Workshop – Optimization Under Uncertainty - 53 Bar Truss

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 = 0.01
  - x2 = 0.01
  - …
  - x53 = 0.01
- Objective:
  - 15.87573E+02
- Max probability of failure:
  - ~0.00% (Actual after UQ with 80 run LHS)

Optimization for Deterministic Responses (MSC Nastran SOL 200)

Optimal Solution

- Variables
  - x1 = 8.0647E-04
  - x2 = 1.2097E-03
  - …
  - x53 = 5.7026E-04
- Objective:
  - 1.156488E+02
- 7 MSC Nastran Runs
- Max probability of failure:
  - 66% (Actual probability after UQ with LHS of size 200 (200 MSC Nastran runs))

Optimization for Stochastic Responses (Sandia Dakota OUU)

Optimal Solution

- Variables
  - x1_mean = 1.3179002125e-03
  - x2_mean = 1.7973456563e-03
  - [...]
  - x53_mean = 1.0674418472e-03
- Objective:
  - 1.8957040000e+02
- 282 MSC Nastran Runs
- Max probability of failure:
  - 5% (Approximated probability after final OUU iteration)
  - 5% (Actual probability after UQ with LHS of size 200 (200 MSC Nastran runs))

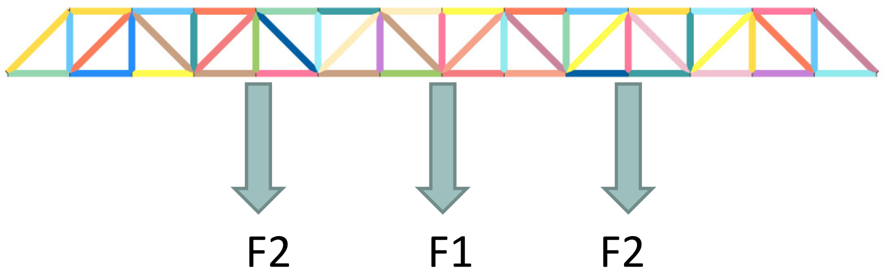
Uncertainty Quantification Problem Statement

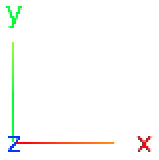
Design Variables

x1: Area (A) of PROD 1
x2: A of PROD 2
x3: A of PROD 3
[...]
x51: A of PROD 51
x52: A of PROD 52
x53: A of PROD 53

Variable	Mean	Standard Deviation	Distribution
xi	0.01	0.0005	Lognormal

Property Name	Property ID	Color of Detail
PROD	1	
PROD	2	
PROD	3	
PROD	51	
PROD	52	
PROD	53	





F1: Force in load case 1
F2: Force in load case 2

Responses

r1: Mass
r2: Axial stress in element 1 (PROD 1), subcase 1
r3: Axial stress in element 2 (PROD 2), subcase 1
[...]
r54: Axial stress in element 53 (PROD 53), subcase 1
r55: Axial stress in element 1 (PROD 1), subcase 2
r56: Axial stress in element 2 (PROD 2), subcase 2
[...]
r107: Axial stress in element 53 (PROD 53), subcase 2

Quantities of interest

r1: Mean and standard deviation (2 quantities)

r2, r3, r4, ..., r107: Mean, standard deviation, and probabilities of exceeding the bounds of -310.E6 < r2 < 310.E6

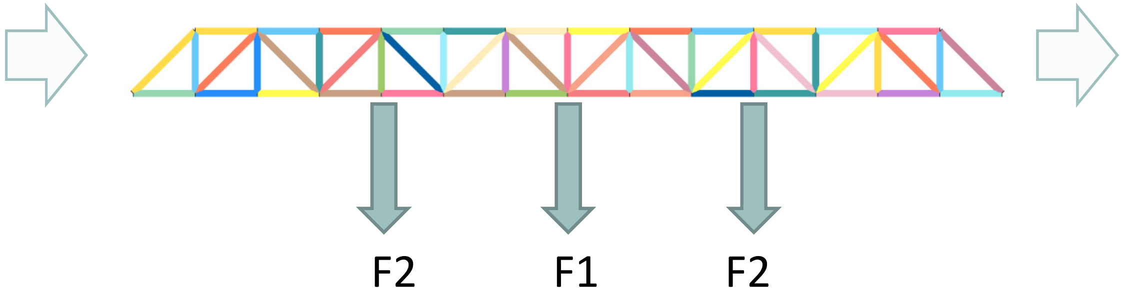
Optimization Under Uncertainty (OUU) Problem Statement

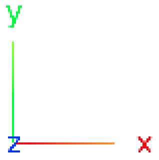
Design Variables

x1_mean: Mean of x1 (A of PROD 1)
x2_mean: Mean of x2 (A of PROD 2)
[…]
x53_mean: Mean of x53 (A of PROD 53)

0.00001 < xi_mean < 1.0

Property Name	Property ID	Color of Detail
PROD	1	
PROD	2	
PROD	3	
PROD	51	
PROD	52	
PROD	53	





F1: Force in load case 1
F2: Force in load case 2

Objective

Minimize mean of r1

Design Constraints

Constraints on probability of failure

For i=2, 3, …, 107

~~ri_pl: 1.0 - P(-310.E6 < ri)~~
~~Probability that r2 < -310.E6~~

~~ri_pu: P(310.E6 < ri)~~
~~Probability that 310.E6 < r2~~

ri_pl: 1.0 - P(-248.E6 < ri)
Probability that r2 < -248.E6

ri_pu: P(248.E6 < ri)
Probability that 248.E6 < r2

ri_pl < 0.05 (5% failure)
ri_pu < 0.05 (5% failure)

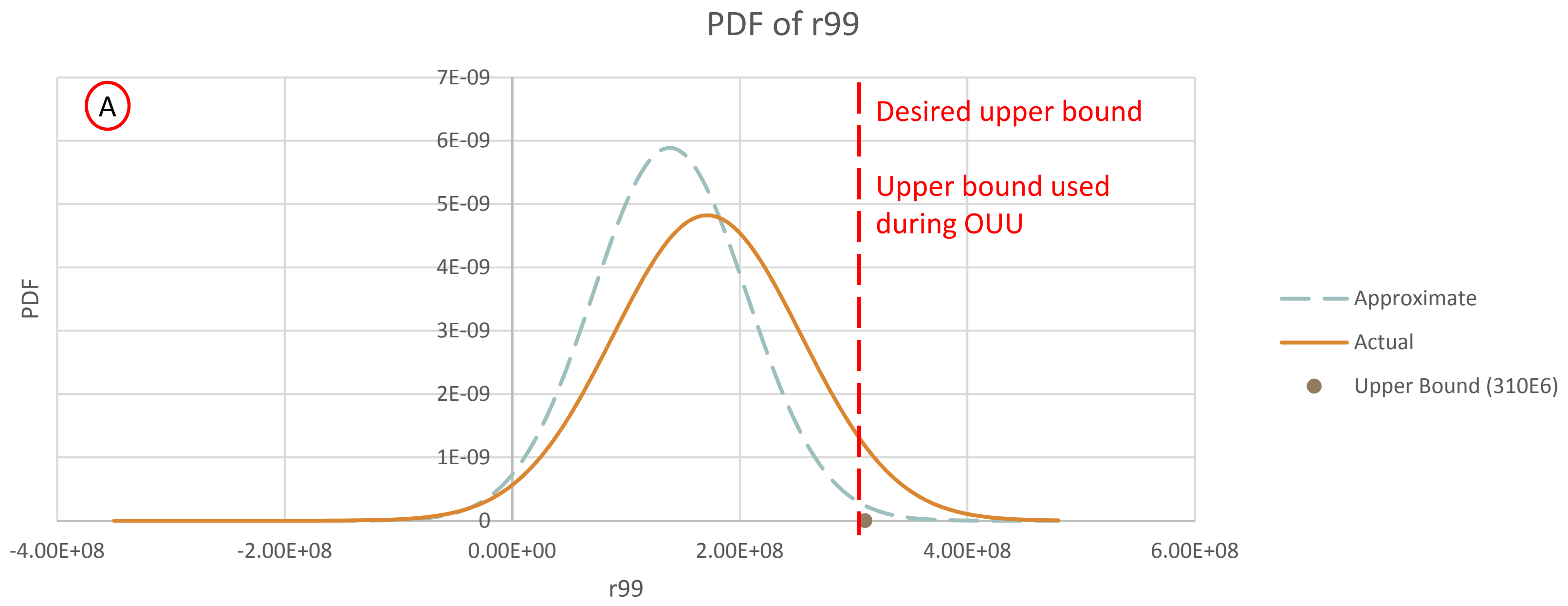
Why a value of 248.E6 instead of 310.E6?

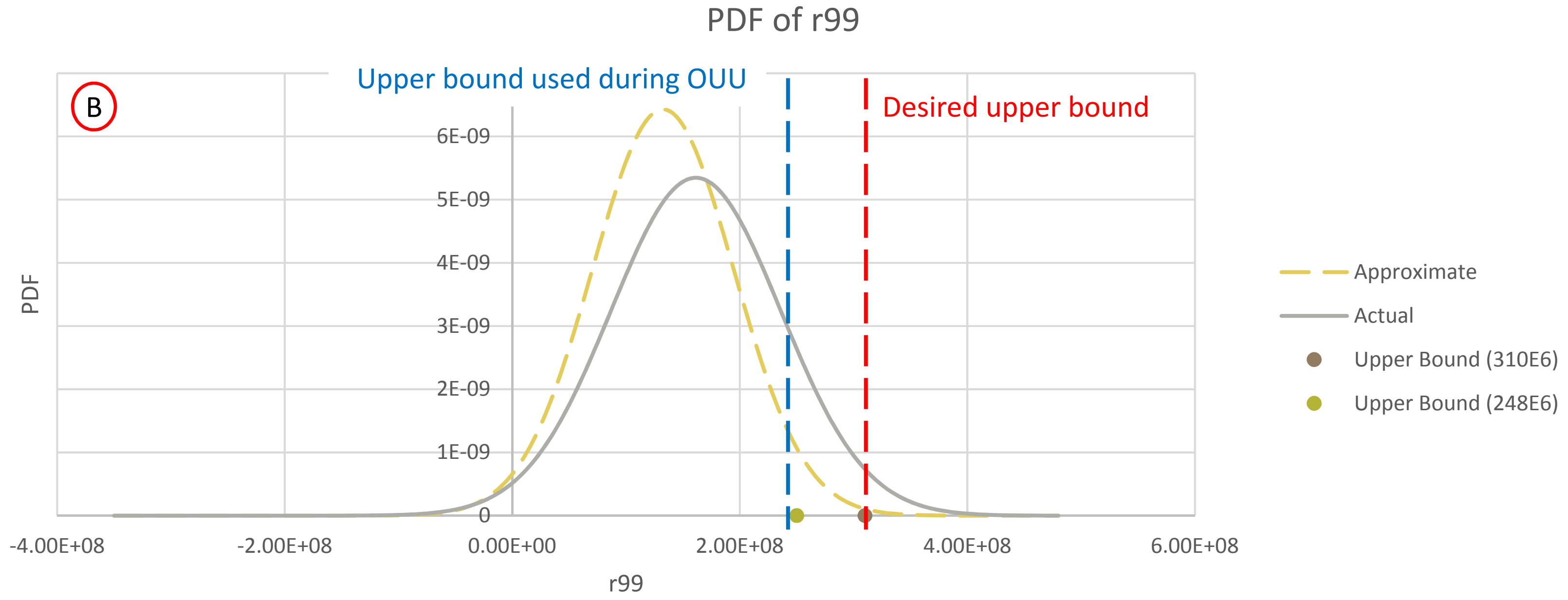
The yield strength of the material in both tension and compression is 310.E6. Instead of constraining the axial stress to 310.E6, the axial stress is constrained to 248.E6 during the OUU. Why?

The goal is to have a probability of failure less than 5% where the upper bound on stress is 310E6.

1. While crafting this tutorial, an attempt was made to perform an OUU where the upper bound was 310E6 and a maximum of 3% probability of failure was imposed. The OUU uses the MVFOSM method for uncertainty quantification, but approximates the statistics. The approximation has an error when compared to the actual statistics, as can be shown in the plot of the approximated and actual probability density functions (PDF). Consequently, the OUU might lead to a feasible solution based on approximated statistics, but upon verification, the actual statistics show the solution is infeasible where some of the probabilities of failure exceed 5%.

2. The OUU was repeated but the upper bound was reduced by 20% to 248E6 and a maximum of 5% probability of failure was imposed. This approach led to a solution that after verification showed the actual probabilities of failure were less than 5%. Why was 20% used? This percentage was determined via trial and error. 15% was shown to yield a good solution, so a conservative value of 20% was ultimately used.





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, …, x53) = (8.0647E-04, 1.2097E-03, …, 5.7026E-04). The original analysis model values were (x1, x2, …, x53) = (0.01, 0.01, …, 0.01).

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 (0.01, 0.01, …, 0.01). In trial 2, the initial values of the OUU variables were equal to the optimal solution values (8.0647E-04, 1.2097E-03, …, 5.7026E-04).

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 in trial 2 is possibly closer 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.

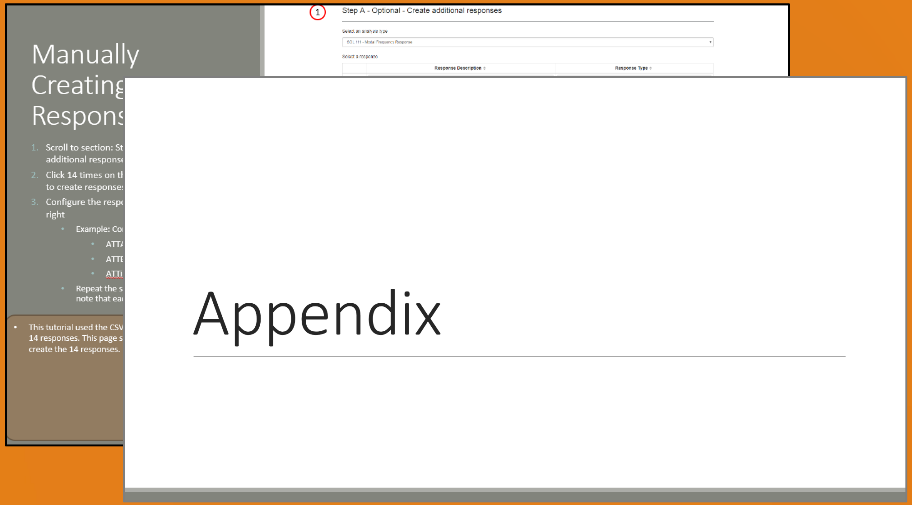
Prior to OUU, it is recommended to perform a local optimization or global optimization to determine ideal initial values for OUU variables.

OUU Trial	Initial Values	Comments	Number of MSC Nastran Runs to Converge During OUU	OUU Solution
Trial 1	x1_mean = 0.01 x2_mean = 0.01 … x53_mean = 0.01	These were the original analysis model values with no prior optimization.	340	x1_mean = 3.3741703868e-03 x2_mean = 3.3741703868e-03 ... x53_mean = 1.0612358999e-03 Objective = 403.702
Trial 2	x1_mean = 8.0647E-04 x2_mean = 1.2097E-03 … x53_mean = 5.7026E-04	The initial values are based on the optimal solution after an MSC Nastran SOL 200 optimization.	282	x1_mean = 1.3179002125e-03 x2_mean = 1.7973456563e-03 ... x53_mean = 1.0674418472e-03 Objective = 189.57

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 For Numerous Uncertain Variables - Uncertainty quantification and optimization under uncertainty will require hundreds or thousands of black box function runs when dozens or hundreds of uncertain variables are considered. Strategies are necessary to minimize the number of black box function runs necessary to reach a converge solution. This exercise discusses such strategies and steps needed to configure an OUU for numerous uncertain variables.

SOL 200 Web App Capabilities

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

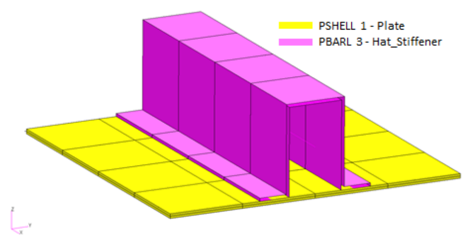
Compatibility

- 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.

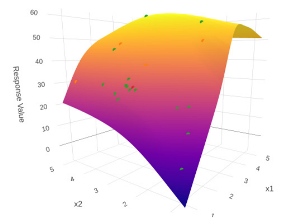
Benefits

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

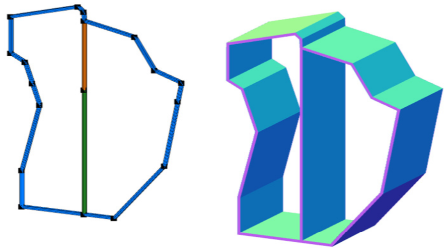
Web Apps



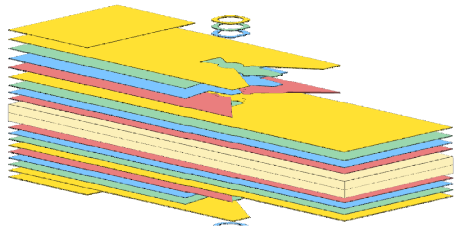
Web Apps for MSC Nastran SOL 200
Pre/post for MSC Nastran SOL 200. Support for size, topology, topometry, topography, multi-model optimization.



