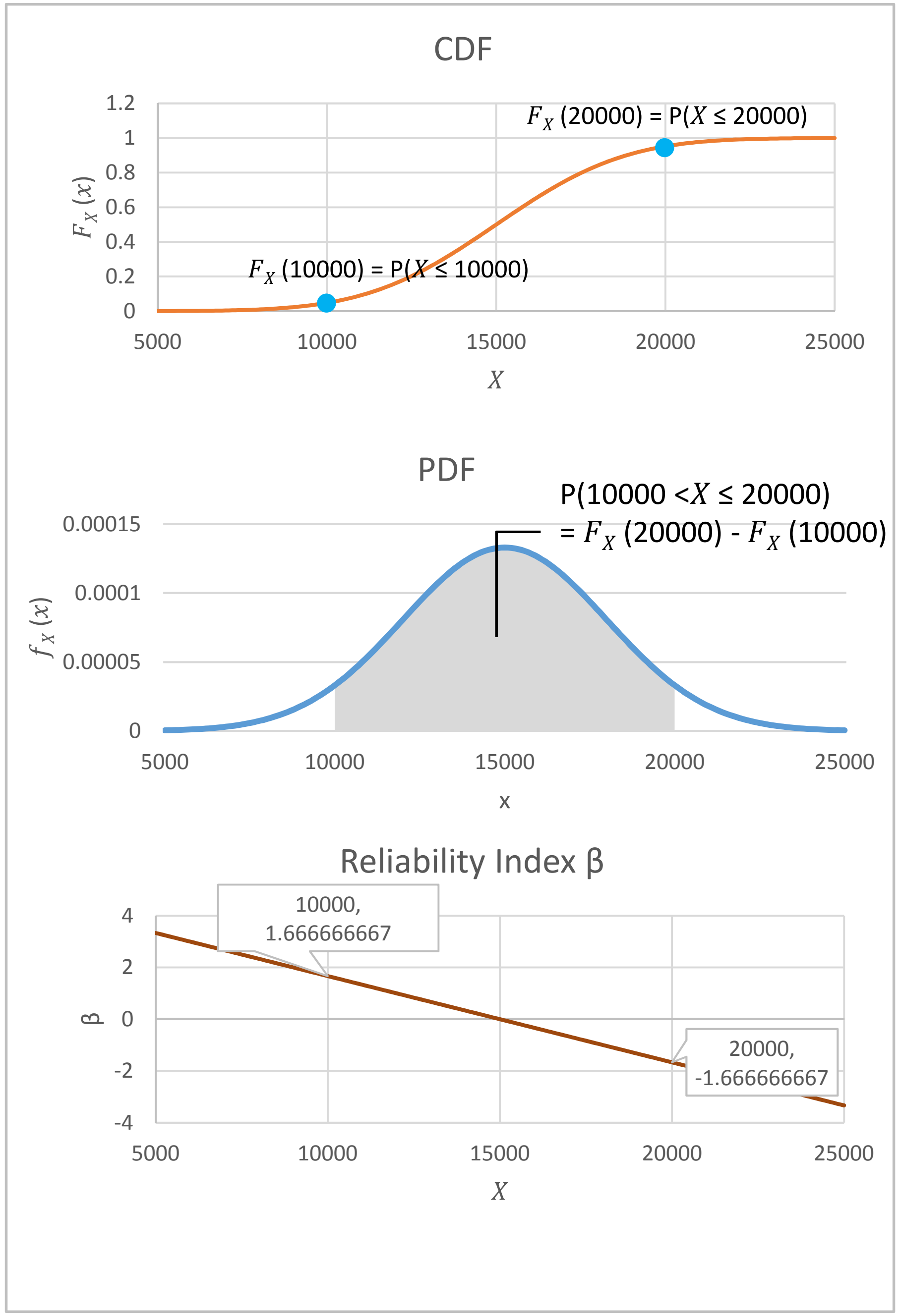
$$P(a < X \le b)$$

$$= P(X \le b) - P(X \le a)$$

$$= F_X(b) - F_X(a)$$

The same cannot be said for reliability indices.

$$P(a < X \le b) \ne \beta(b) - \beta(a)$$



Clarification On Reliability Indices

This probability P(a < X ≤ b) is the area under the PDF curve and can be thought of having units of area.

The reliability index is the number of standard deviations between the mean and response level a or b. The reliability index can be thought of being unitless.

The expression below yields unitless values.

β(b) - β(a)

but this expression has units of area

P(a < X ≤ b) = F_X (b) - F_X (a).

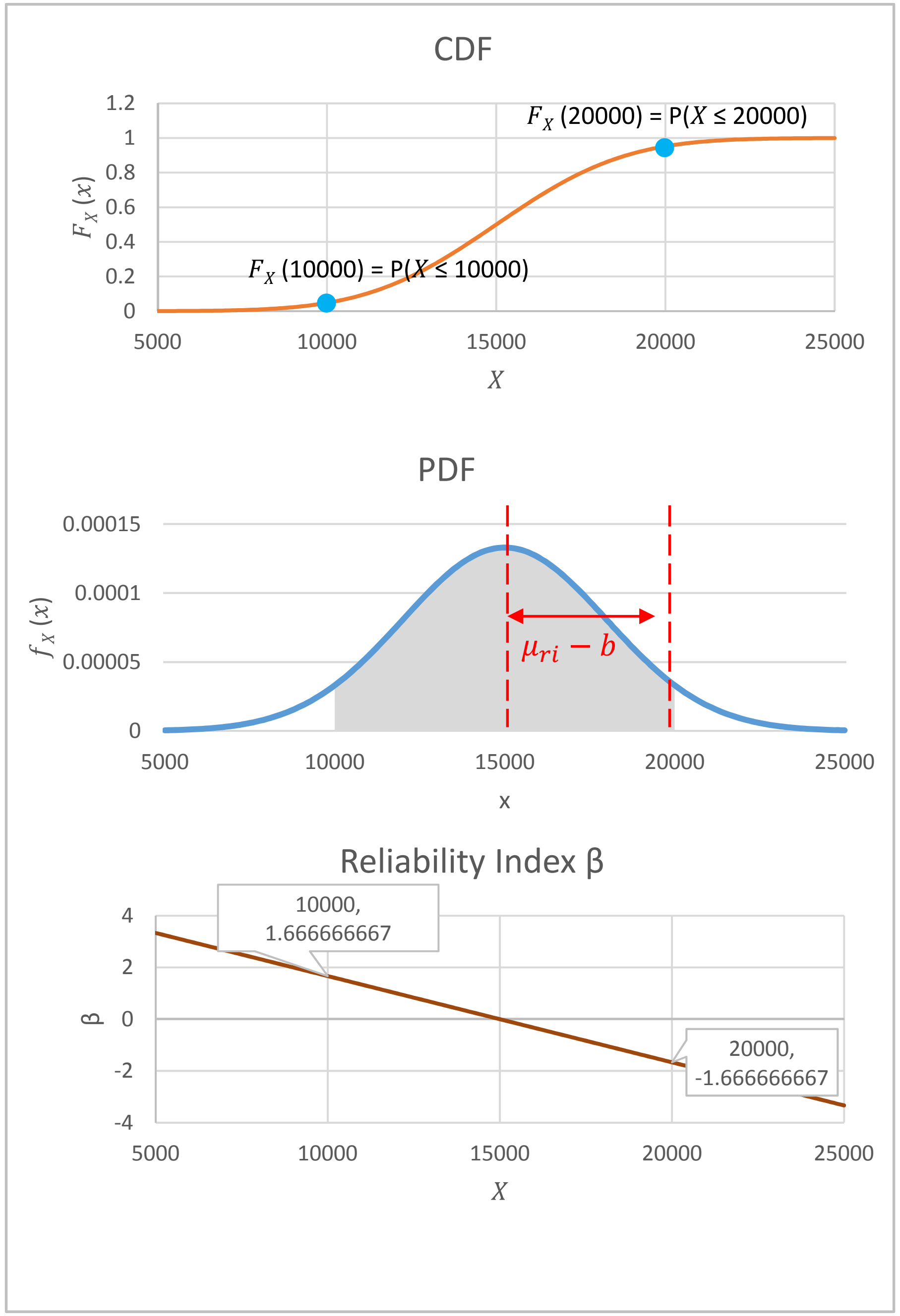
Therefore, this expression is not valid.

P(a < X ≤ b) ≠ β(b) - β(a)

The reliability indices are constrained separately, not together.

β(a) < -1.644854

1.644854 < β(b)



Reliability Index Expression

$$\beta(b) = \frac{\mu_{ri} - b}{\sigma_{ri}}$$

Questions? Email: christian@ the-engineering-lab.com

HEXAGON Technology Partner

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                           (5)
      sample_type
            lhs                  (6)
      samples 5000
      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) (2)

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

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.

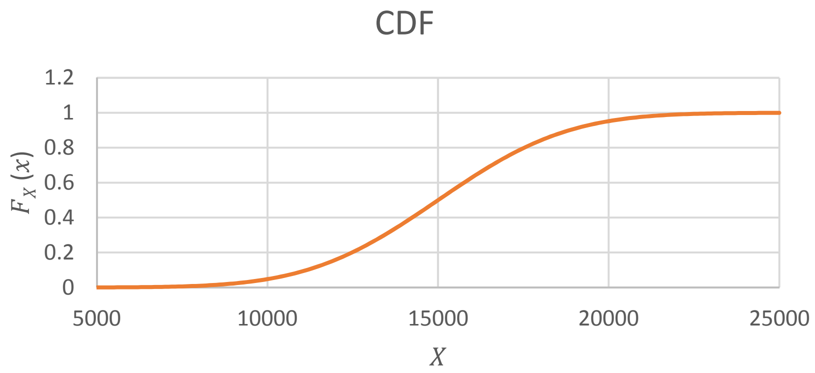
It must be decided if CDF or CCDF values are used throughout the UQ or OUU.