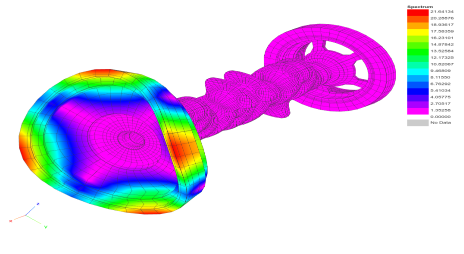
Machine Learning Web App
Bayesian Optimization for nonlinear response optimization (SOL 400)



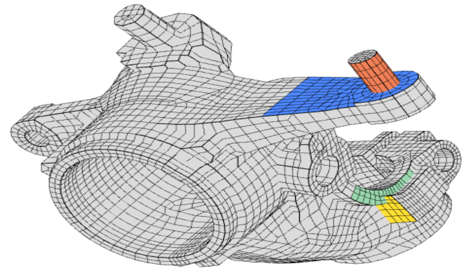
PBMSECT Web App
Generate PBMSECT and PBRSECT entries graphically



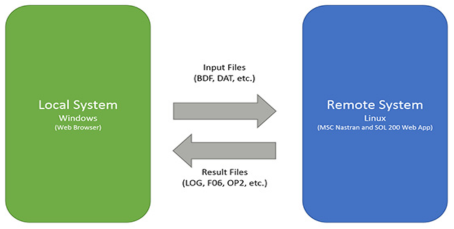
Ply Shape Optimization Web App
Optimize composite ply drop-off locations, and generate new PCOMPG entries



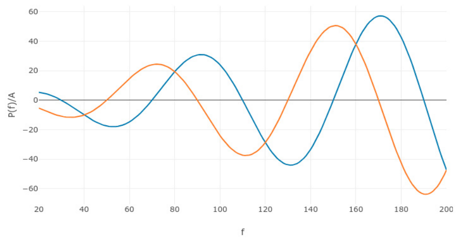
Post-processor Web App
View MSC Nastran results in a web browser on Windows and Linux



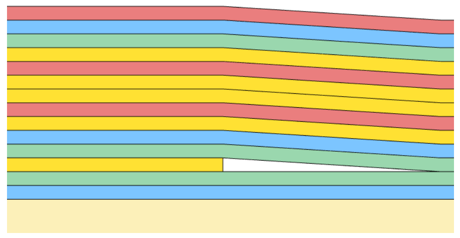
Shape Optimization Web App
Use a web application to configure and perform shape optimization.



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

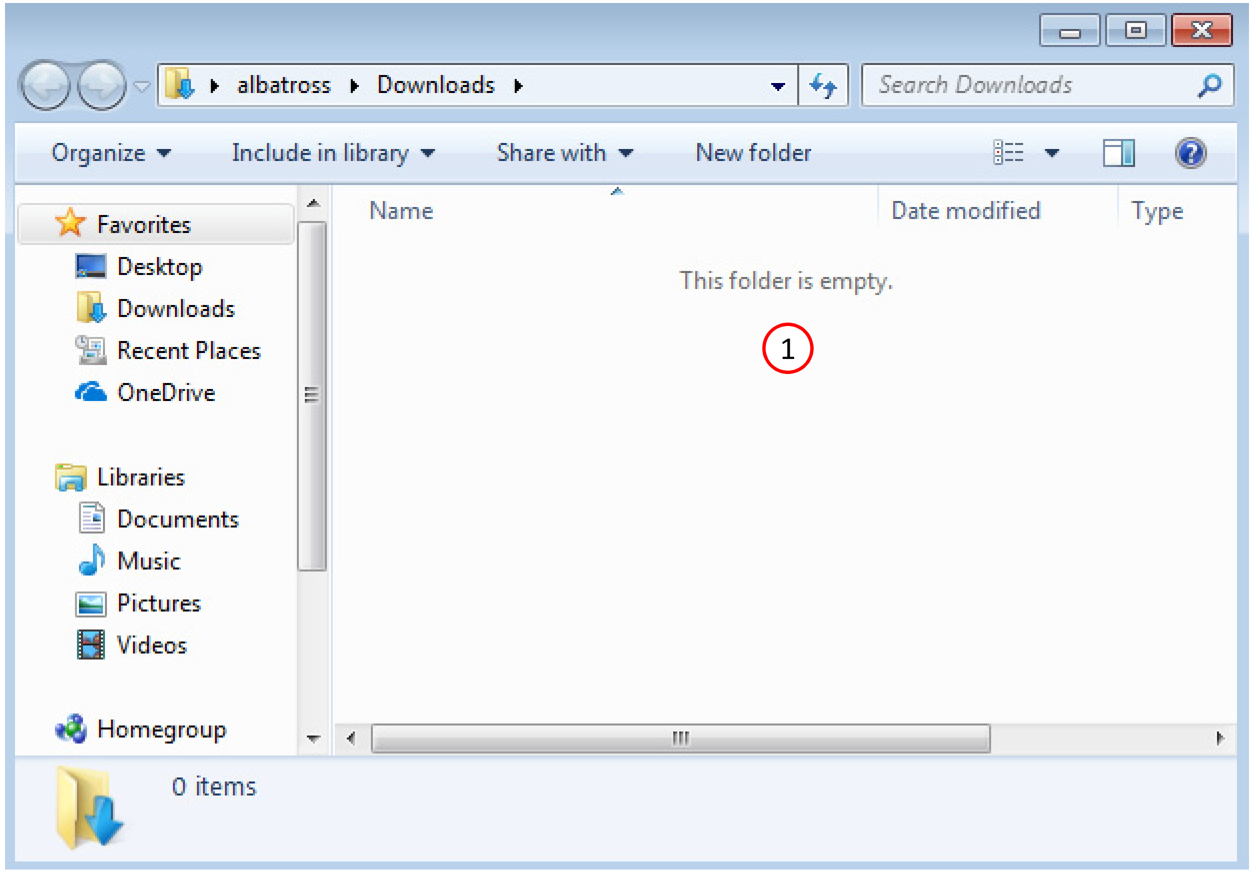


Stacking Sequence Web App
Optimize the stacking sequence of composite laminate plies

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

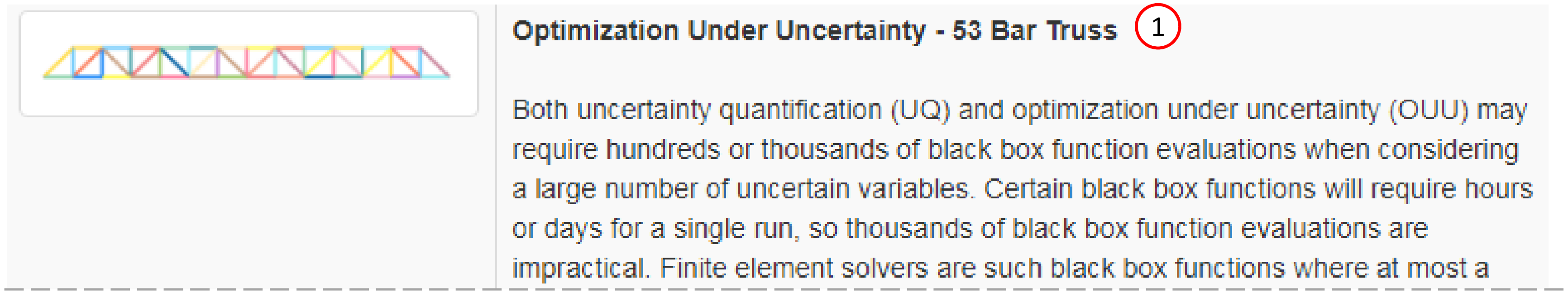
1. Click on the indicated link

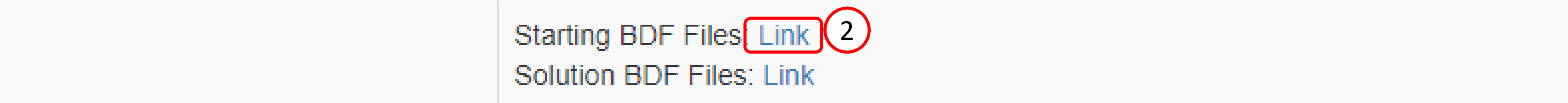
- The necessary BDF files for this tutorial are available in the Tutorials section of the User's Guide.

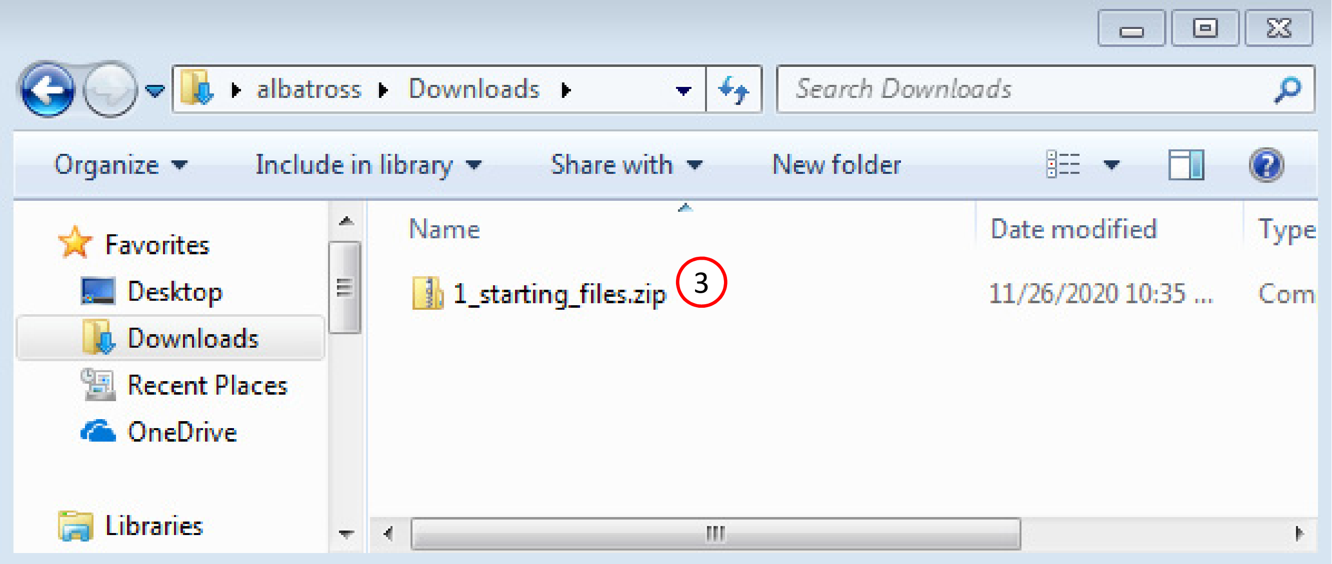
The Engineering Lab

Obtain Starting Files

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

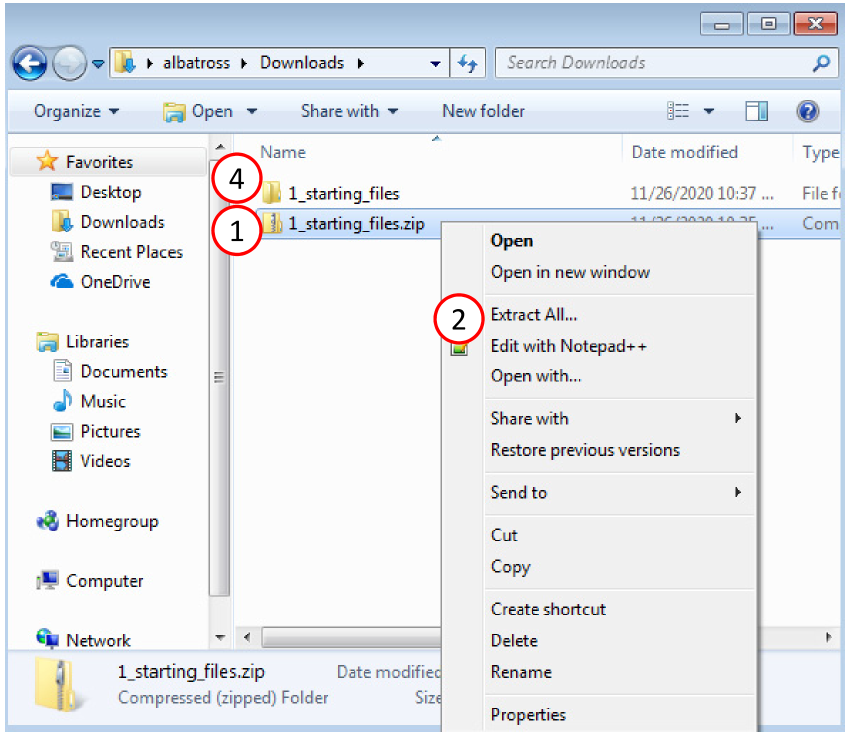


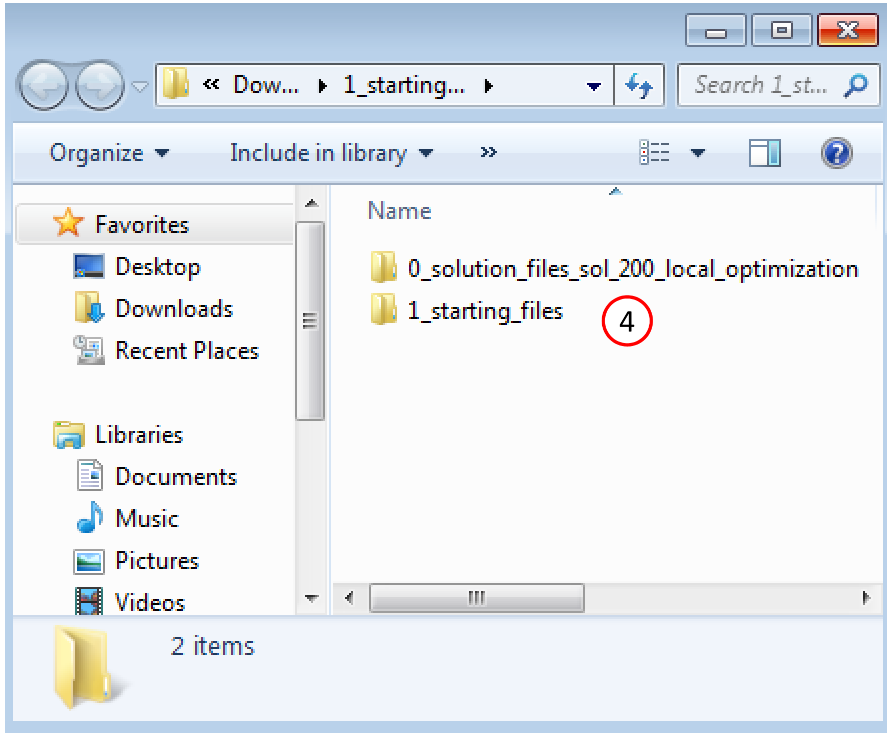


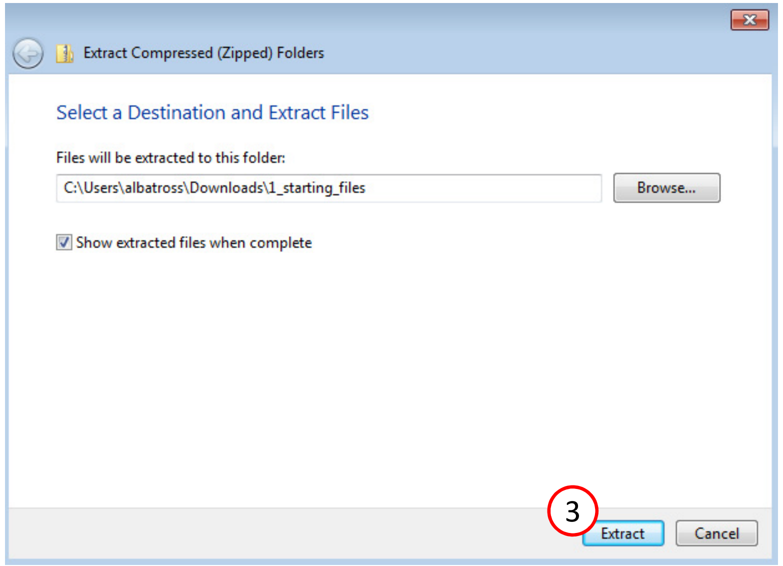


Obtain Starting Files

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





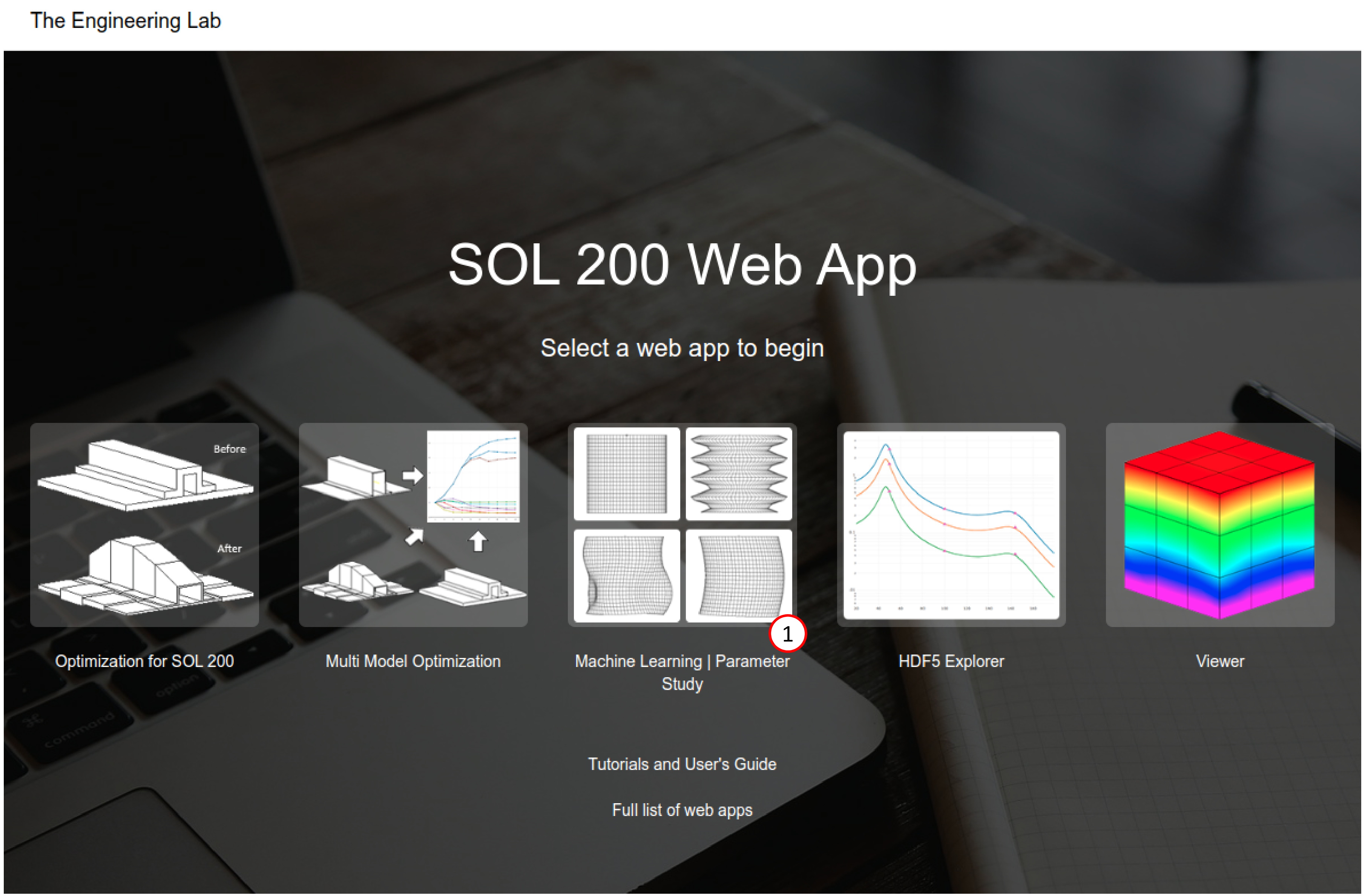


Part A – Optimization Under Certainty

Open the Correct Page

1. Click on the indicated link

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



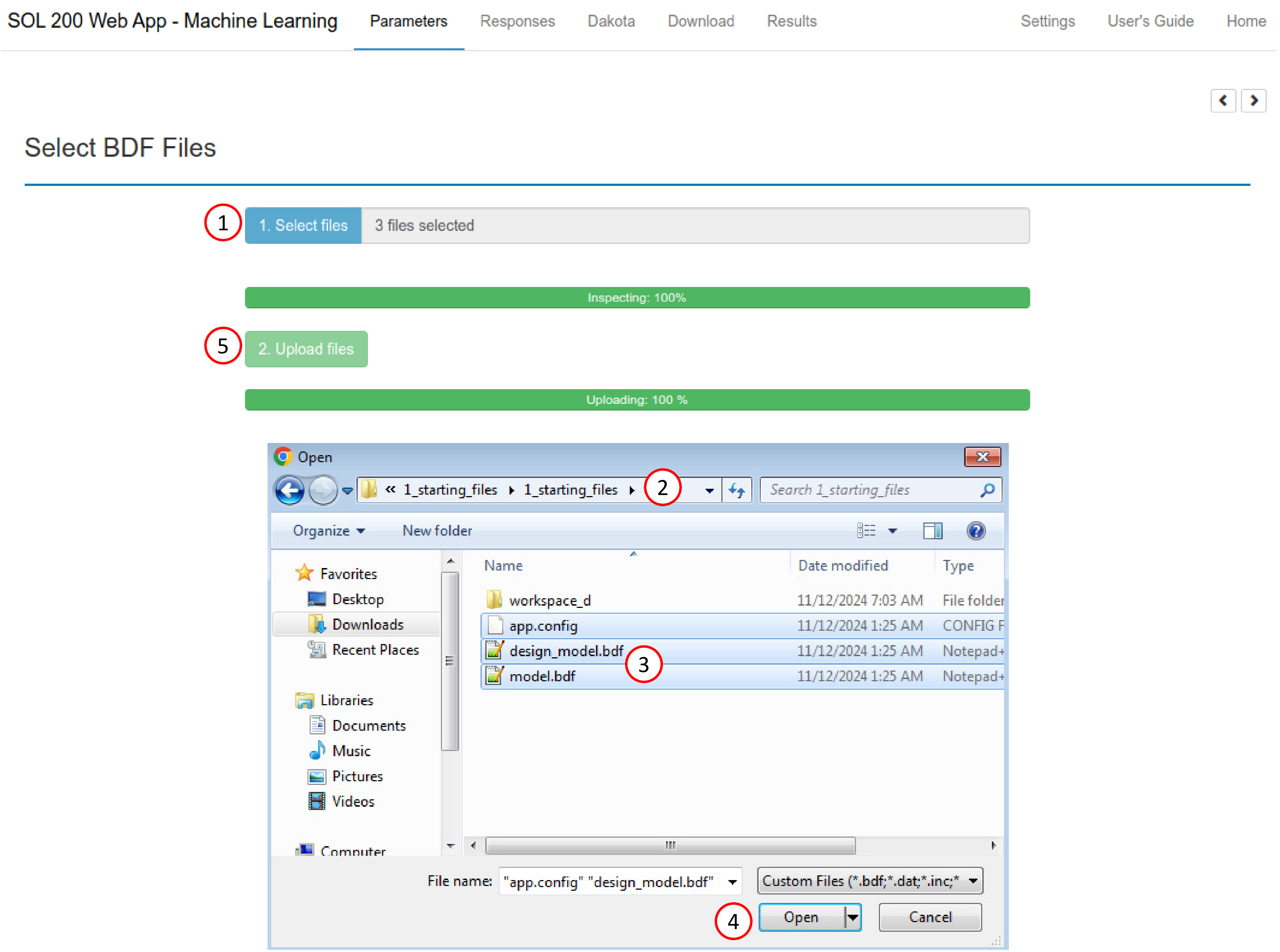
Select BDF Files

1. Click Select files
2. Navigate to this directory

`1_starting_files/1_starting_files`

3. Select the indicated file
4. Click Open
5. 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.

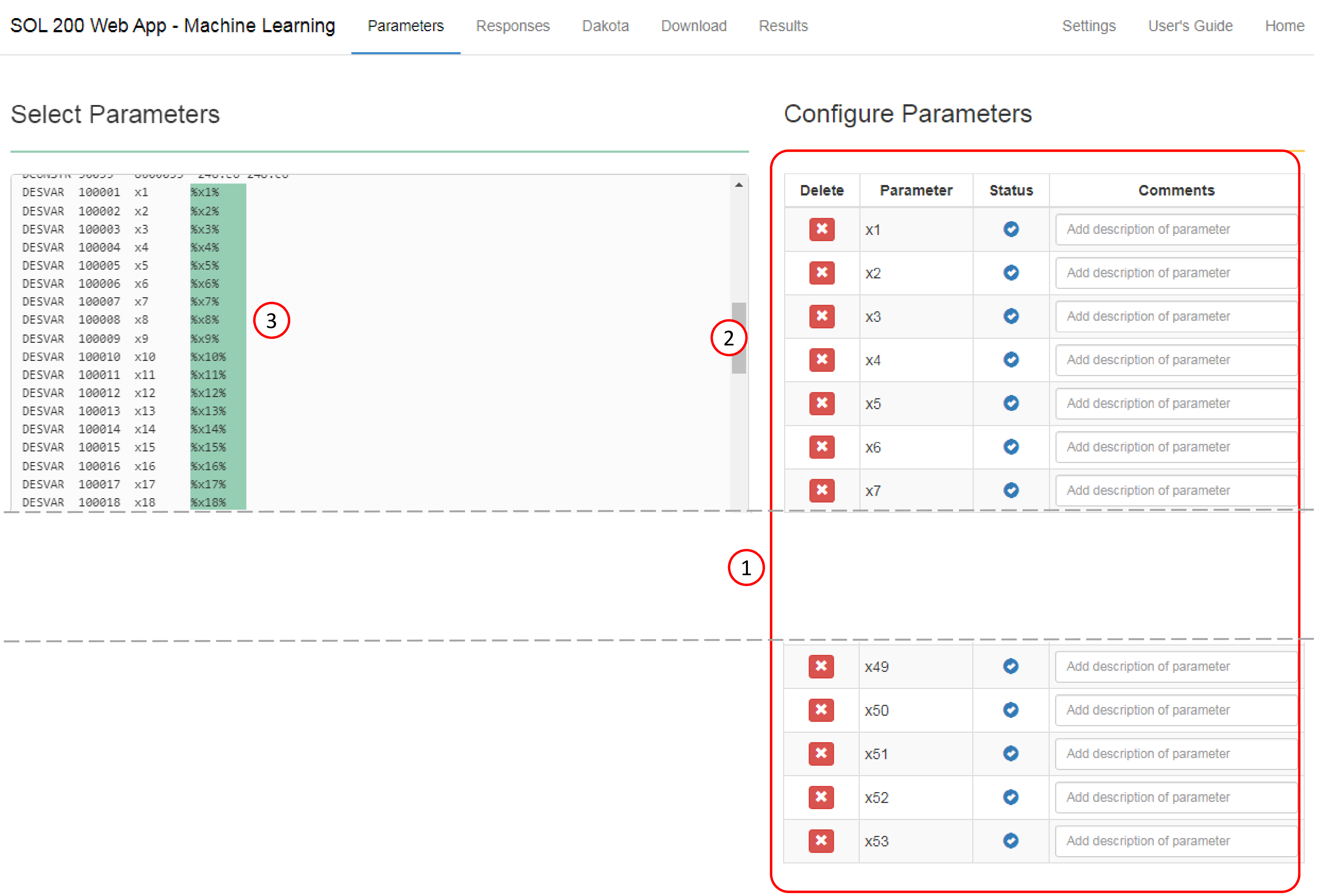


Parameters

No modifications are required in the following steps. Only inspection is required.

1. 53 parameters, or variables, have already been configured
2. Use the vertical scroll bar to scroll to the DESVAR entries
3. The variables correspond to the initial value of DESVAR entries. Each DESVAR entry corresponds to a cross section area of a member of the truss. The OUU will vary the initial value of the DESVAR entry, which will vary the cross section area of each member of the truss.

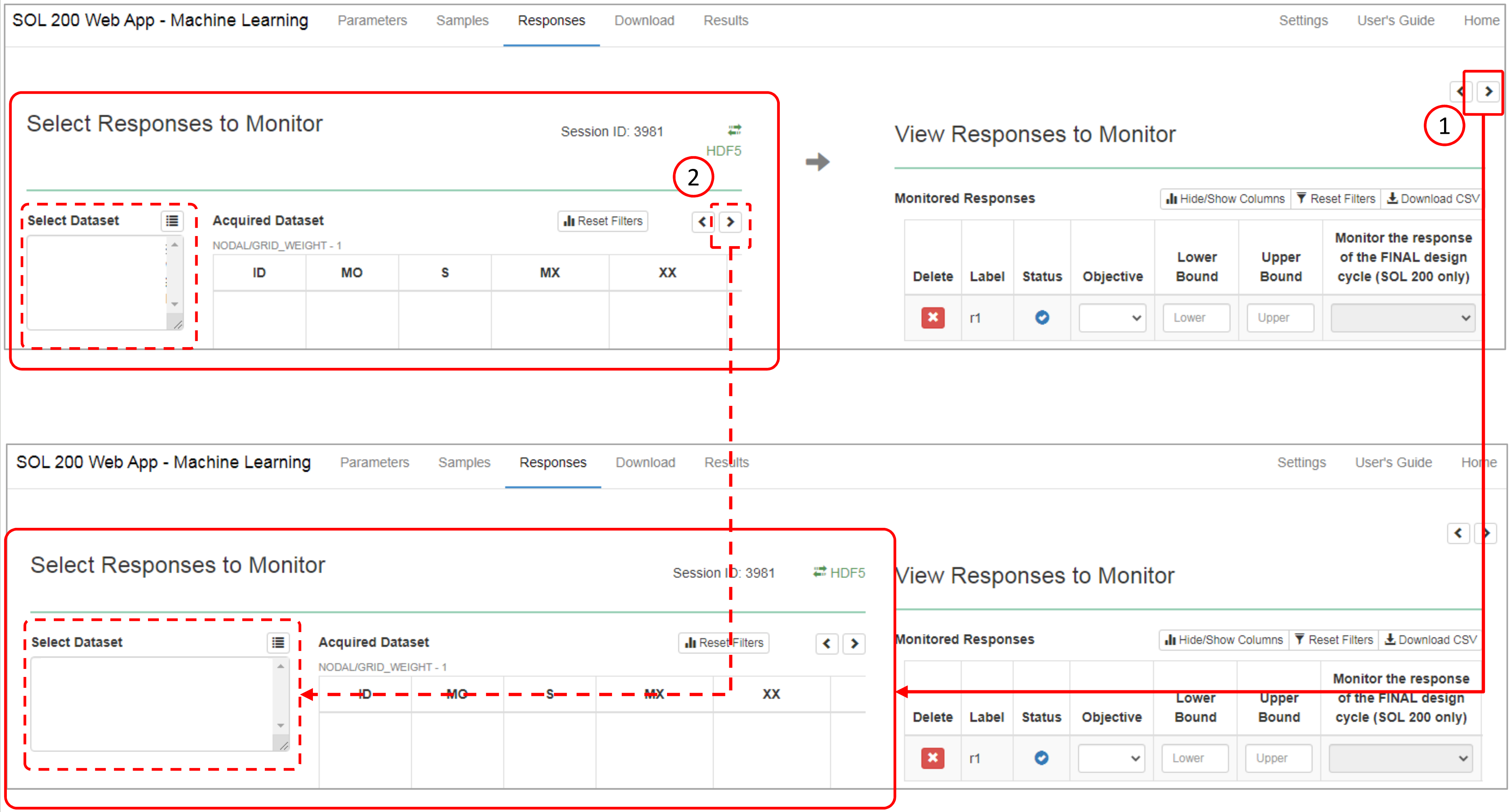
- If gradients are expected to be provided to Dakota, select only the initial values of DESVAR entries.
    - When the initial values of DESVASR entries are selected, only the independent DESVAR entries should be selected.
- If gradients are not expected, any other field with real values may be selected.



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.

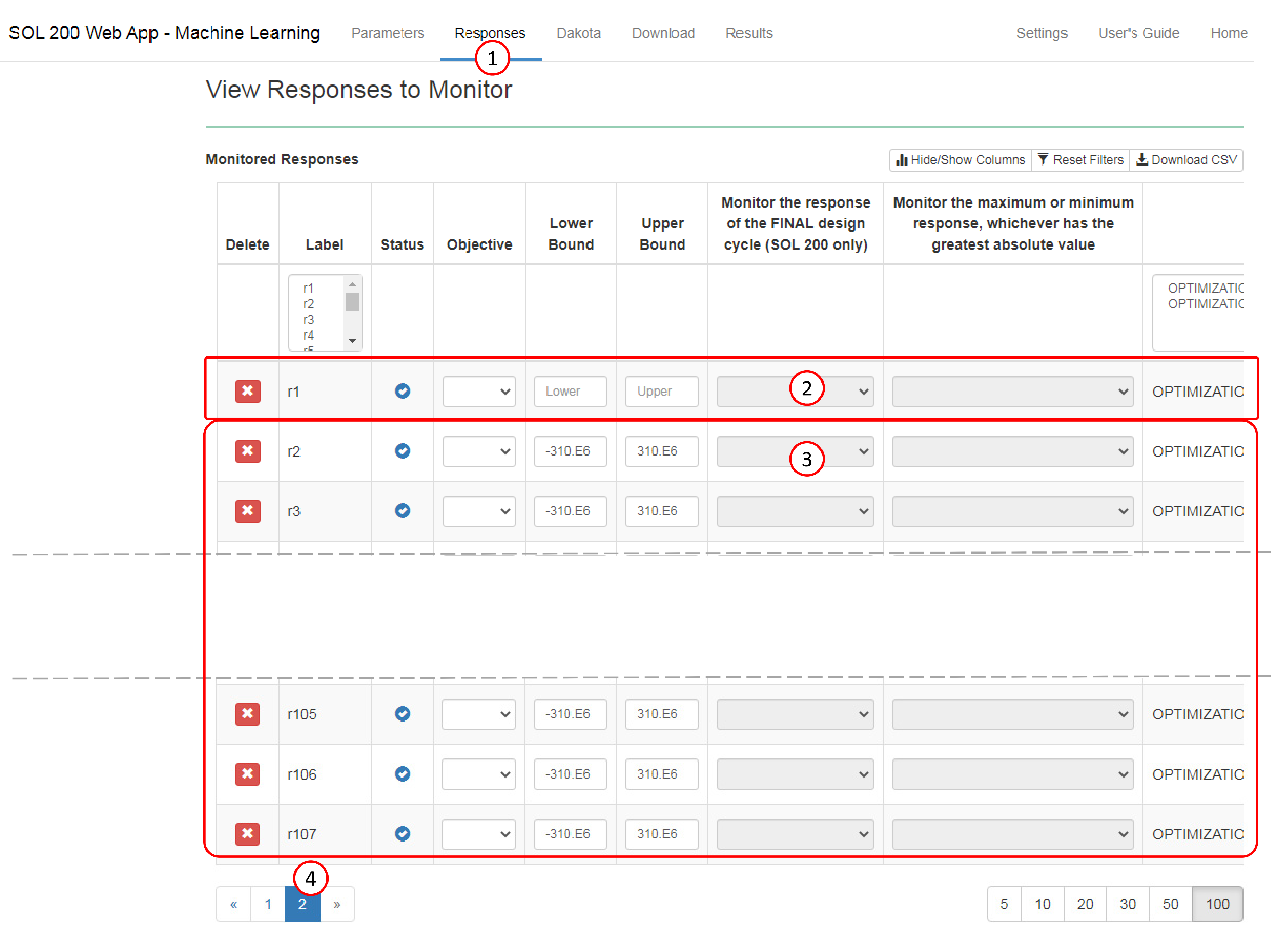


Responses

1. Click Responses

Responses have already been configured.