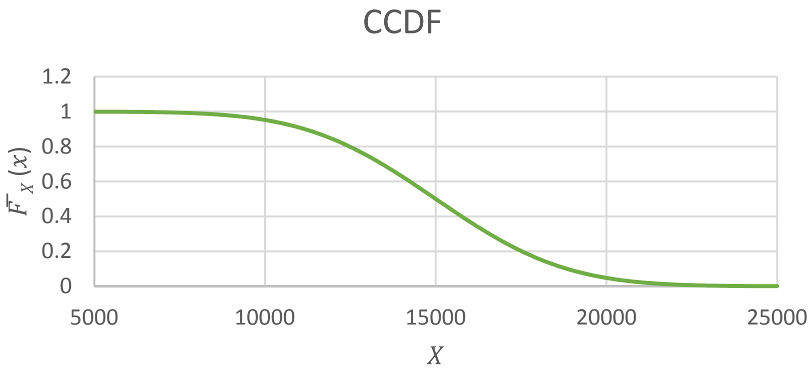
The CDF and CCDF are related by the following relationships

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).

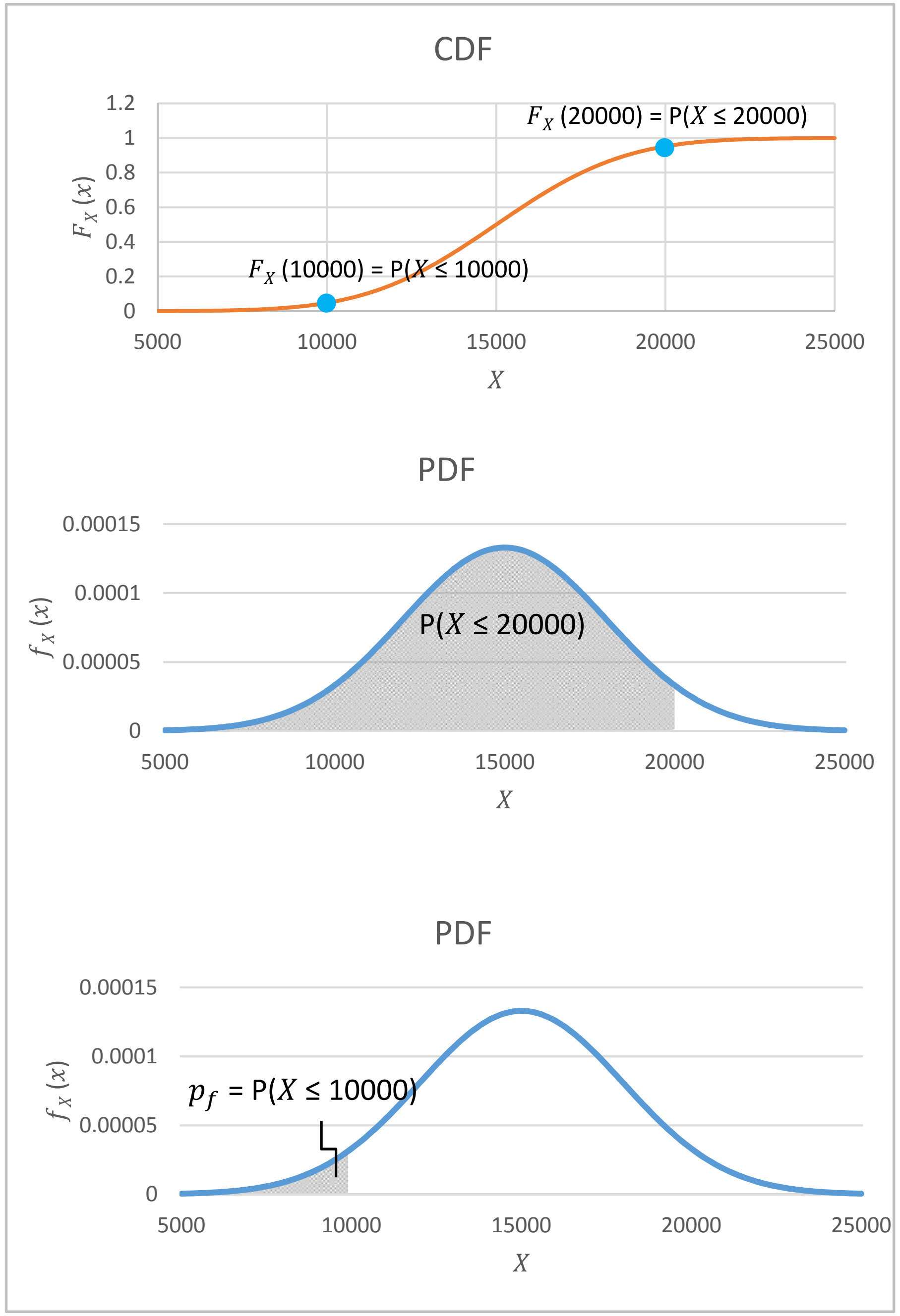
- For the lower bound, if CDF values are used, the probability of failure is

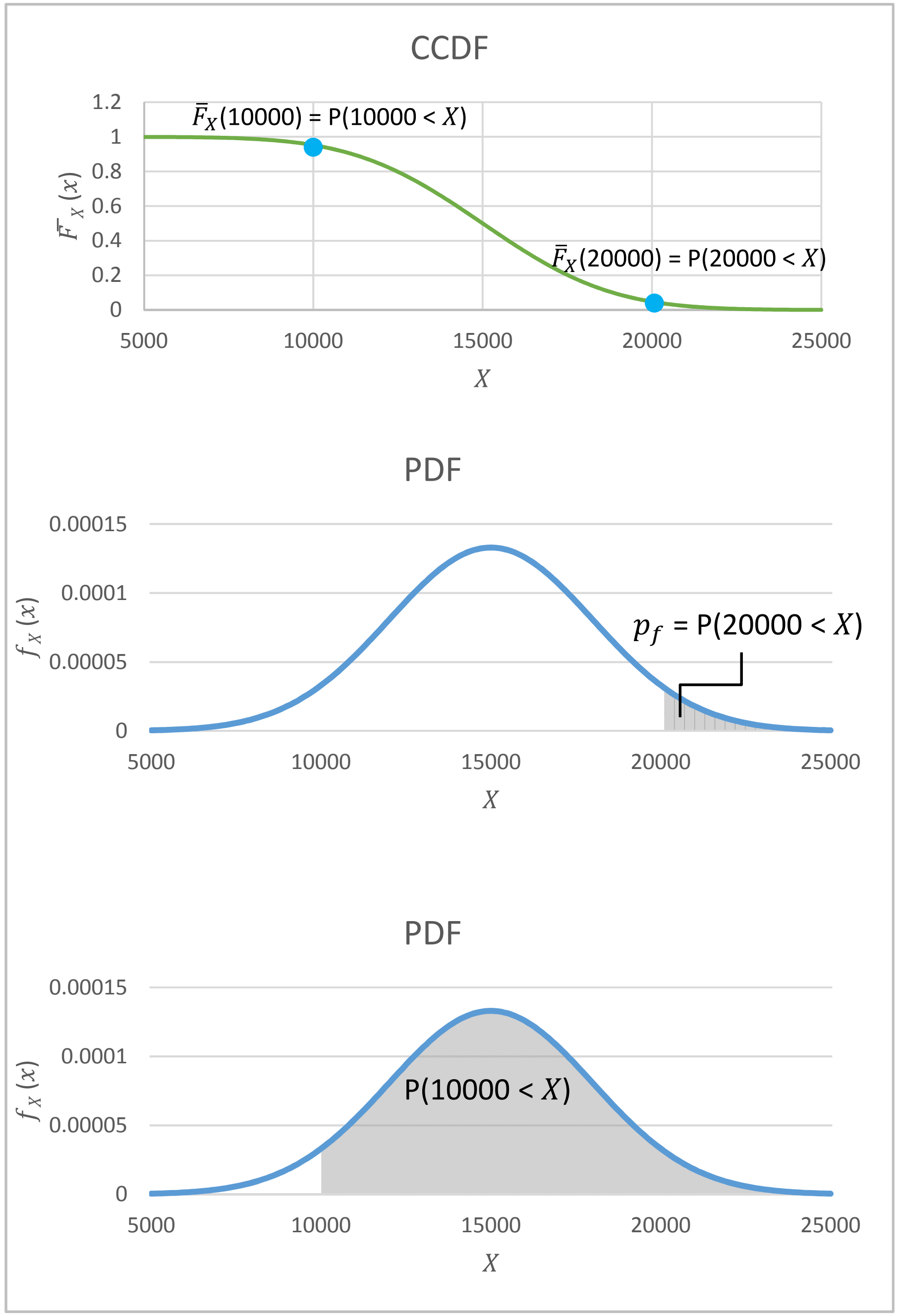
  p_f = P(X ≤ 10000).

- 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($X \leq x$).
- 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($X \leq 10000$)

p_s = P($X \leq 20000$)

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%.

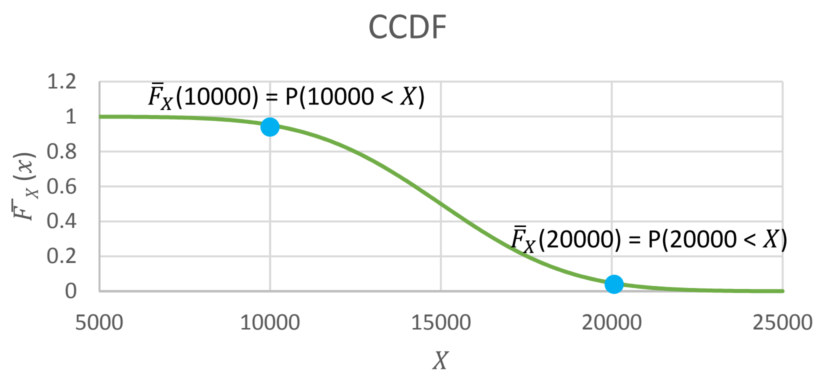
$$p_f = \text{P}(20000 < X) < 0.05$$

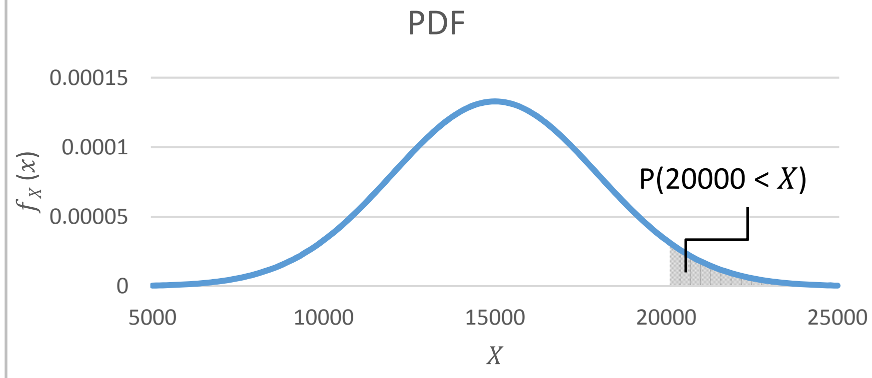
- For the lower bound, the quantity available is the probability of survival. If at the most, a 5% probability of failure is imposed, this is equivalent to saying the probability of survival is greater than 95%. The constraint on the probability of survival is as follows:

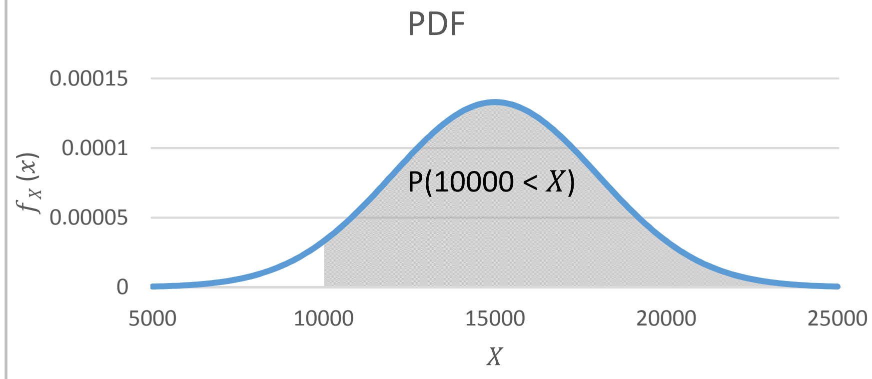
$$0.95 < p_s = \text{P}(10000 \leq X).$$

Dakota Input File

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

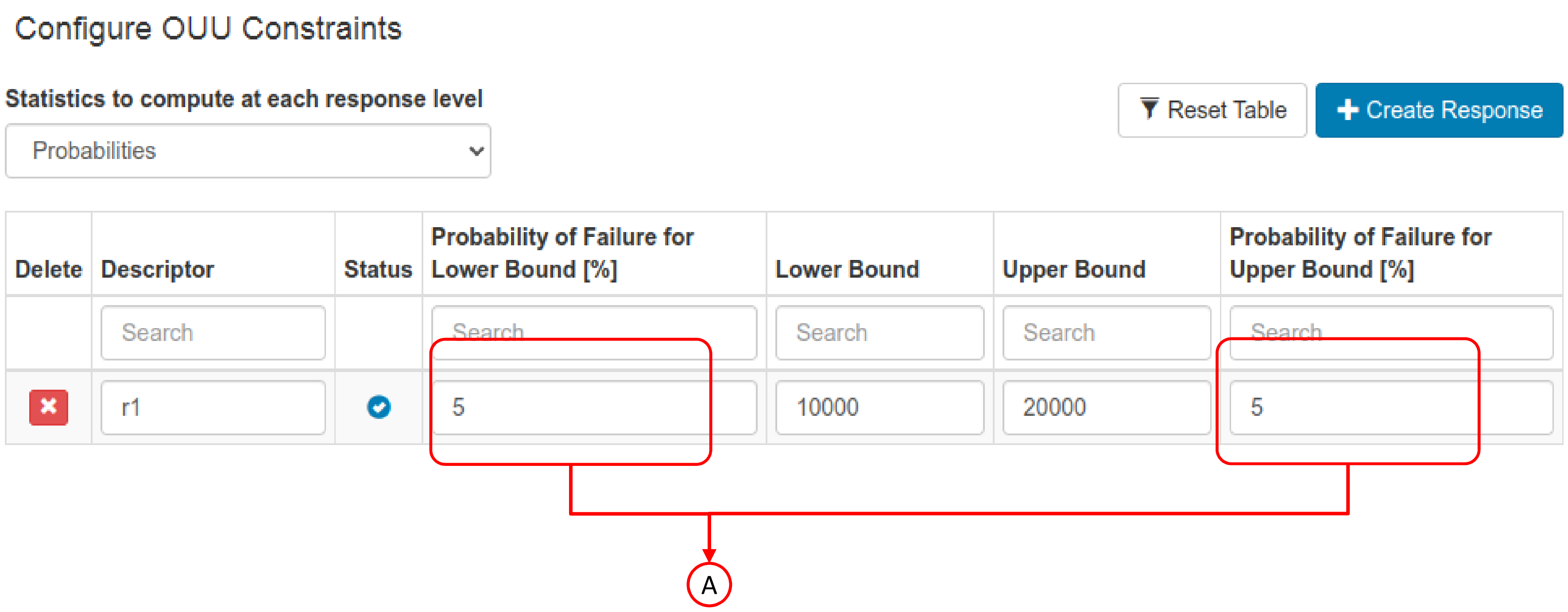






CDF and CCDF Values

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:

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

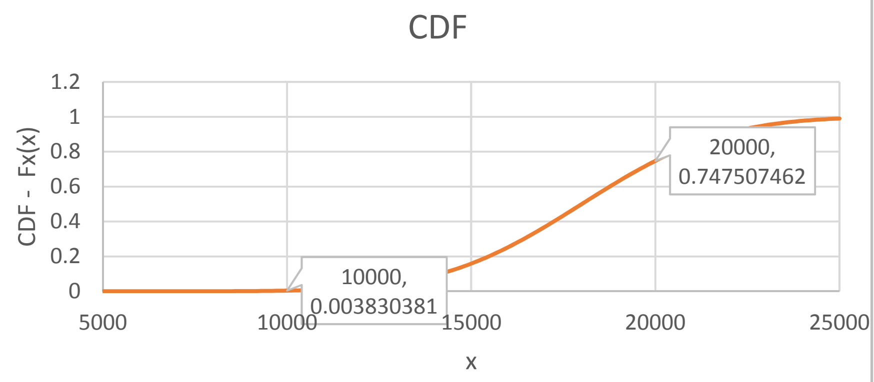
While this is valid, there is a drawback. A single probability value does not indicate if the distribution is violating the lower or upper bound.

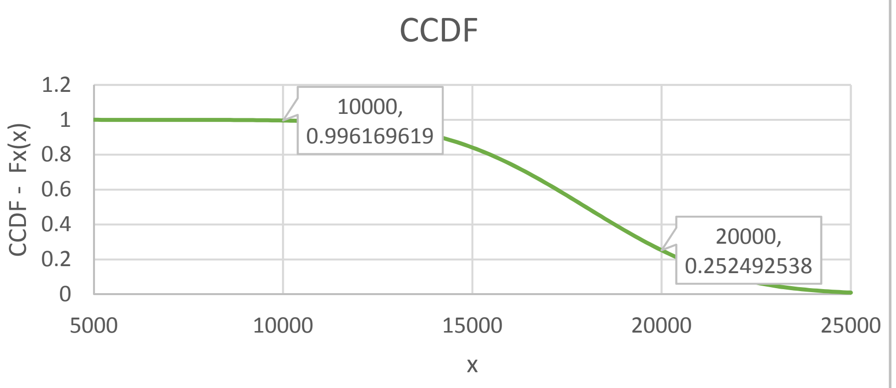
For example, suppose the following single probability is used: $P(10000 < X \leq 20000) = 0.74$ (74% survival). Since this single probability is less than the desired 95%, failure is expected. With a single probability, it is not known if the distribution is violating the lower or upper bound.

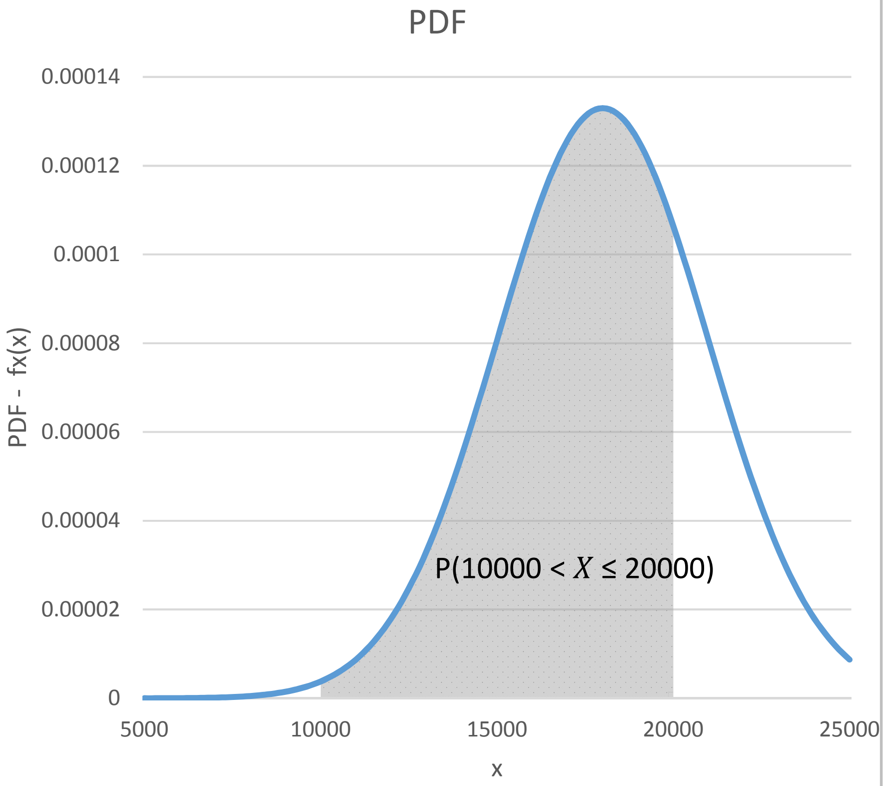
If separate probabilities are constrained, one for the lower and upper bounds, it makes it simpler to identify which of the bounds is being violated.

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.

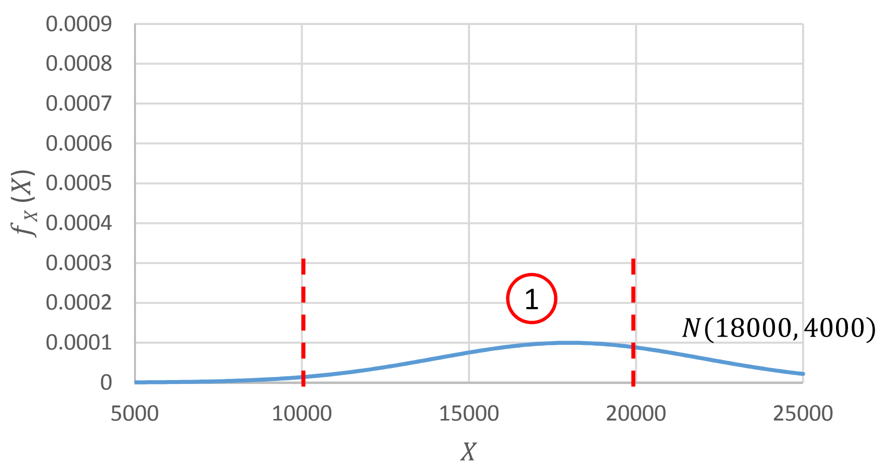
1. The standard deviation is too large and the probabilities of failure for both the lower and upper bound are greater than 5%. The design is infeasible.

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.

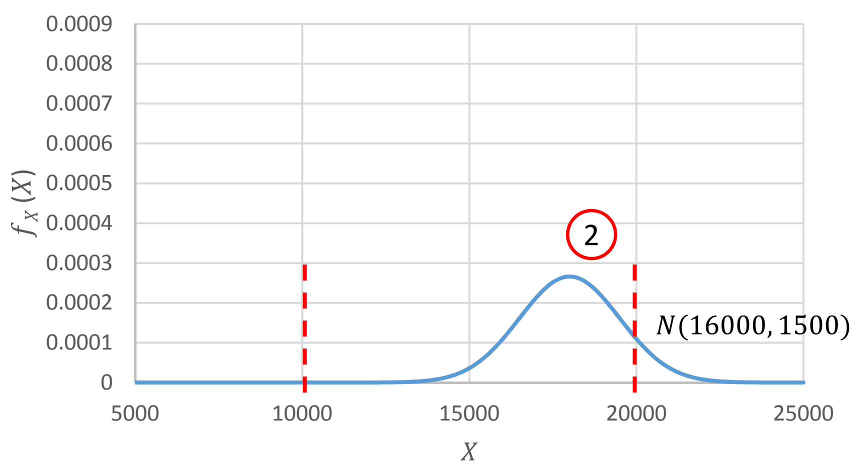
3. The mean is approximately half way between the lower and upper bound and yields a probability of failure within 5% for both lower and upper bounds. The design is feasible.

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.

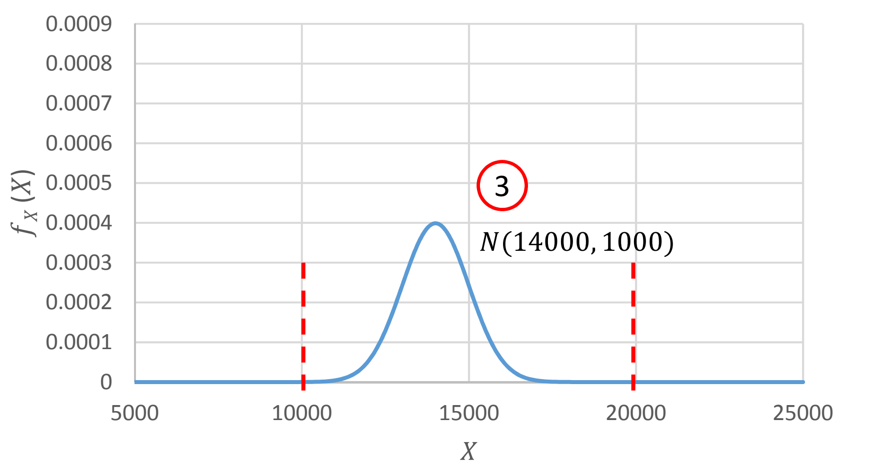
PDF of Response – Design Cycle 1



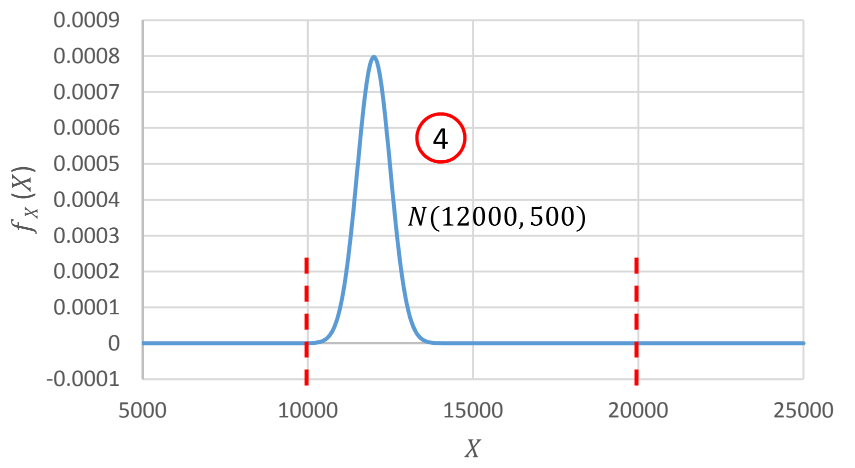
PDF of Response – Design Cycle 2



PDF of Response – Design Cycle 3



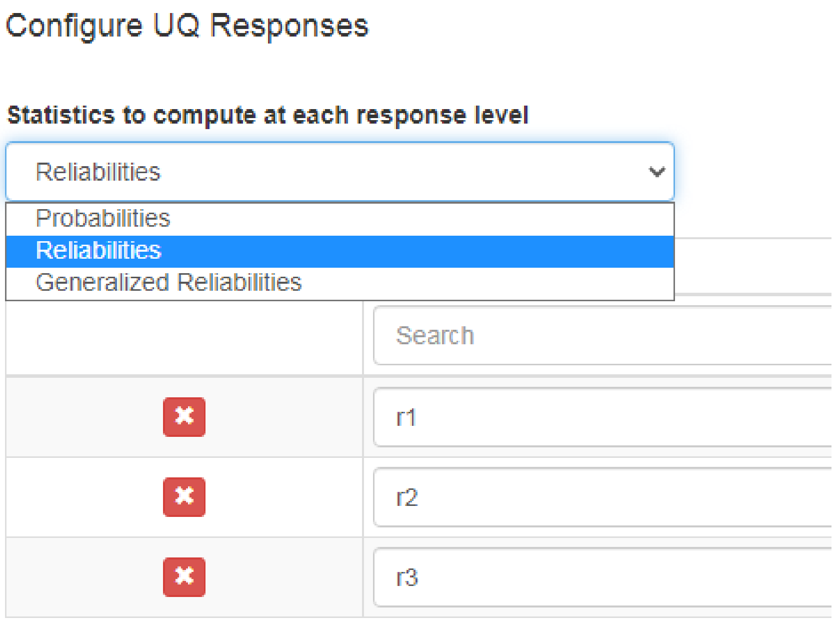
PDF of Response – Design Cycle 4



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($X \leq$ 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.

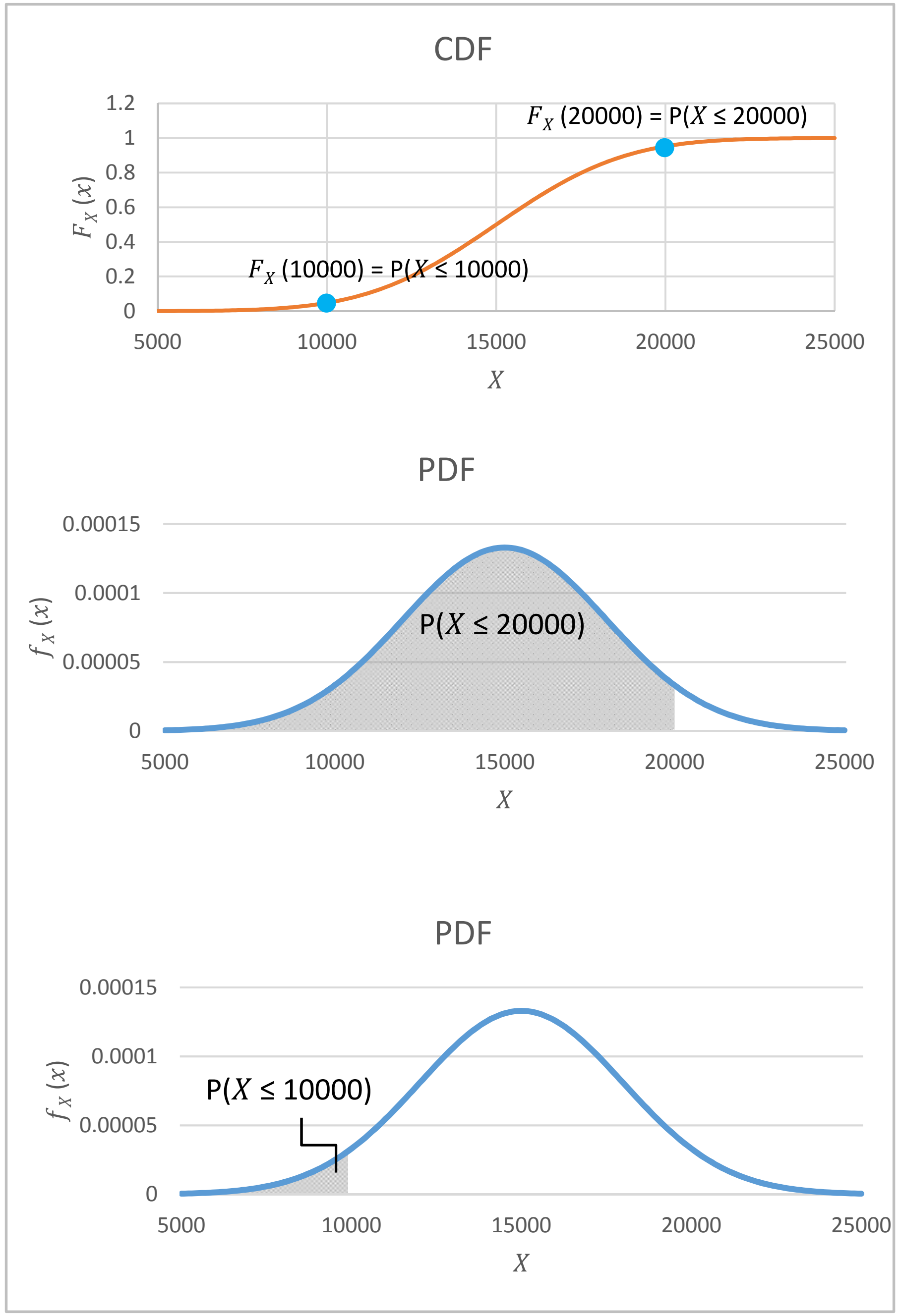
- P($X \leq$ 20000)
- P($X \leq$ 10000)