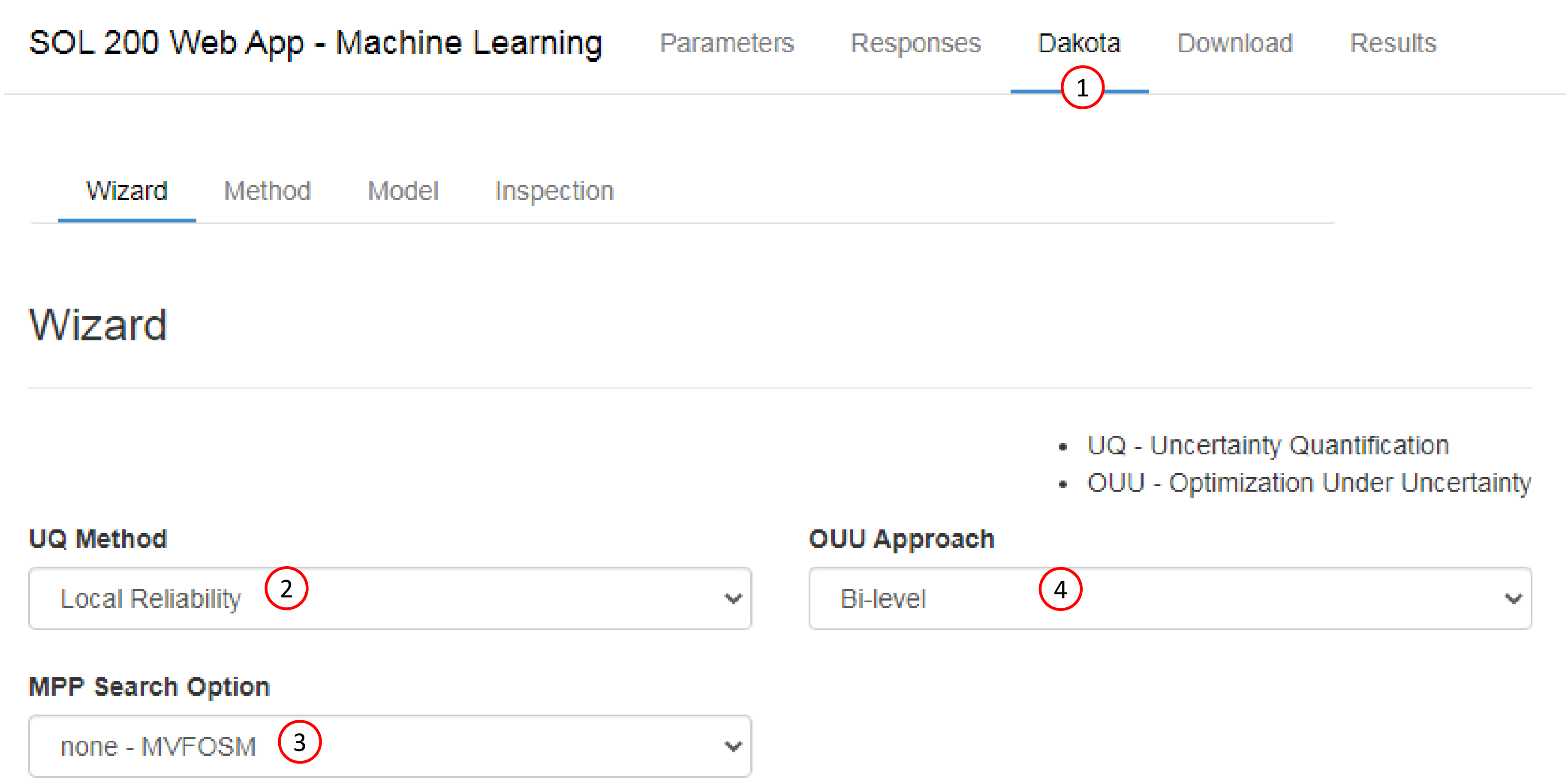
2. Response r1 corresponds to the weight
3. Responses r2 through r107 correspond to the axial stress of each member for subcases 1 and 2
4. Click page 2 to view the additional responses



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 was performed and revealed the responses were normally distributed. MSC Nastran SOL 200 can also output gradients/sensitivities for the responses. Since gradients are available, reliability methods may be used during the OUU. Also, since the standard deviations of the UQ variables are small, the response functions are locally smooth and nearly linear. The Mean Value First-Order Second-Moment Method (MVFOSM) yields an acceptable level of approximating the tail probabilities.



Dakota - Uncertainty Quantification (UQ)

1. Scroll to section Uncertainty Quantification
2. The distribution, mean and standard deviation have already been configured for each variable. No modifications are necessary.

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

Uncertainty Quantification 1

Configure UQ Variables

Reset Table

Delete	Descriptor	Status	Distribution	Mean	Standard Deviation	Initial Value	Lower Bound	Upper Bound	Description
	Search		Search	Search	Search	Search	Search	Search	Search
			Lognormal	Batch	Batch	Batch	Batch	Batch	
	x1		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x2		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x3		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x4		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x5		Lognormal Unc	.01	0.0005	2	UQ Lower Boun	UQ Upper Boun	
	x6		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x7		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x8		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x9		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	
	x10		Lognormal Unc	.01	0.0005		UQ Lower Boun	UQ Upper Boun	

« 1 2 3 4 5 6 »

10 20 30 40 50

Dakota - Uncertainty Quantification (UQ)

1. Navigate to section Select OUU Variables
2. Set the number of visible rows to 50
3. Click On
4. All the visible variable’s means have been automatically be set as OUU variables

Optimization Under Uncertainty

Select OUU Variables 1

Reset Table

Descriptor	Initial Value	Mean	Description
Search	Search	Search	Search
	Off On	Off On 3	
x1		+ Mean 4	
x2		+ Mean	
x3		+ Mean	
x4		+ Mean	
x5		+ Mean	
x6		+ Mean	
x7		+ Mean	
x8		+ Mean	
x9		+ Mean	
x10		+ Mean	

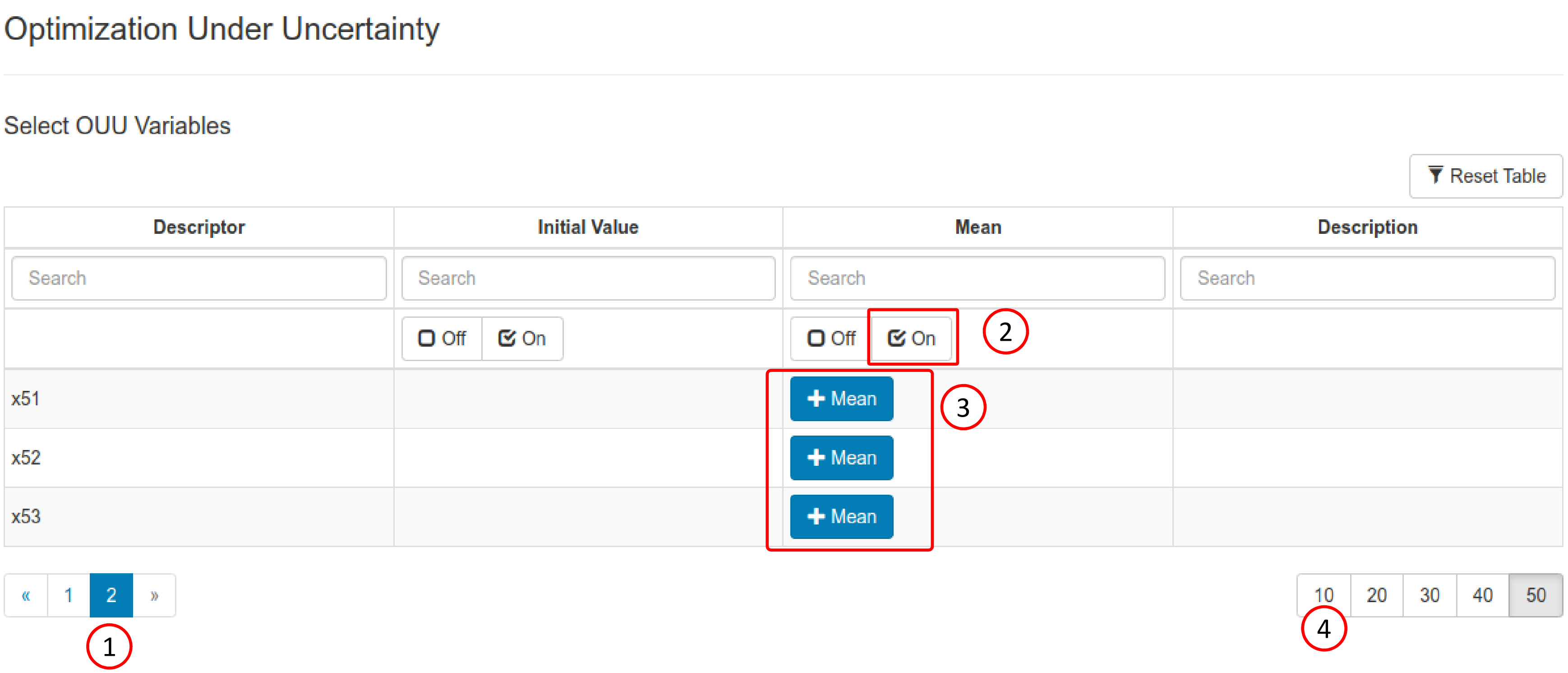
« 1 2 »

10 20 30 40 50

2

Dakota - Uncertainty Quantification (UQ)

1. Click page 2 to view the next set of rows of the table
2. Click On
3. All the visible variable's means have been automatically be set as OUU variables
4. Click the indicated button to view at most 10 rows in the table. Reducing the number of visible rows helps improve the performance of the web app.



Dakota - Optimization Under Uncertainty (OUU)

1. Scroll to section Configure OUU Variables
2. Set the number of visible rows to 50
3. Set the indicated input box to .00001
4. Click the indicated button
5. A lower bound of .00001 has been defined for each visible variable
6. Set the indicated input box to 1.0
7. Click the indicated button
8. An upper bound of 1.0 has been defined for each visible variable

Configure OUU Variables 1

Reset Table

Delete	Descriptor	Status	Initial Value	Lower Bound	Upper Bound	Description
	Search		Search	3 arch 4	6 arch 7	Search
			Batch ↓	.00001 ↓	1.0 ↓	
✖	x1_mean	✔	.01	.00001	1.0	Mean -
✖	x2_mean	✔	.01	.00001	1.0	Mean -
✖	x3_mean	✔	.01	.00001	1.0	Mean -
✖	x4_mean	✔	.01	.00001	1.0	Mean -
✖	x5_mean	✔	.01	.00001 5	1.0 8	Mean -
✖	x6_mean	✔	.01	.00001	1.0	Mean -
✖	x7_mean	✔	.01	.00001	1.0	Mean -
✖	x8_mean	✔	.01	.00001	1.0	Mean -
✖	x9_mean	✔	.01	.00001	1.0	Mean -
✖	x10_mean	✔	.01	.00001	1.0	Mean -

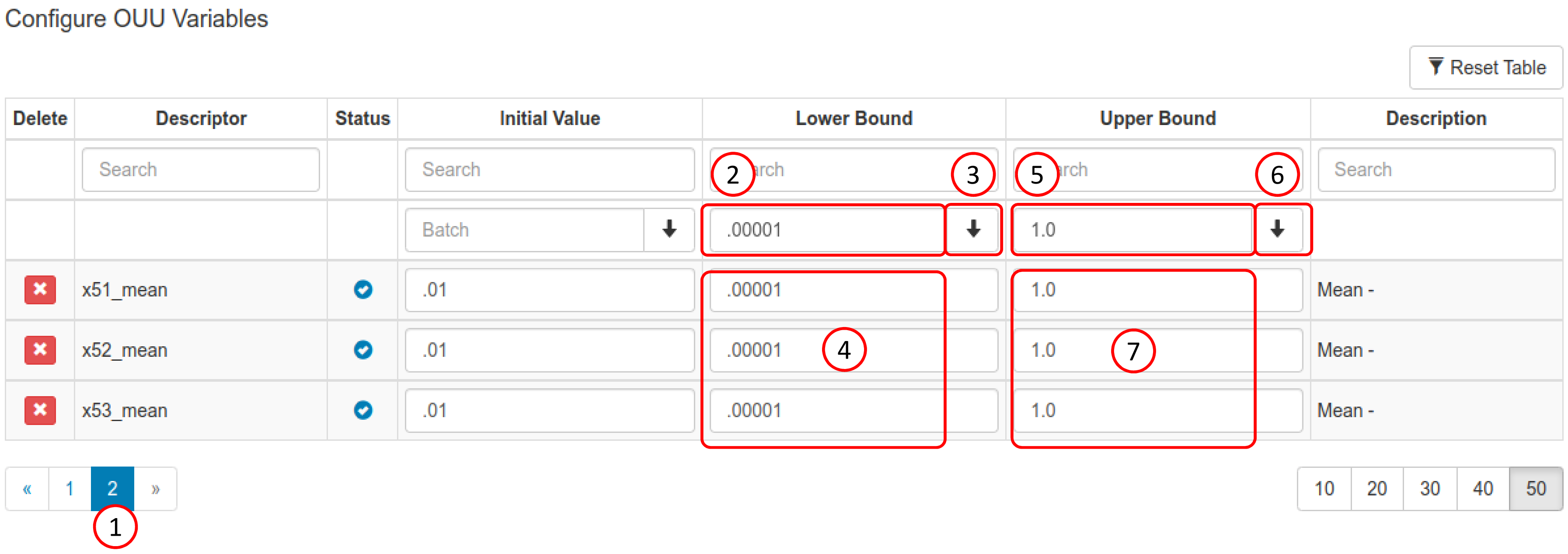
« 1 2 »

10 20 30 40 50

2

Dakota - Optimization Under Uncertainty (OUU)

1. Click page 2 to view the next set of rows of the table
2. Set the indicated input box to .00001
3. Click the indicated button
4. A lower bound of .00001 has been defined for each visible variable
5. Set the indicated input box to 1.0
6. Click the indicated button
7. An upper bound of 1.0 has been defined for each visible variable



Dakota - Optimization Under Uncertainty (OUU)

A separate optimization was performed with MSC Nastran SOL 200 and considered responses that were deterministic. The solution of that optimization is used as the initial starting point for the OUU.

1. Set the number of visible rows to 10
2. Click page 1 to view the first set of rows of the table
3. Specify the following initial values for variables x1_mean through x4_mean
   - 8.0647E-04
   - 1.2097E-03
   - 1.6129E-03
   - 1.6129E-03
4. It will be too time consuming to update each initial value. For now, only 4 variables are updated and the initial values for the other variables are left as is. Later in this exercise, all the initial values will be updated using a faster procedure.