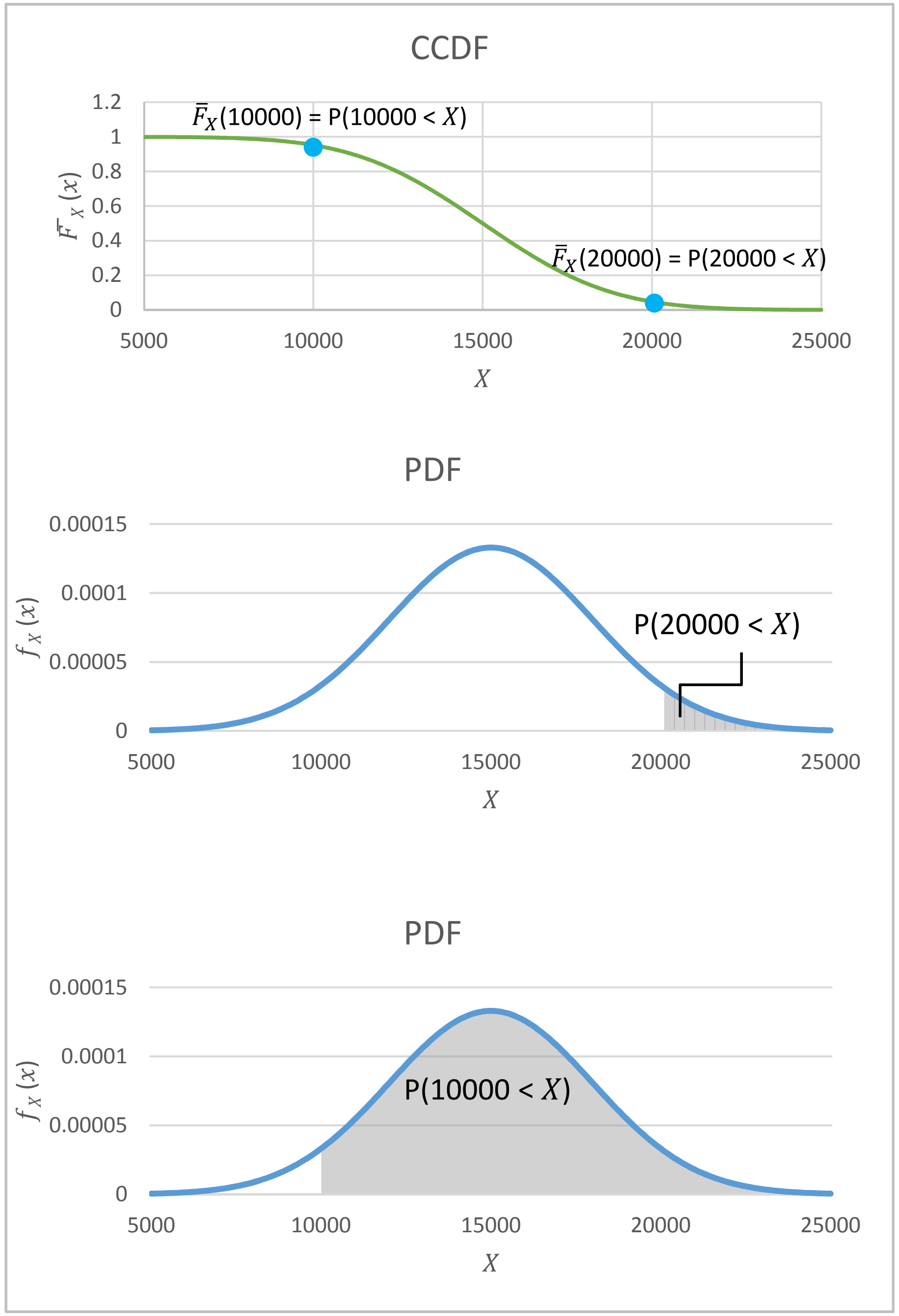
If complementary cumulative distribution function (CCDF) values are available, the following probabilities may be determined.

- 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.

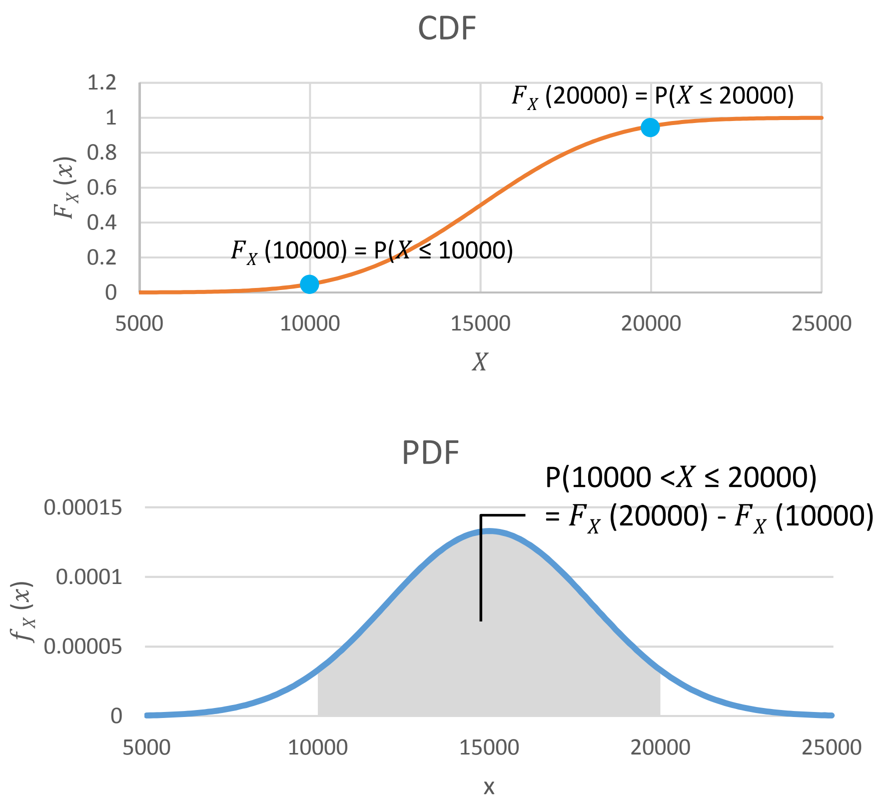
- P(10000 < X ≤ 20000)

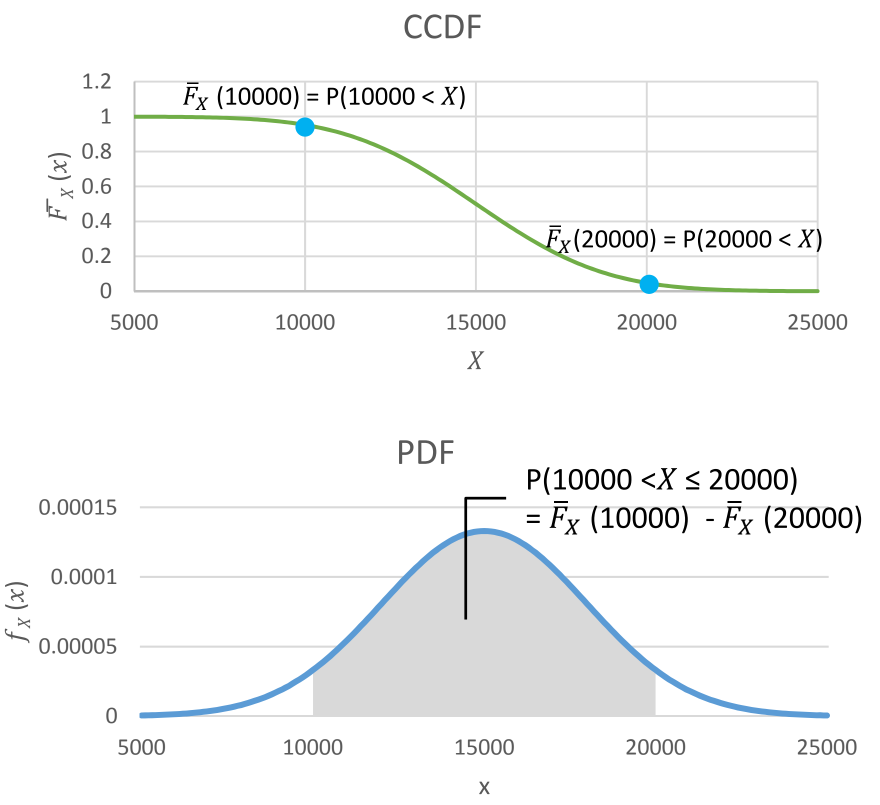
If cumulative distribution function (CDF) values are available, this probability may be determined as follows.

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

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

Where erf is defined as

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

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

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

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:

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

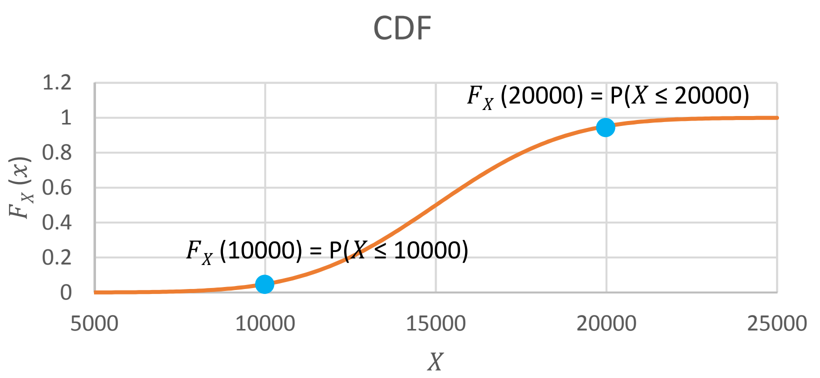
When the CCDF option is used, the probability and reliability index $\bar{\beta}$ are related via the following expression:

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

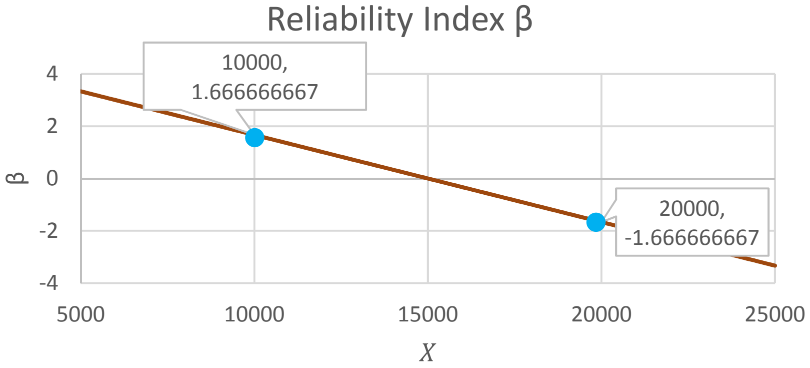
Constraining reliability indices is equivalent to constraining probabilities.

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.

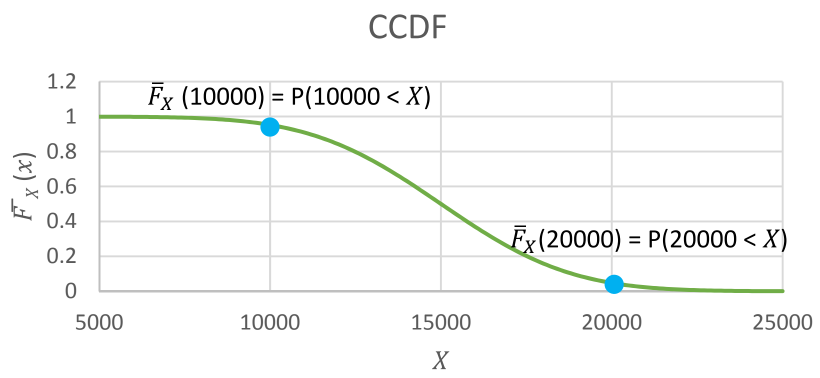


$$p(X \leq x) = F_X(x) = \Phi(-\beta)$$

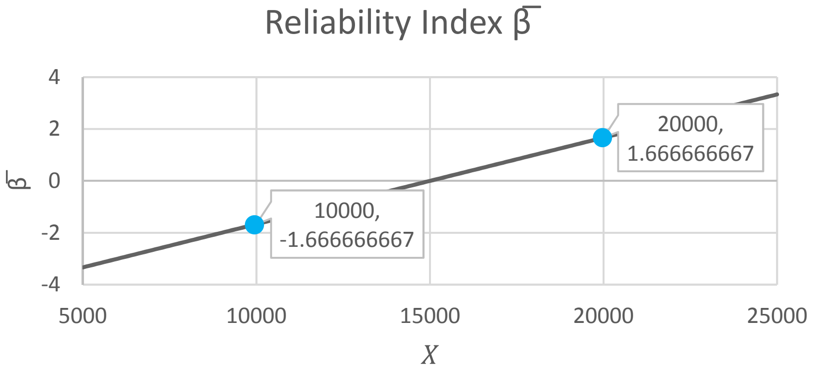


$$\beta = \frac{15000 - X}{3000}$$

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



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



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

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

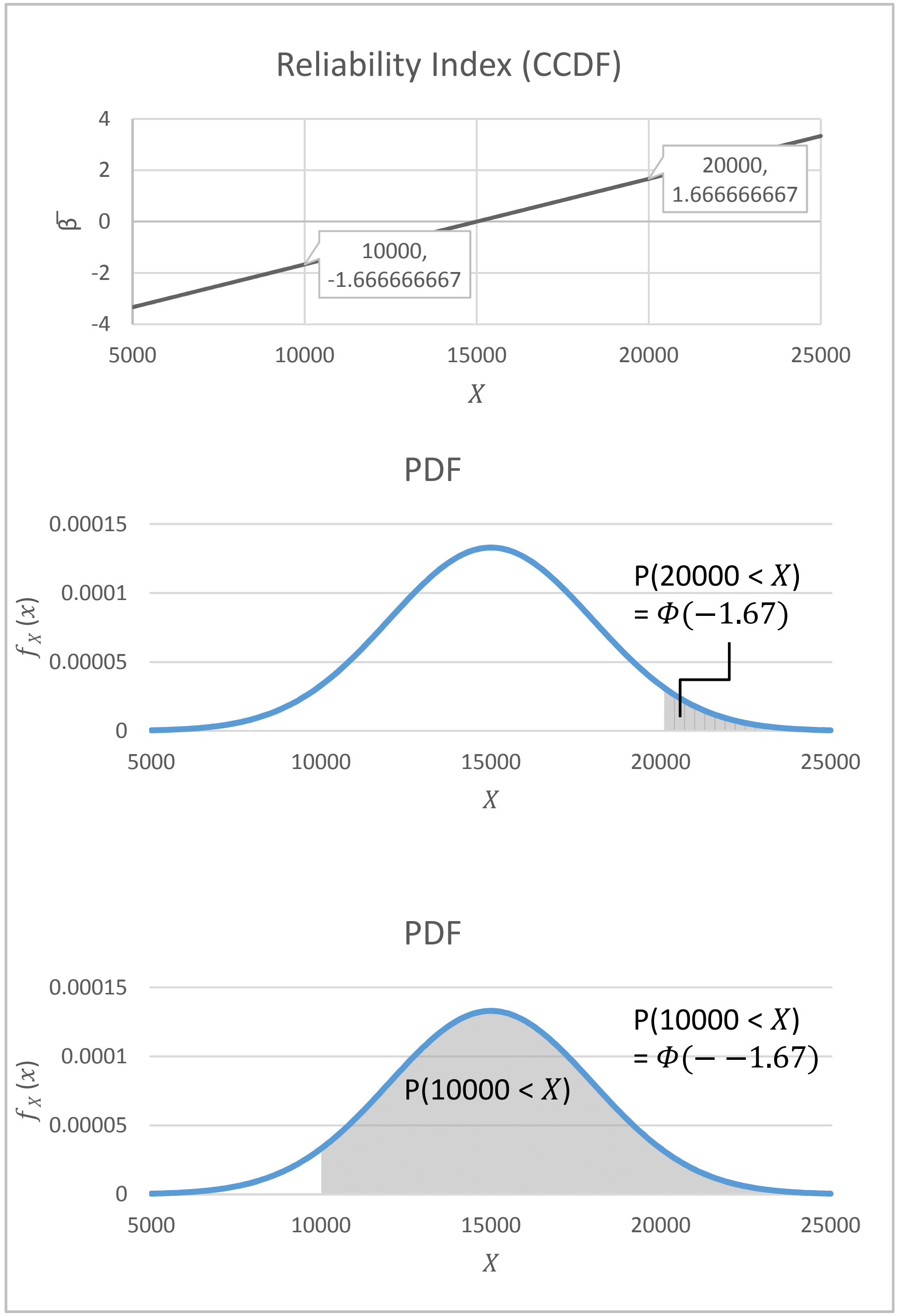
$p_{f,\,upper}$ = P($20000 < X$) < 0.05

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

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

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

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



HEXAGON Technology Partner

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
```

(3) (25 new,

(1) p_s = P(-20000 < X)

p_f = P(20000 < X) (2)

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

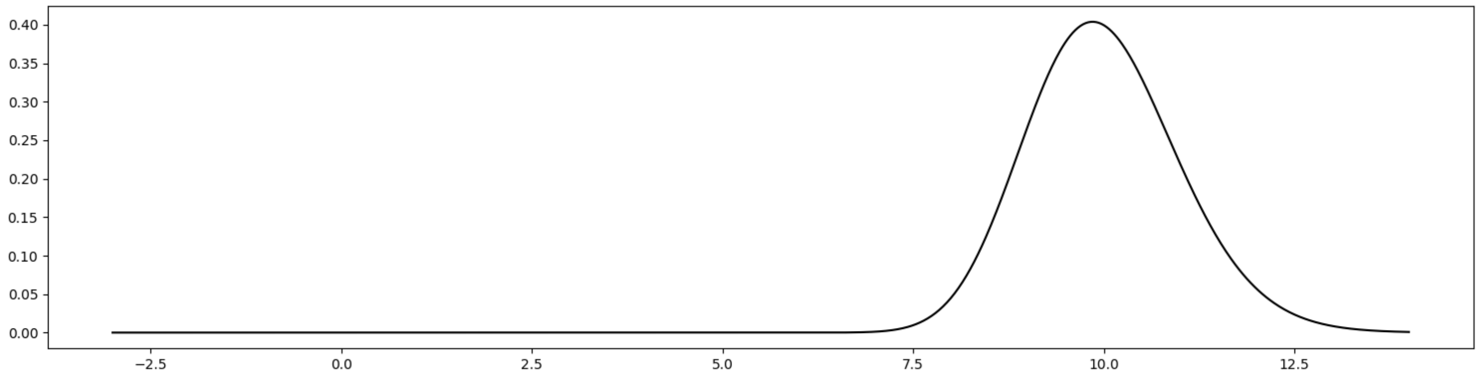
The following applies if uncertain variables have a normal or lognormal distribution.

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.

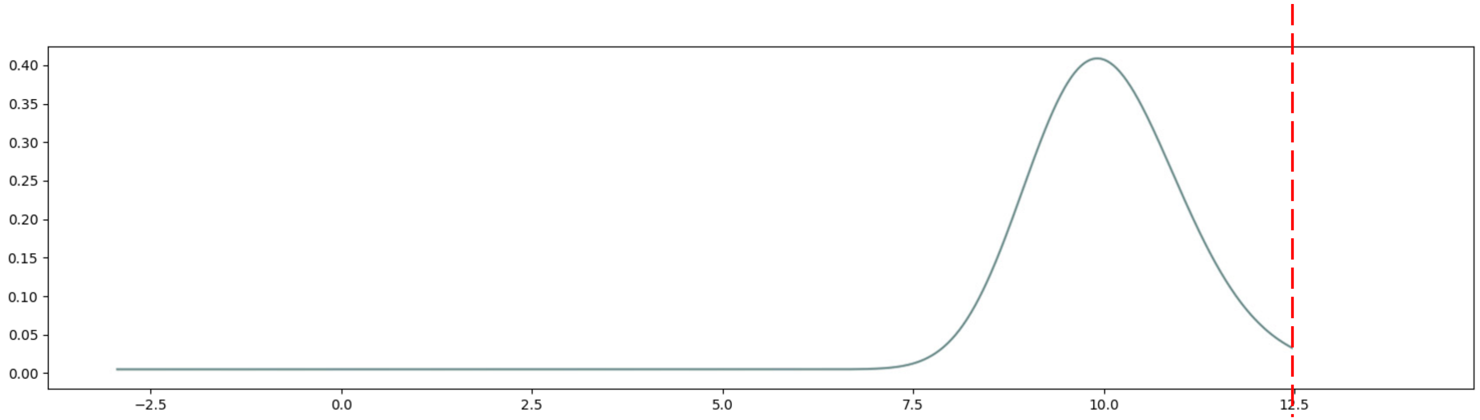
```
File LHS.ERR