Configure OUU Variables

Delete	Descriptor	Status	Initial Value			
	Search		Search			
			Batch			
	x1_mean		8.0647E-04			
	x2_mean		1.2097E-03			
	x3_mean		1.6129E-03			
	x4_mean		1.6129E-03			
	x5_mean		.01			
	x6_mean		.01	.00001	1.0	Mean -
	x7_mean		.01	.00001	1.0	Mean -
	x8_mean		.01	.00001	1.0	Mean -
	x9_mean		.01	.00001	1.0	Mean -
	x10_mean		.01	.00001	1.0	Mean -

« 1 2 3 4 5 6 »

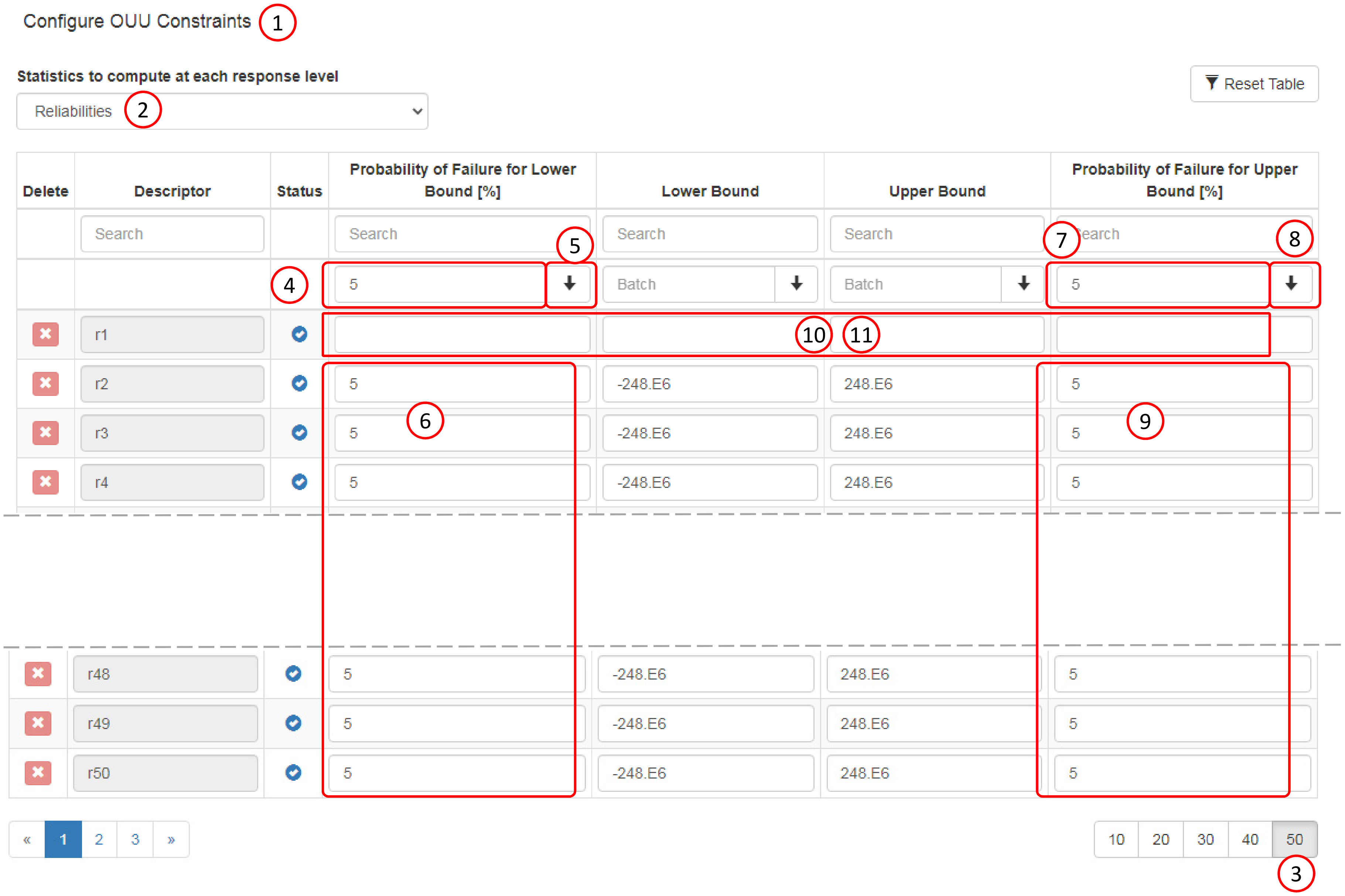
10 20 30 40 50

File:
./1_starting_files/0_solution_files_sol_200_local_optimization/model.f06

INTERNAL DV. ID.	EXTERNAL DV. ID.	LABEL	6	7
1	100001	X1	8.0647E-04 :	
2	100002	X2	1.2097E-03 :	
3	100003	X3	1.6129E-03 :	
4	100004	X4	1.6129E-03 :	
5	100005	X5	1.6129E-03 :	
6	100006	X6	1.6129E-03 :	
7	100007	X7	1.6129E-03 :	
8	100008	X8	1.2097E-03 :	
9	100009	X9	8.0647E-04 :	
10	100010	X10	4.0324E-04 :	
11	100011	X11	5.7026E-04 :	
12	100012	X12	4.0324E-04 :	
13	100013	X13	8.0647E-04 :	

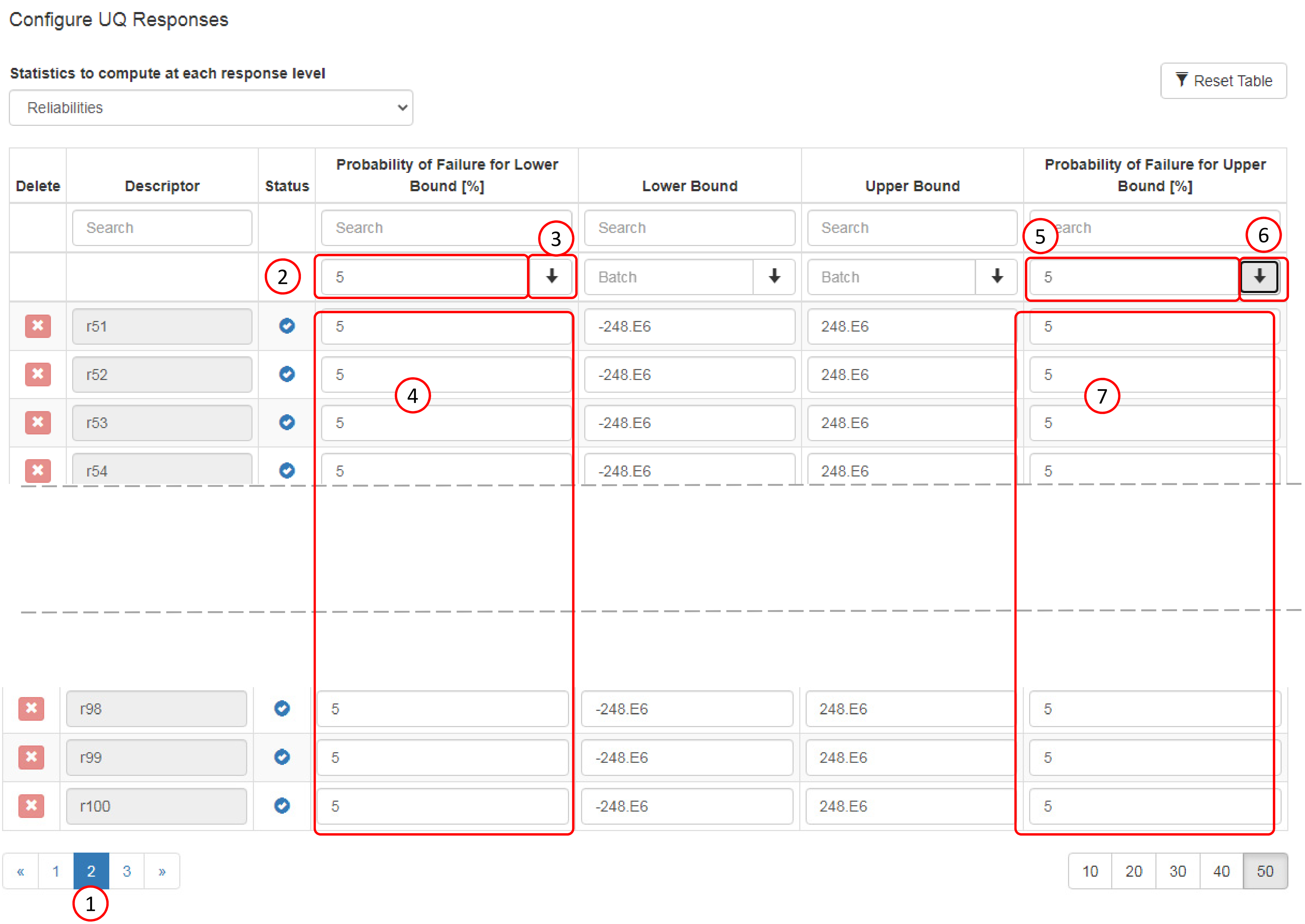
Dakota - Optimization Under Uncertainty (OUU)

1. Navigate to section Configure OUU Constraints
2. Set Statistics to compute at each response level to Reliabilities
3. Set the number of visible rows to 50
4. Set the indicated input box to 5
5. Click the indicated button
6. A maximum probability of failure of 5% has been defined for each response's lower bound
7. Set the indicated input box to 5
8. Click the indicated button
9. A maximum probability of failure of 5% has been defined for each response's upper bound
10. Clear any constraints and bounds for response r1. The weight response r1 is not to have any constraints.
11. Verify that there are no constraints or bounds defined for response r1.



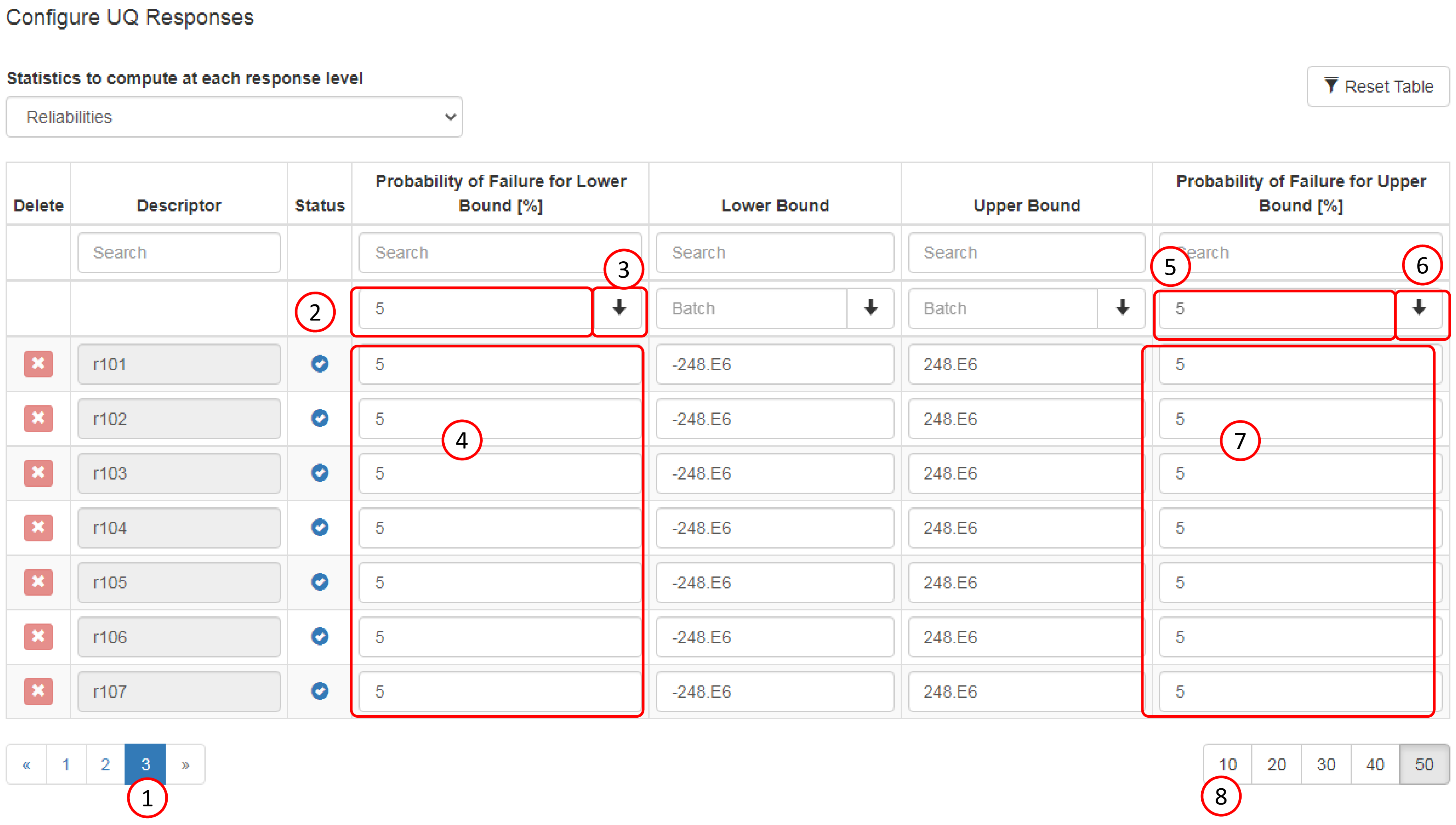
Dakota - Optimization Under Uncertainty (OUU)

1. Click page 2 to view the next set of rows of the table
2. Set the indicated input box to 5
3. Click the indicated button
4. A maximum probability of failure of 5% has been defined for each response's lower bound
5. Set the indicated input box to 5
6. Click the indicated button
7. A maximum probability of failure of 5% has been defined for each response's upper bound



Dakota - Optimization Under Uncertainty (OUU)

1. Click page 3 to view the next set of rows of the table
2. Set the indicated input box to 5
3. Click the indicated button
4. A maximum probability of failure of 5% has been defined for each response's lower bound
5. Set the indicated input box to 5
6. Click the indicated button
7. A maximum probability of failure of 5% has been defined for each response's upper bound
8. Click the indicated button to view at most 10 rows in the table. Reducing the number of visible rows helps improve the performance of the web app.



Dakota - Optimization Under Uncertainty (OUU)

1. Navigate 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. A positive scale factor will minimize the response. A negative scale factor will maximize the response.
4. Notice that the constraints on probability of failure have been automatically created.

Configure OUU Objective and Additional Constraints

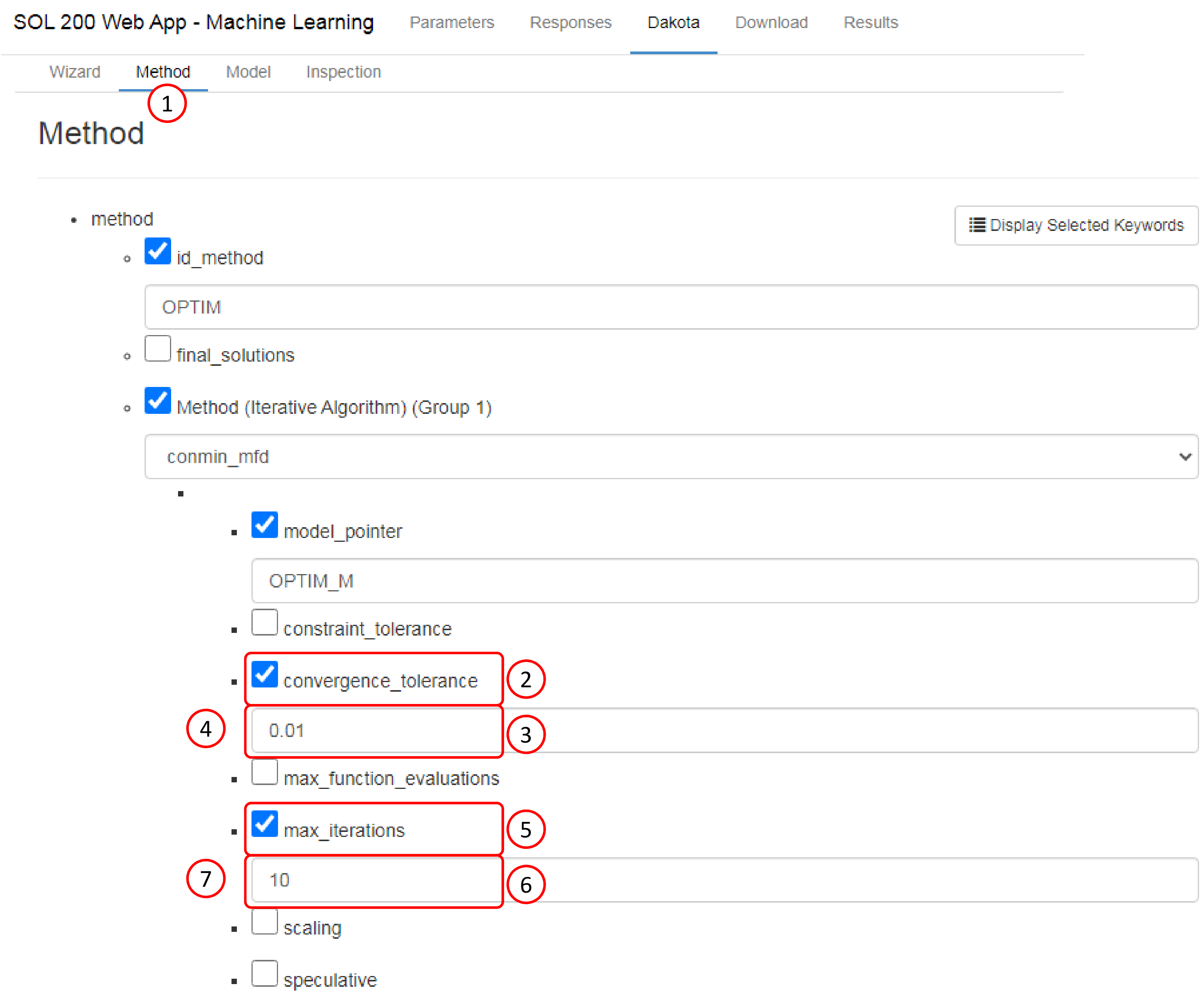
+ Create Constraint

	Objective (f_obj, g1)		Constraint 1 (r2_pl)		Constraint 2 (r2_pu)	
Label	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						
r3_p2						
r4_mean						
r4_standard_deviation						

Uncertainty Quantification

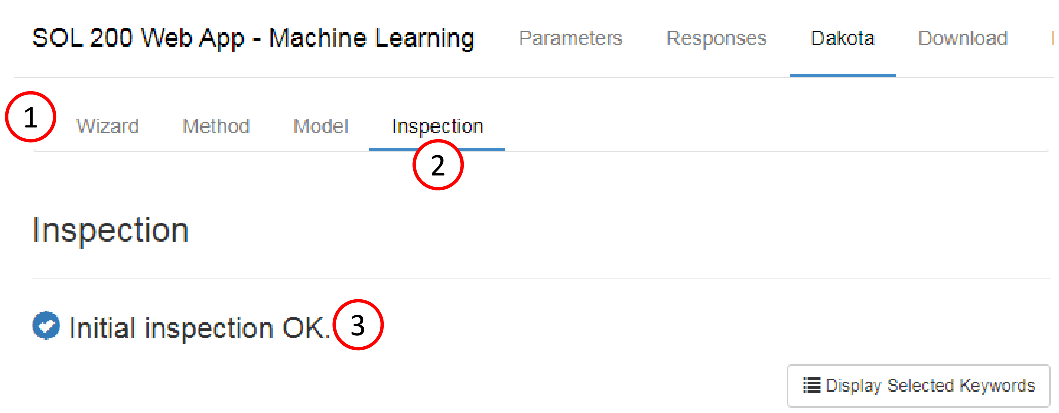
1. Click Method
2. Mark the indicated checkbox to turn on the keyword convergence_tolerance
3. Set the indicated input box to 0.01
4. Reminder! Ensure convergence_tolerance is set to 0.01. This is a step that is very easy to overlook.
5. Mark the indicated checkbox to turn on the keyword max_iterations
6. Set the indicated input box to 10
7. Reminder! Ensure max_iterations is set to 10. This is a step that is very easy to overlook.