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*.

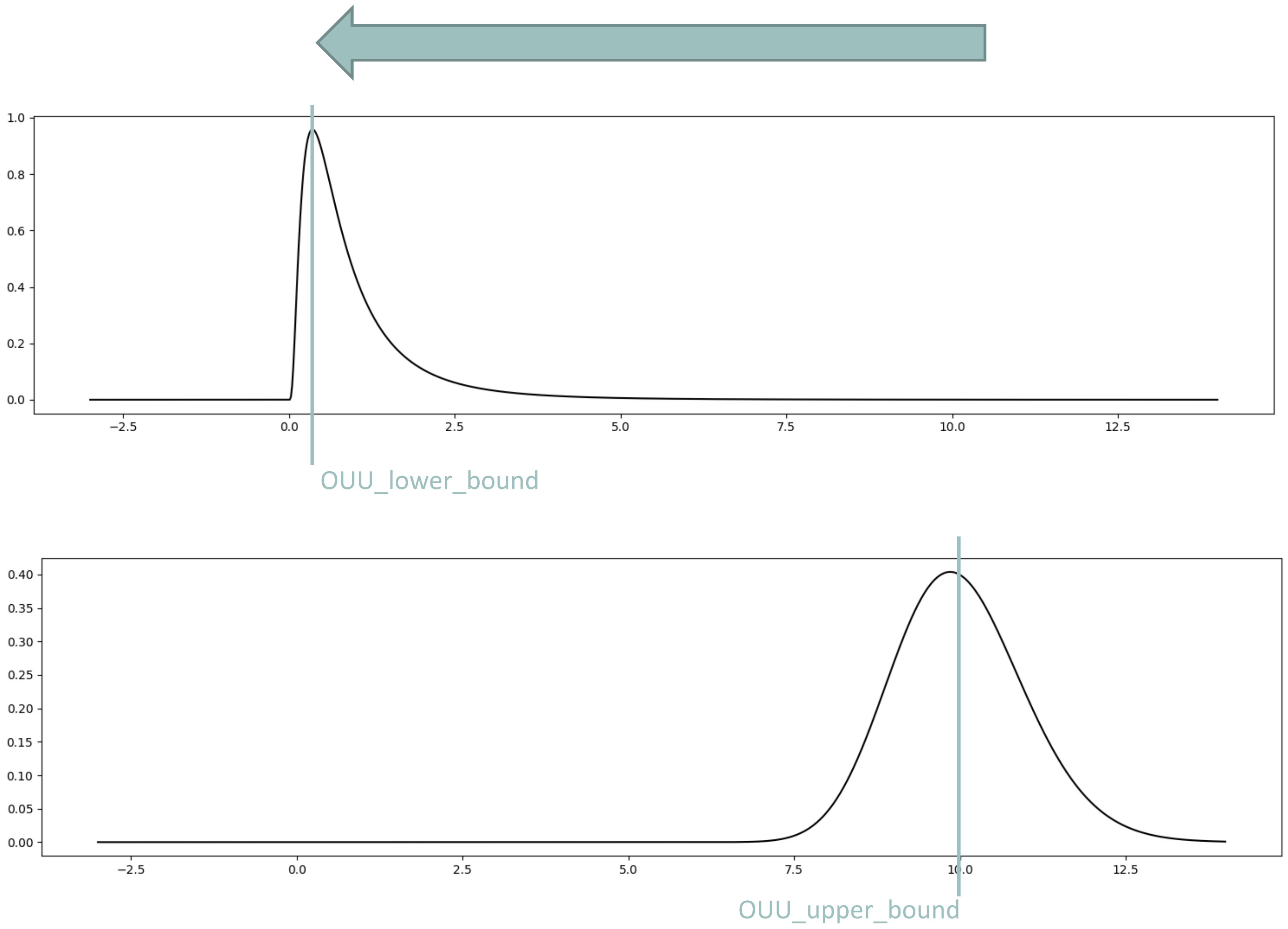


During OUU, the mean of the variables may be varied and optimized.

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*.

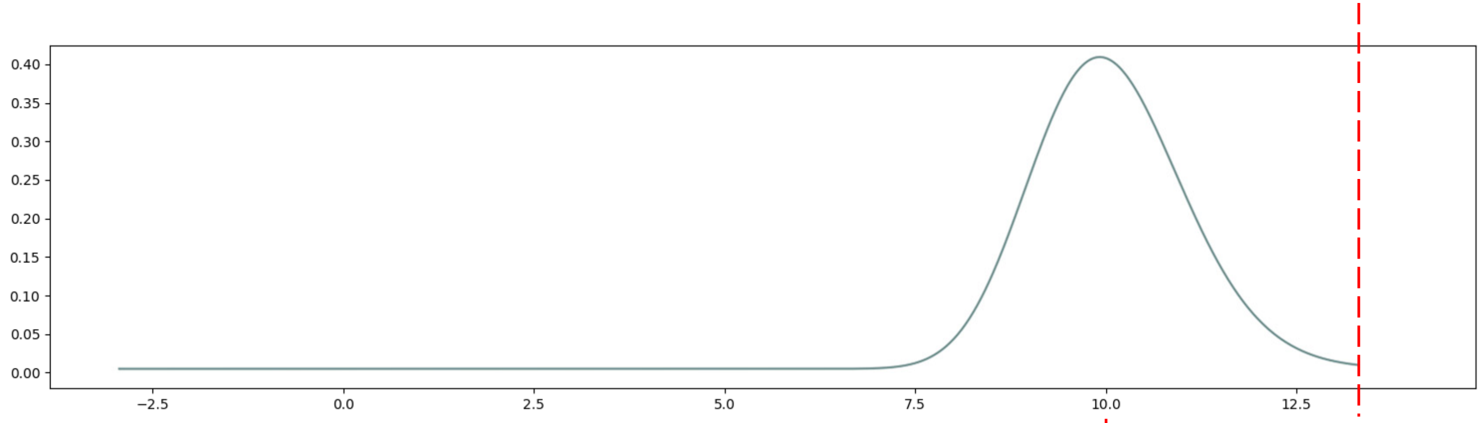
Direction of variable’s mean during the optimization.



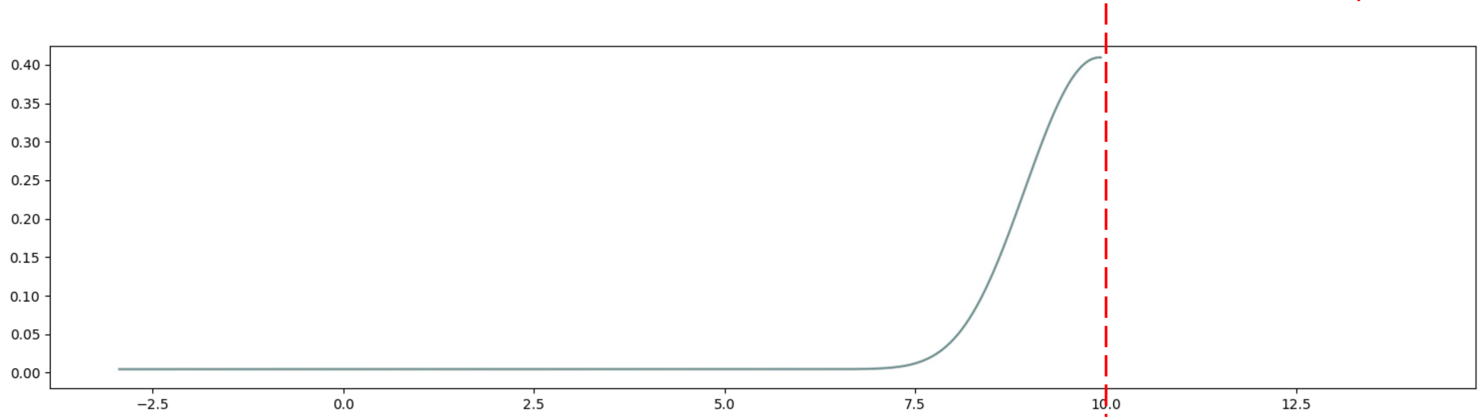
Suppose the OUU variable's initial value is at the upper bound of the OUU variable, which is 10.0.

Three different UQ upper bounds are displayed.

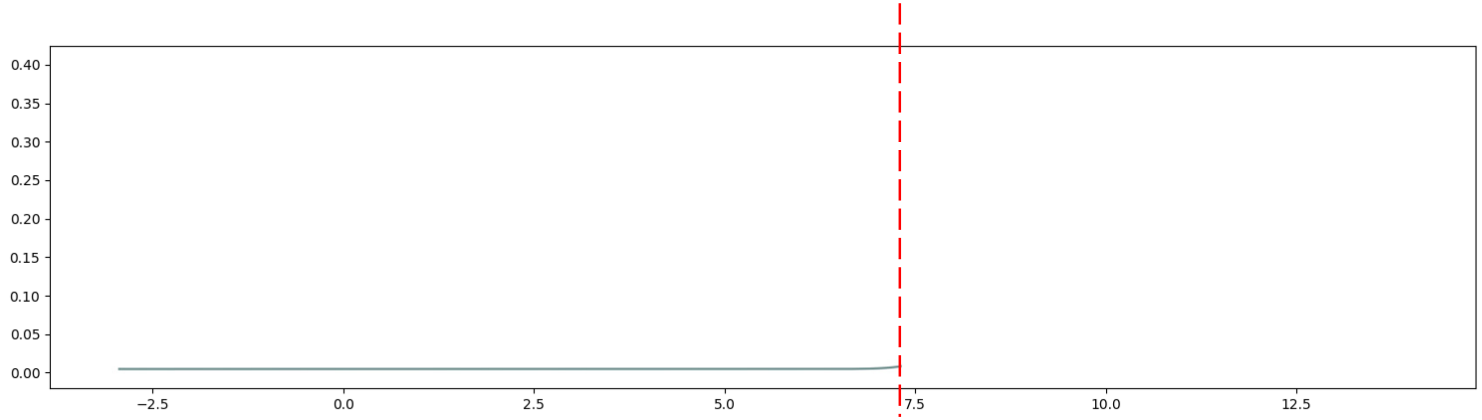
If the UQ or OUU upper bounds are not properly configured, there will be a nearly 0% probability of drawing a sample from the distribution. This 0% probability causes the error.



P(xi < UQ_upper_bound)
= 99.9%



P(xi < UQ_upper_bound)
= 50%



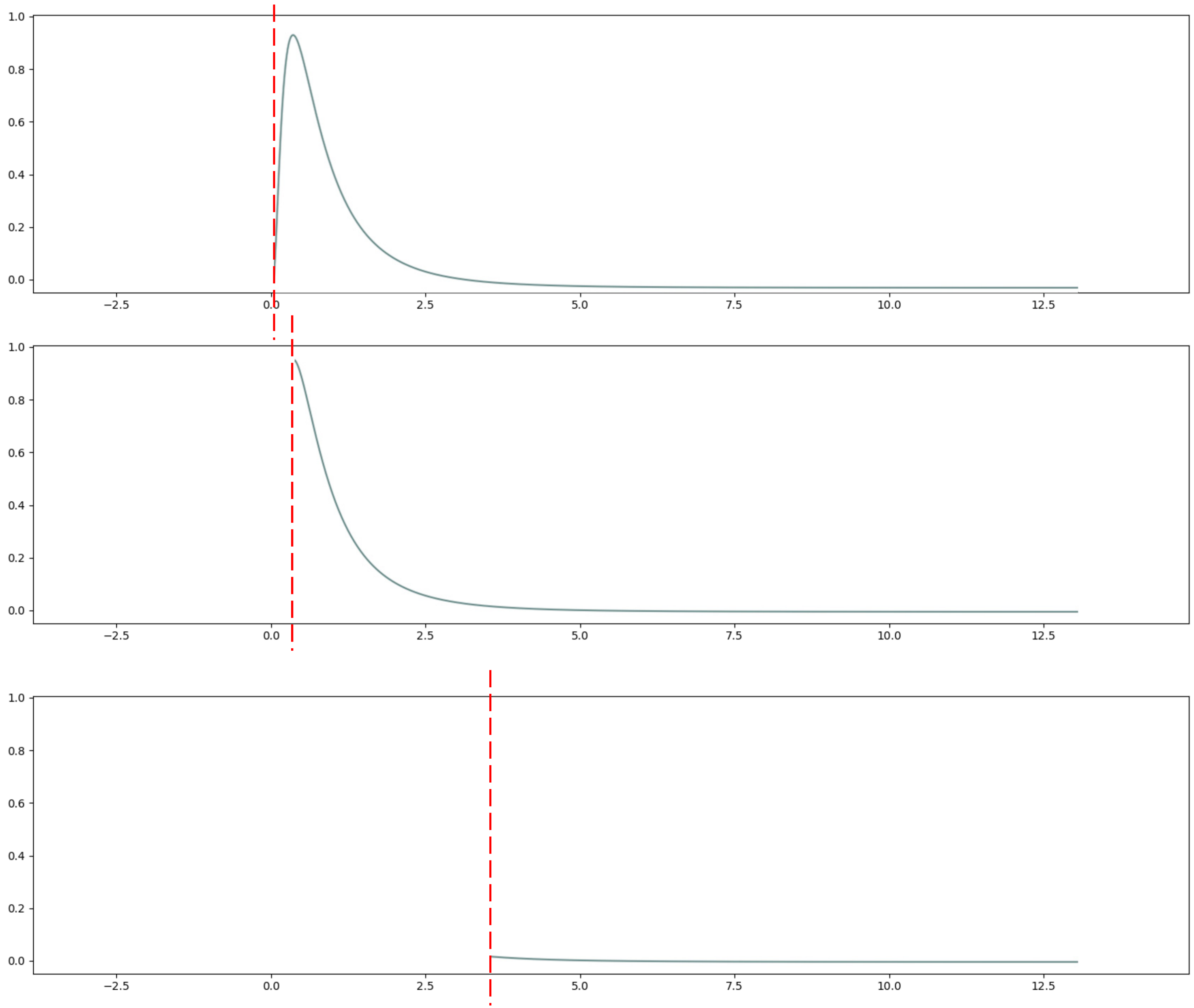
P(xi < UQ_upper_bound)
= .1%

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
```

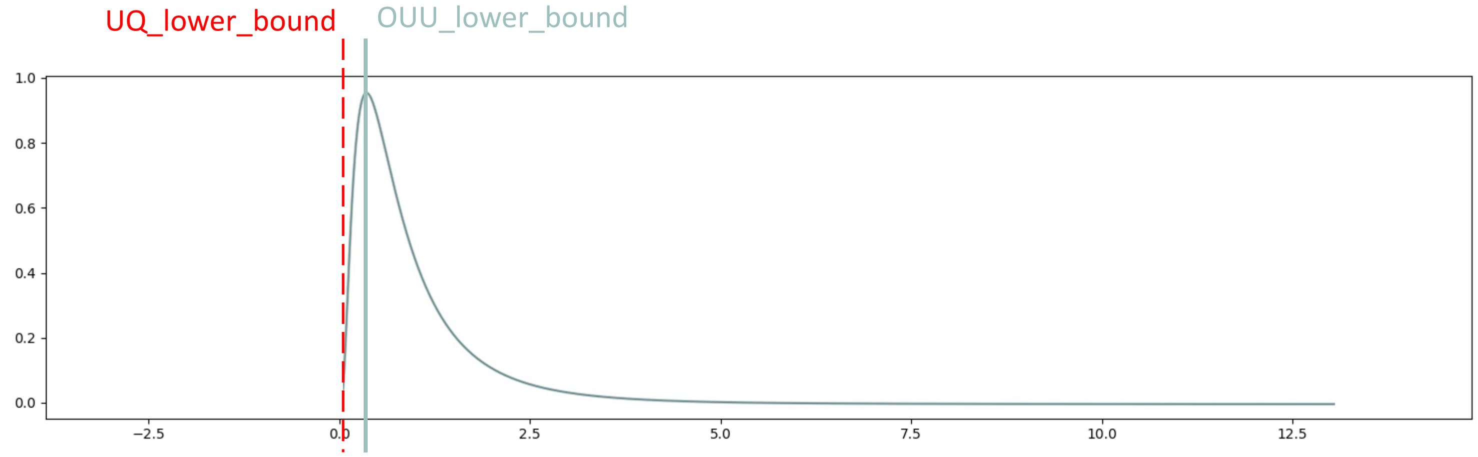
For first time users, the best practice is to ensure the following

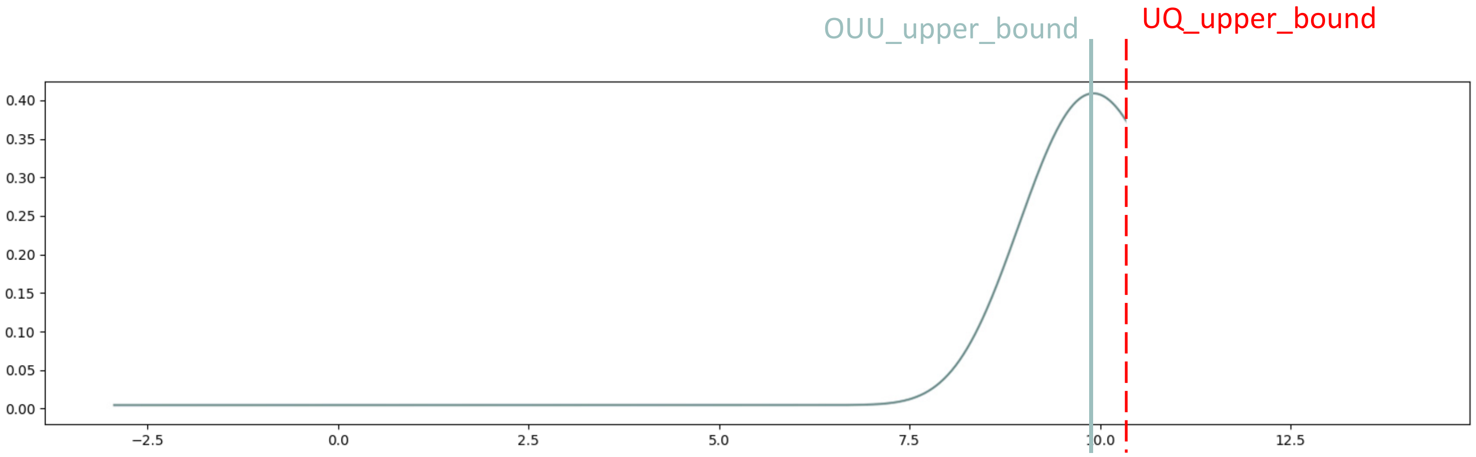
UQ_lower_bound <
OUU_lower_bound

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