- This OUU takes approximately 50 iterations to converge with the default tolerances, but yields very little improvement in the objective after 5-10 iterations. The OUU may be terminated earlier by increasing the convergence tolerance OR by limiting the maximum number of iterations. In this exercise, both are adjusted.
- Another keyword worth considering is the keyword max_function_evaluations. This controls the maximum number of times the optimizer acquires statistics and gradients of the statistics. This is not to be confused with the number of times the black box function is executed to acquire responses. For example, if max_function_evaluations=3, the UQ is performed at most 3 times to acquire statistics or gradients. Each UQ will require a separate number of black box function runs.



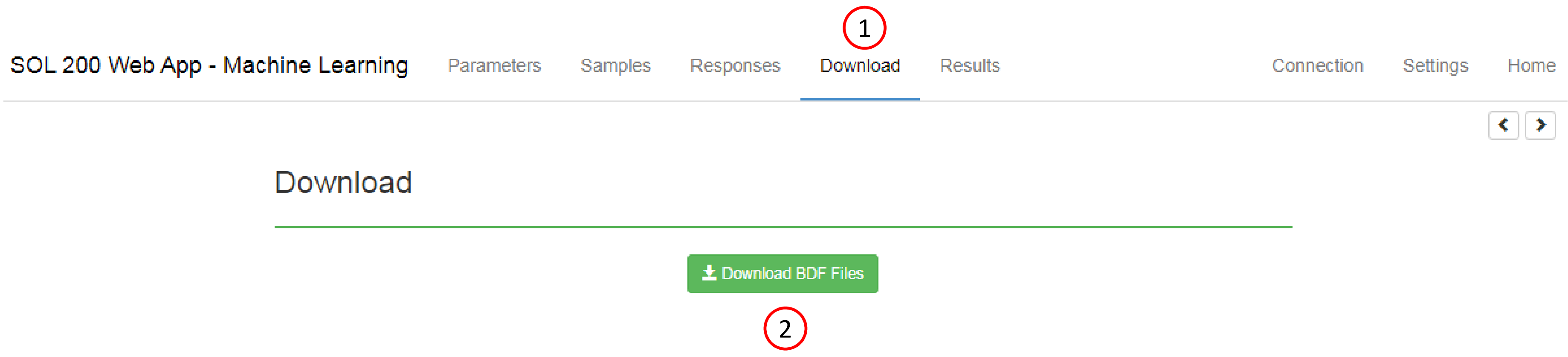
Inspection

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



Download

1. Click Download
2. Click Download BDF Files

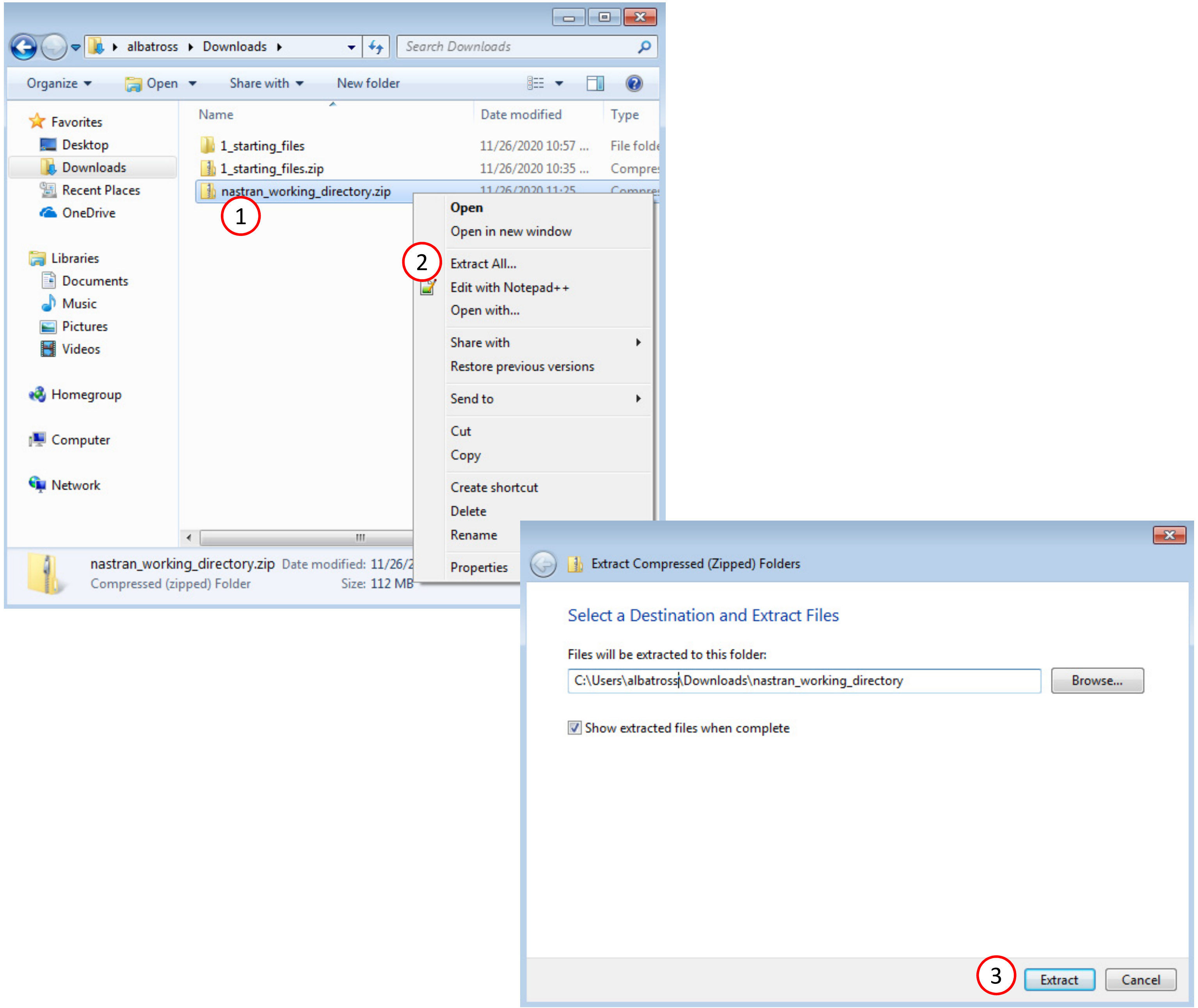


Start MSC Nastran

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.



Update the Initial Values for the Variables

1. Navigate to the following directory

   `1_starting_files/0_solution_files_sol_200_local_optimization`

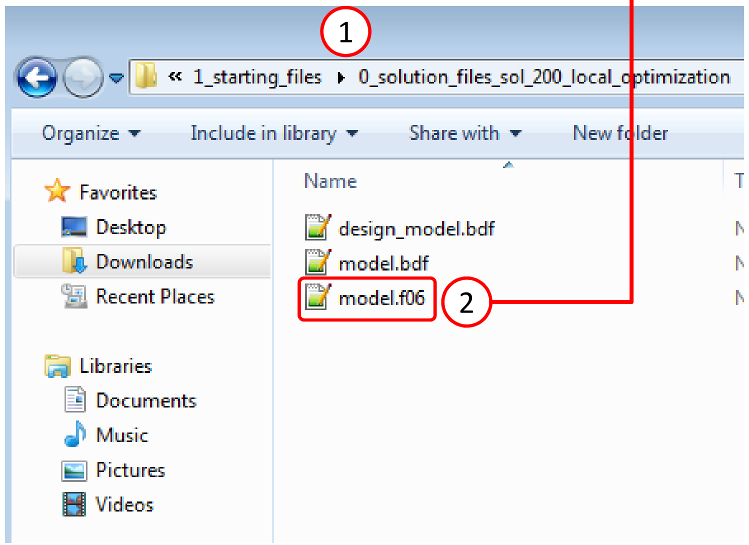
2. Open file model.f06 in Notepad++

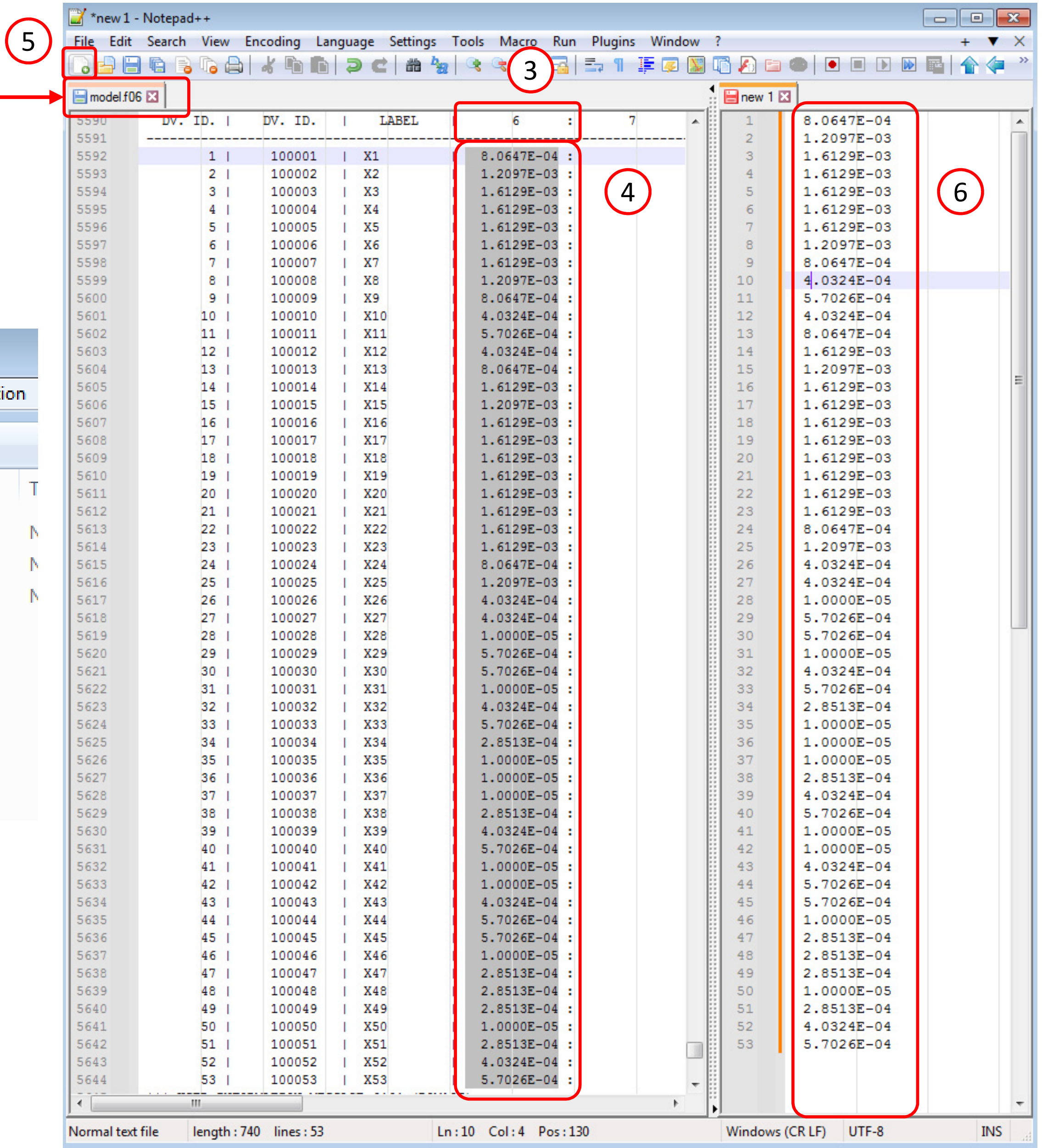
3. Scroll to end of this file and find the optimal solution listed under design cycle 6. These are the optimal variable values when an optimization with deterministic responses was performed with MSC Nastran SOL 200.

4. Press and hold CTRL+ALT on the keyboard and use the left mouse button to select the indicated values for all 53 variables. CTRL+ALT allows for vertically selecting text.

5. Click the indicated button to create a new text file. This new file will be opened in a new table, e.g. new 1.

6. Paste the selected values to the new text file.

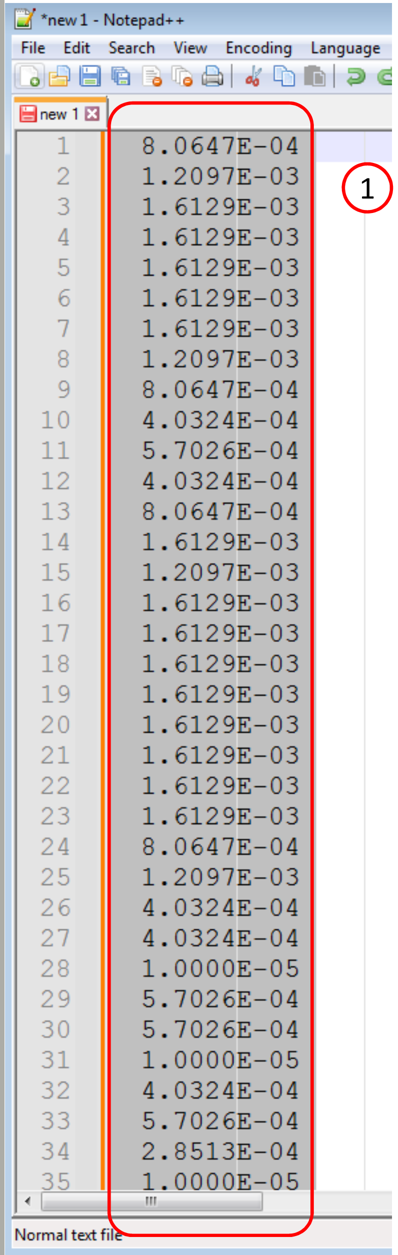


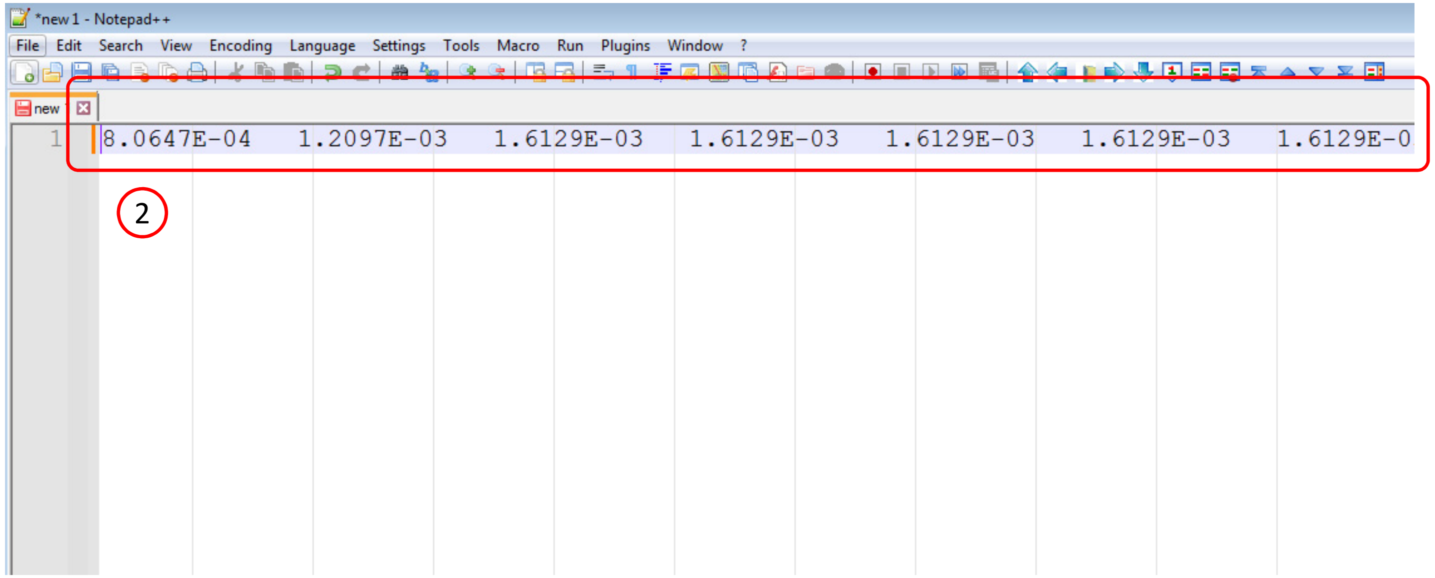


Update the Initial Values for the Variables

1. In the new text file, select all the values with the mouse or with CTRL+A

2. Press CTRL+J to join all the lines





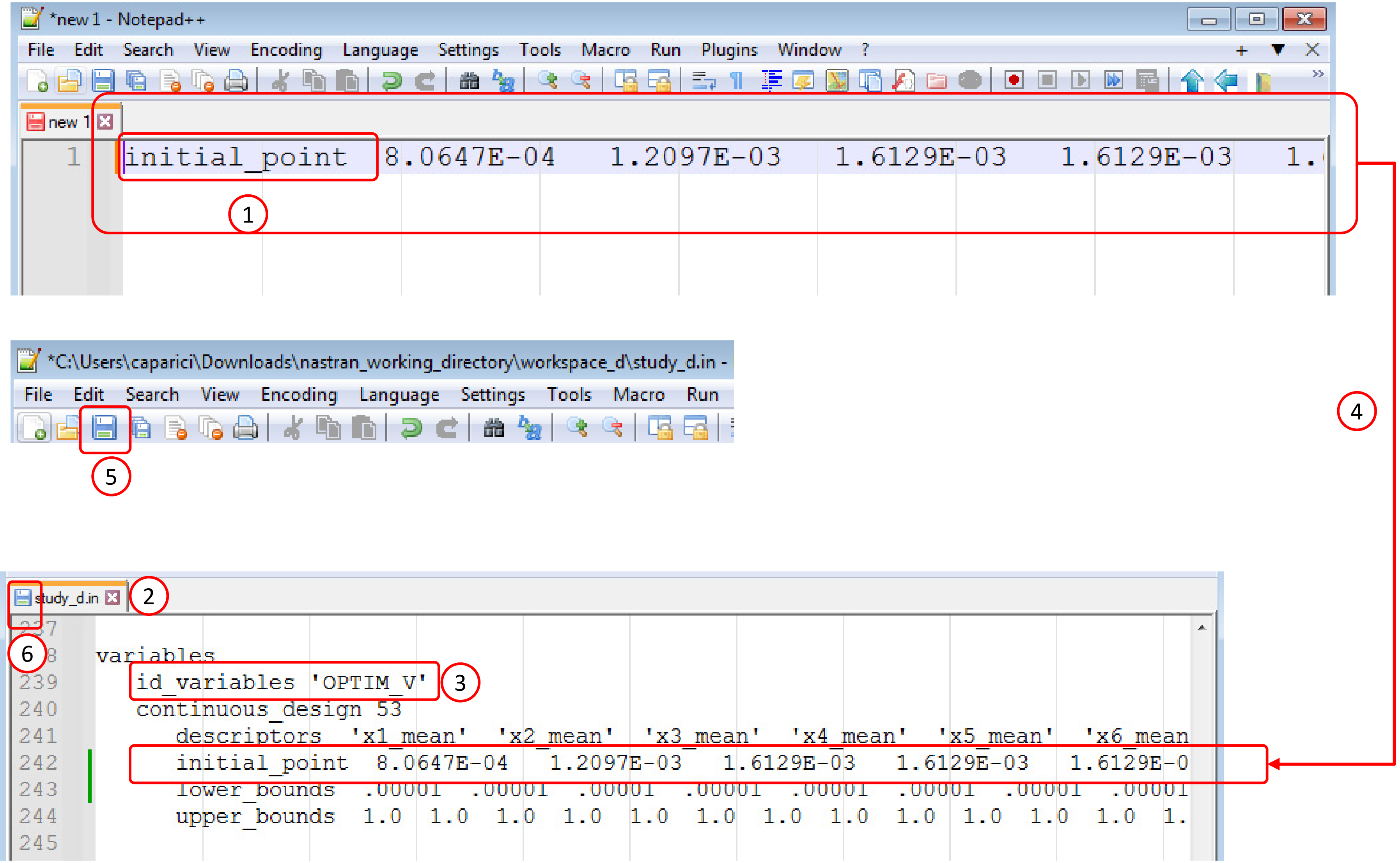
Update the Initial Values for the Variables

1. Add the keyword *initial_point* to the beginning of the line and include a space. Example:

   `initial_point    8.0647E-04    [...]`

2. Open file nastran_working_directory/workspace_d/study_d.in
3. Search for the keyword block with id_variables='OPTIM_V'
4. Copy the new line and replace the old line found in study_d.in
5. Click save
6. Ensure the save icon is blue, not red. A blue color indicates the file has been saved. A red icon indicates the file has not been saved.

The initial values of the variables has been updated for all 53 OUU variables.



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

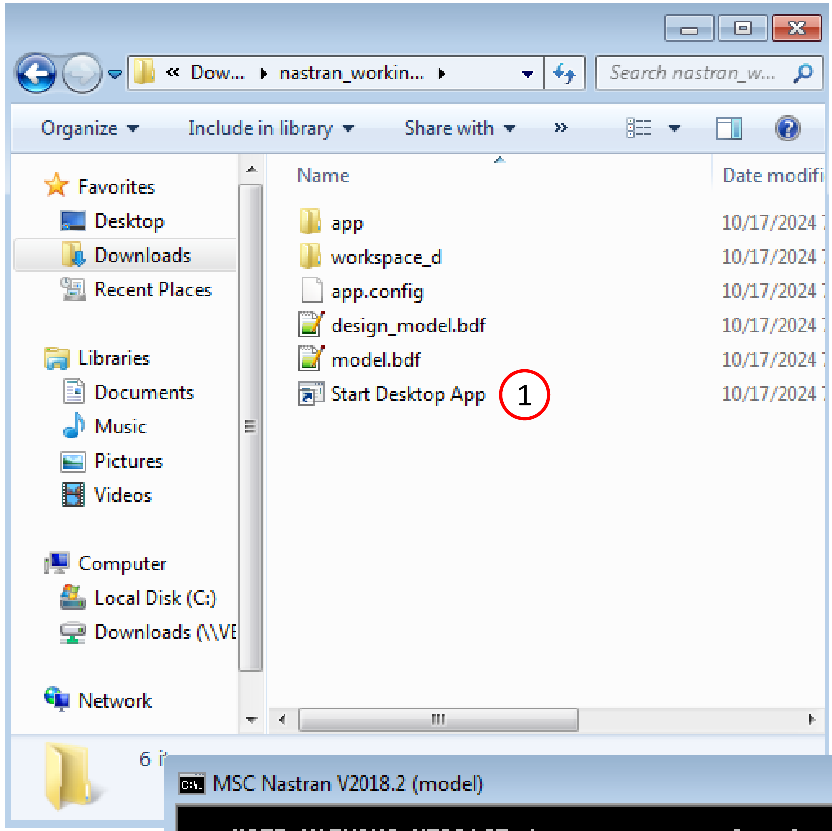
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click "Start Desktop App" to display the results.

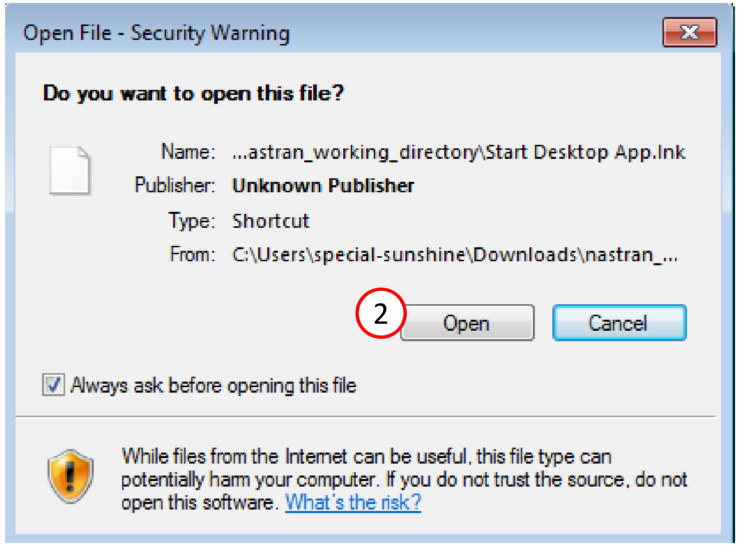
Using Linux?

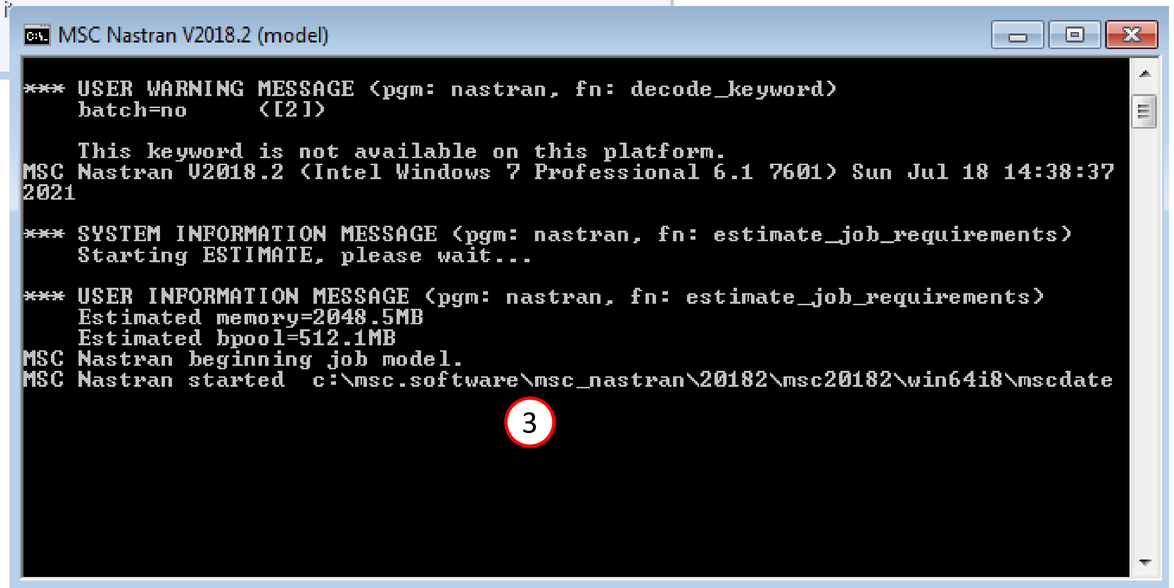
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

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

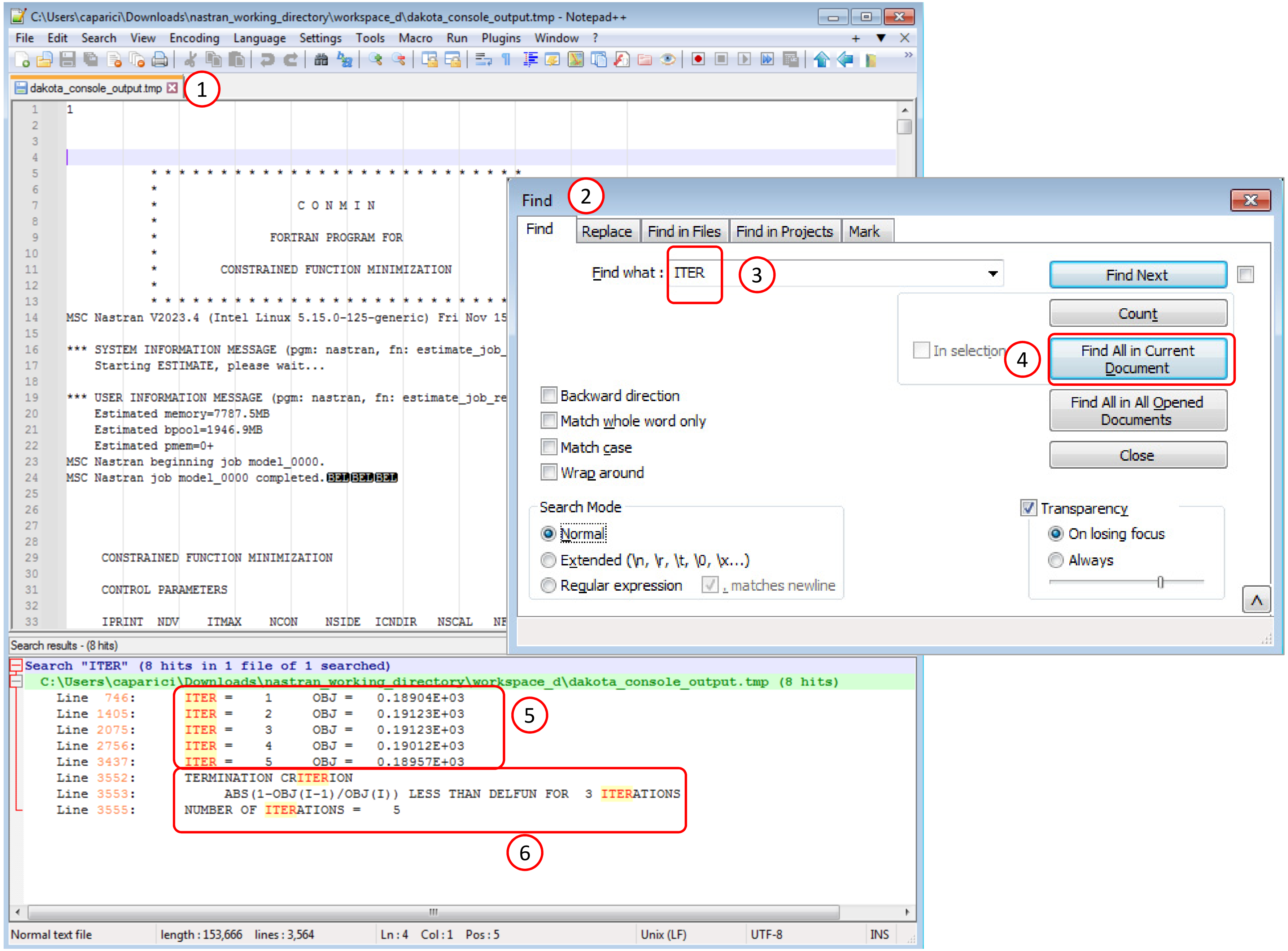
Name	Status of Job	Design Cycle	RUN TERMINATED DUE TO
model.bdf	Running	None	

Long OUU

Since this OUU has over 50 variables, the OUU is expected to take a long time and may require between 100 to 500 FEA runs.

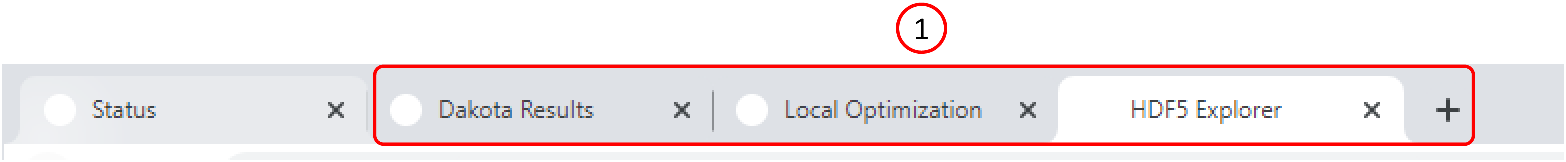
During OUU, the progress may be inspected as follows.

1. Open file dakota_console_output.tmp in Notepad++
2. Press CTRL+F on the keyboard to open the Find window
3. Search for ITER
4. Click Find All in Current Document
5. Information regarding the OUU progress is displayed
6. Since the convergence tolerance was reached, the OUU terminates. If max_iterations was set to 3, the OUU would have terminated after iteration 3. In this case, max_iterations was set to 10. The termination criteria for convergence tolerance was reached first before the maximum number of iterations.



OUU Completion

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

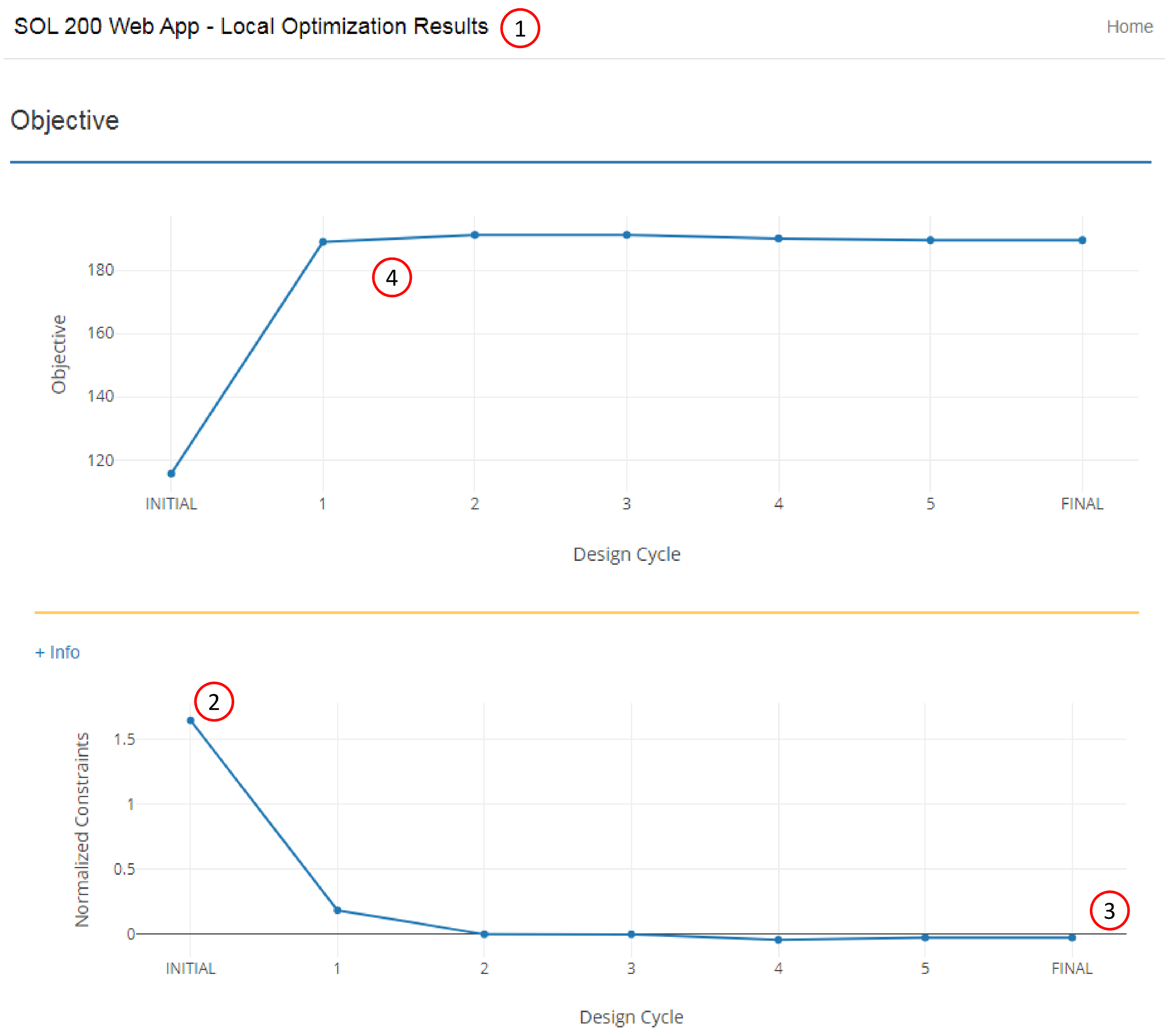


OUU Results

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.

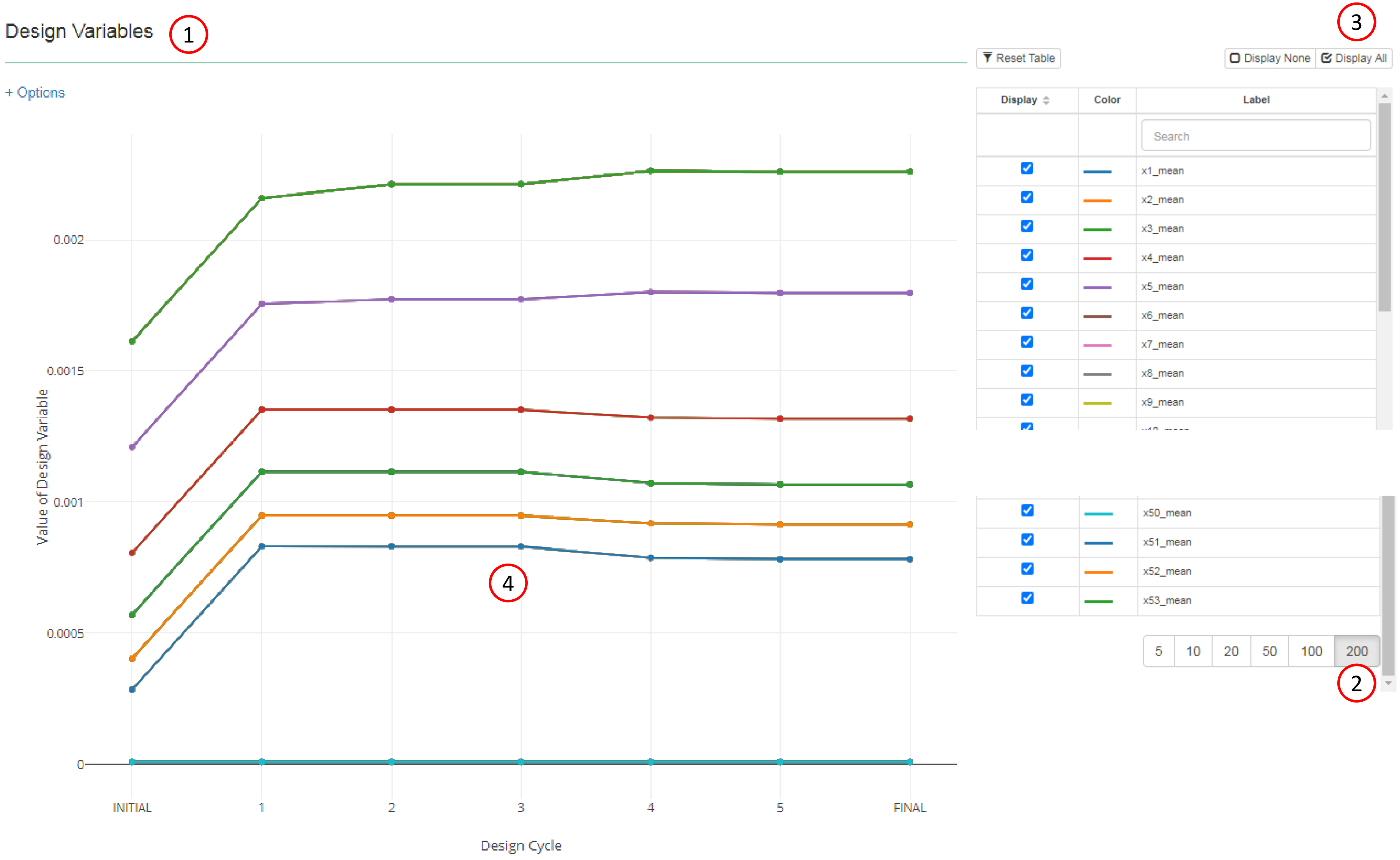
2. Note that the start of the optimization, the normalized constraint is very high and positive, indicating the initial design was infeasible.

3. At the end of the optimization, the normalized constraint is negative. Negative or near zero constraint values indicate a feasible design. This optimization has terminated at a feasible design.



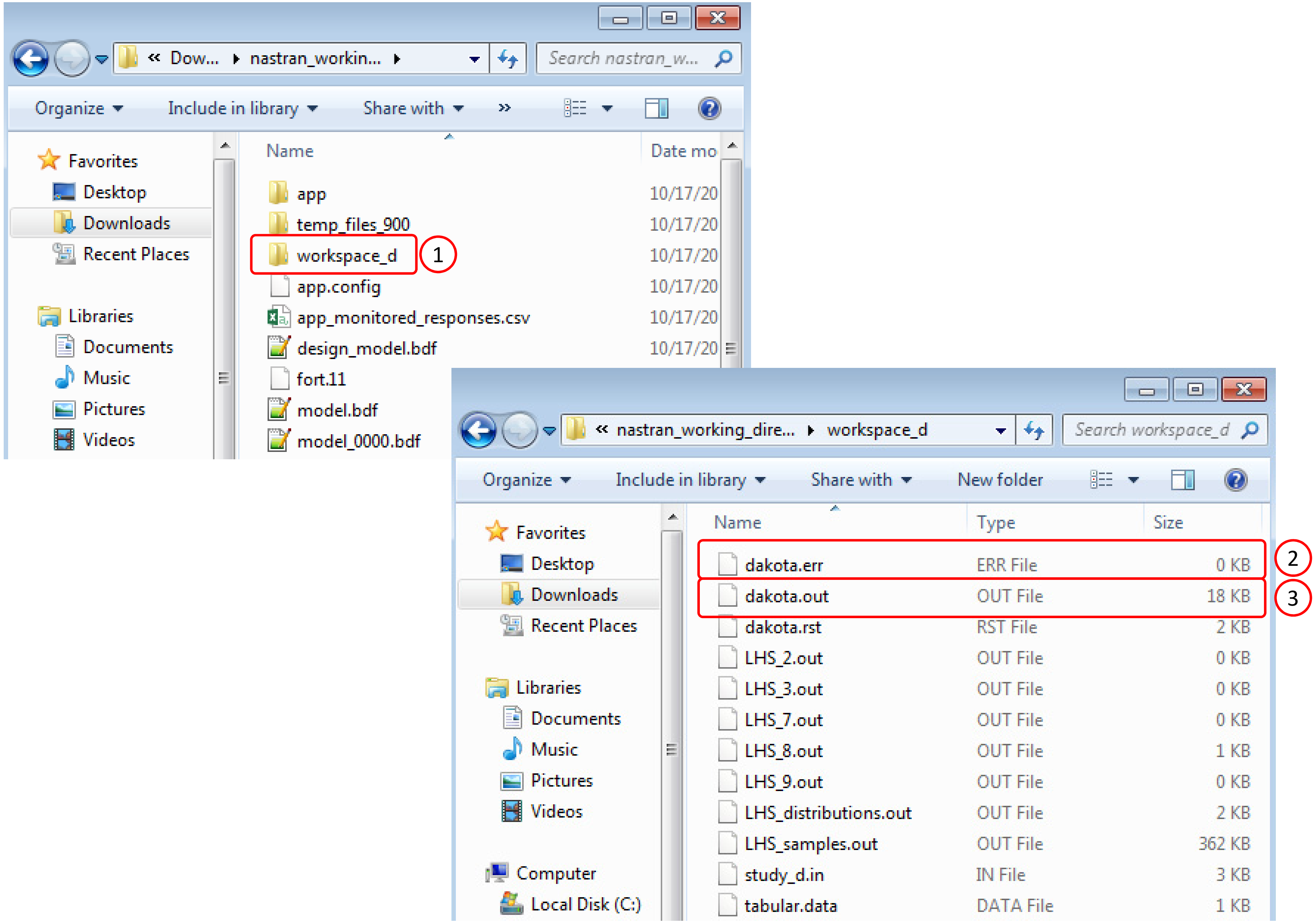
OUU Results

1. Navigate to section Design Variables
2. Click the indicated button to display at most 200 rows in the table
3. Click Display All
4. The change of all the design variables during the OUU are displayed



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 the variables are listed. Only a partial set of the results are displayed.
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 index values.
5. During the OUU, the optimizer has acquired response or gradients for 284 designs, where a design is a different set of variable values. 282 of these evaluations were unique, while 2 evaluations were non-unique. MSC Nastran was run 282 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 1262.77 seconds (~21 minutes)
    1. Local reliability with MVFOSM was used.
    2. A SOL 200 optimization was performed to determine ideal initial values for the OUU variables.
    3. Internally, reliabilities were constrained instead of probabilities.

1 5

```
<<<<< Function evaluation summary (UQ_I): 284 total (282 new, 2 duplicate)
<<<<< Best parameters          =
                      1.3179002125e-03 x1_mean
                      1.7973456563e-03 x2_mean
                      2.2591465096e-03 x3_mean
                      [...]
                      7.8231184718e-04 x51_mean
                      9.1467021248e-04 x52_mean
                      1.0674418472e-03 x53_mean
<<<<< Best objective function  =
                      1.8957040000e+02
<<<<< Best constraint values   =
                     -1.1250630719e+01
                      5.9790298704e+00
                     -7.0873346063e+00
                      1.4276717017e+01
                     -1.7175574977e+01
                     [...]
                     -1.8613891157e+00
                      6.1311565463e+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        =     3.19415 [parent =     3.19433, child =   -0.000179]
  Total wall clock =     1262.77
```

2 3 4 6

OUU Results

Although initial configurations were set to expedite the optimization, more than 100 MSC Nastran runs were performed during the OUU, which some might say is still costly.

OUU repeats the following 2 steps until convergence.

1. Perform an uncertainty quantification at xi
2. The optimizer uses the UQ responses to find the next best xi

The uncertainty quantification is expedited by using the MVFOSM method, which depends on responses ri and gradients dri/dxi. MSC Nastran is executed to output both ri and dri/dxi.

The optimizer must then perform its search for the next best xi while considering statistical responses such as means, standard deviations and probabilities, e.g. ri_mean, ri_standard_deviation, ri_probability_1, etc. Analytic gradients for statistical responses (dri_mean/dxi, dri_standard_deviation/dxi, etc.) are needed by the optimizer but are not immediately not available, so Dakota relies on numerical gradients or the finite difference method to determine the necessary gradients. The determination of numerical gradients is what contributes to the large number of black box function evaluations or MSC Nastran runs.

1. There are alternative UQ and OUU methods, but many of these methods are subject to the curse of dimensionality and may require thousands of black box function runs to perform the OUU. Ultimately, many of these alternative methods may be limited to OUU problems involving 1-10 variables.

```
<<<<< Function evaluation summary (UQ_I): 284 total (282 new, 2 duplicate)
<<<<< Best parameters          =
                      1.3179002125e-03 x1_mean
                      1.7973456563e-03 x2_mean
                      2.2591465096e-03 x3_mean
                      [...]
                      7.8231184718e-04 x51_mean
                      9.1467021248e-04 x52_mean
                      1.0674418472e-03 x53_mean
<<<<< Best objective function  =
                      1.8957040000e+02
<<<<< Best constraint values   =
                     -1.1250630719e+01
                      5.9790298704e+00
                     -7.0873346063e+00
                      1.4276717017e+01
                     -1.7175574977e+01
[...]
                     -1.8613891157e+00
                      6.1311565463e+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        =     3.19415 [parent =     3.19433, child =  -0.000179]
  Total wall clock =     1262.77
```

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	1.8957040000e+02

Constraints

Reset Table | Display Additional Columns

5

Feasible

3 4

Descriptor	Lower Bound	Upper Bound	Value	Constraint Satisfied
Search				
r2_pl	-inf	-1.644854	-1.1250630719e+01	✔
r2_pu	1.644854	inf	5.9790298704e+00	✔
r3_pl	-inf	-1.644854	-7.0873346063e+00	✔
r3_pu	1.644854	inf	1.4276717017e+01	✔
r4_pl	-inf	-1.644854	-1.7175574977e+01	✔
r4_pu	1.644854	inf	8.1389891037e+00	✔
r5_pl	-inf	-1.644854	-5.6075327977e+00	✔
r5_pu	1.644854	inf	1.4644118467e+01	✔
r6_pl	-inf	-1.644854	-1.2956481235e+01	✔
r6_pu	1.644854	inf	3.9198953937e+00	✔

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

Reset Table | Display Additional Columns (1) | Display All as Probability of Failure

Feasible

Descriptor	Lower Bound	Upper Bound	Value	pf/ps	Description	Probability	Reliability Index	Constraint Satisfied
Search					(2)			
r2_pl	-inf	-1.644854	-1.1250630719e+01	p_s	P(-2.4800e+8 < r2)	1.0000e+0	-1.1250630719e+01	
r2_pu	1.644854	inf	5.9790298704e+00	p_f	P(2.4800e+8 < r2)	1.1224e-9	5.9790298704e+00	
r3_pl	-inf	-1.644854	-7.0873346063e+00	p_s	P(-2.4800e+8 < r3)	1.0000e+0	-7.0873346063e+00	
r3_pu	1.644854	inf	1.4276717017e+01	p_f	P(2.4800e+8 < r3)	0.0000e+0	1.4276717017e+01	
r4_pl	-inf	-1.644854	-1.7175574977e+01	p_s	P(-2.4800e+8 < r4)	1.0000e+0	-1.7175574977e+01	

Constraints

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					(4)			
r2_pl	-inf	-1.644854	-1.1250630719e+01	p_f	P(r2 < -2.4800e+8)	0.0000	-inf	
r2_pu	1.644854	inf	5.9790298704e+00	p_f	P(2.4800e+8 < r2)	1.1224e-9	5.9790298704e+00	
r3_pl	-inf	-1.644854	-7.0873346063e+00	p_f	P(r3 < -2.4800e+8)	0.0000	-inf	
r3_pu	1.644854	inf	1.4276717017e+01	p_f	P(2.4800e+8 < r3)	0.0000e+0	1.4276717017e+01	
r4_pl	-inf	-1.644854	-1.7175574977e+01	p_f	P(r4 < -2.4800e+8)	0.0000	-inf	

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. In the indicated search bar, search for character *
2. The search reveals the following constraints are the most active or most violated
3. The probabilities of failure are 4.73%, which are very close to the limit of 5% specified in this exercise

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	1.8957040000e+02

Constraints

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

Feasible

Descriptor	Lower Bound	Upper Bound	Value	pf/ps	Description	Probability	Reliability Index	Constraint Satisfied
* 1						3		
r55_pu *	1.644854	inf	1.6716146649e+00	p_f	P(2.4800e+8 < r55)	4.7300e-2	1.6716146649e+00	✓
r63_pu *	1.644854	inf	1.6716146649e+00	p_f	P(2.4800e+8 < r63)	4.7300e-2	1.6716146649e+00	✓
r67_pu *	1.644854	inf	1.6716146649e+00	p_f	P(2.4800e+8 < r67)	4.7300e-2	1.6716146649e+00	✓
r78_pu *	1.644854	inf	1.6716146649e+00	p_f	P(2.4800e+8 < r78)	4.7300e-2	1.6716146649e+00	✓

2

* This constraint is the most active or most violated

Discussion of Final Probabilities of Failure

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

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

The highest probability of failure is 4.73%.

Status | Dakota Results | Local Optimization | HDF5 Explorer

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

Constraints P(a < X ≤ b) 1

Reset Table

Descriptor	Description	ps	pf
* 2	4	5	6
r55 *	P(-248.E6 < r55 ≤ 248.E6)	95.2700%	4.7300%
r63 * 3	P(-248.E6 < r63 ≤ 248.E6)	95.2700%	4.7300%
r67 *	P(-248.E6 < r67 ≤ 248.E6)	95.2700%	4.7300%
r78 *	P(-248.E6 < r78 ≤ 248.E6)	95.2700%	4.7300%

* This response has the highest probability of failure

Comparison of Approximate and Actual p_f

- In part A, the OUU results are based on approximated statistics.
- In part B, an UQ is configured by using the sampling method with an LHS of size 200 . These resulting statistics are deemed the actual statistics. The approximated and actual statistics are compared and a conclusion is determined if the final design satisfies the constraints on probabilities of failure.

Response	**Part A - OUU** **Approximated Reliability Index (Equivalent Probability of Failure)**	**Part B - UQ Generated** **(Actual Probability of failure)**
Comments	The OUU considered and reported reliability indices	These are the probabilities after an LHS of size 200
Maximum Probability of Failure	1.6716146649e+00 (4.73%) for responses r55_pu, r63_pu, r67_pu, r78_pu	(5%) for response r98_pl

Part B – Verification of OUU Solution

Motivation

Part A - An OUU was performed using reliability methods. Specifically, the MVFOSM was used to approximate the tail probabilities (probabilities of failure). These are termed the approximate tail probabilities or approximate probabilities of failure.

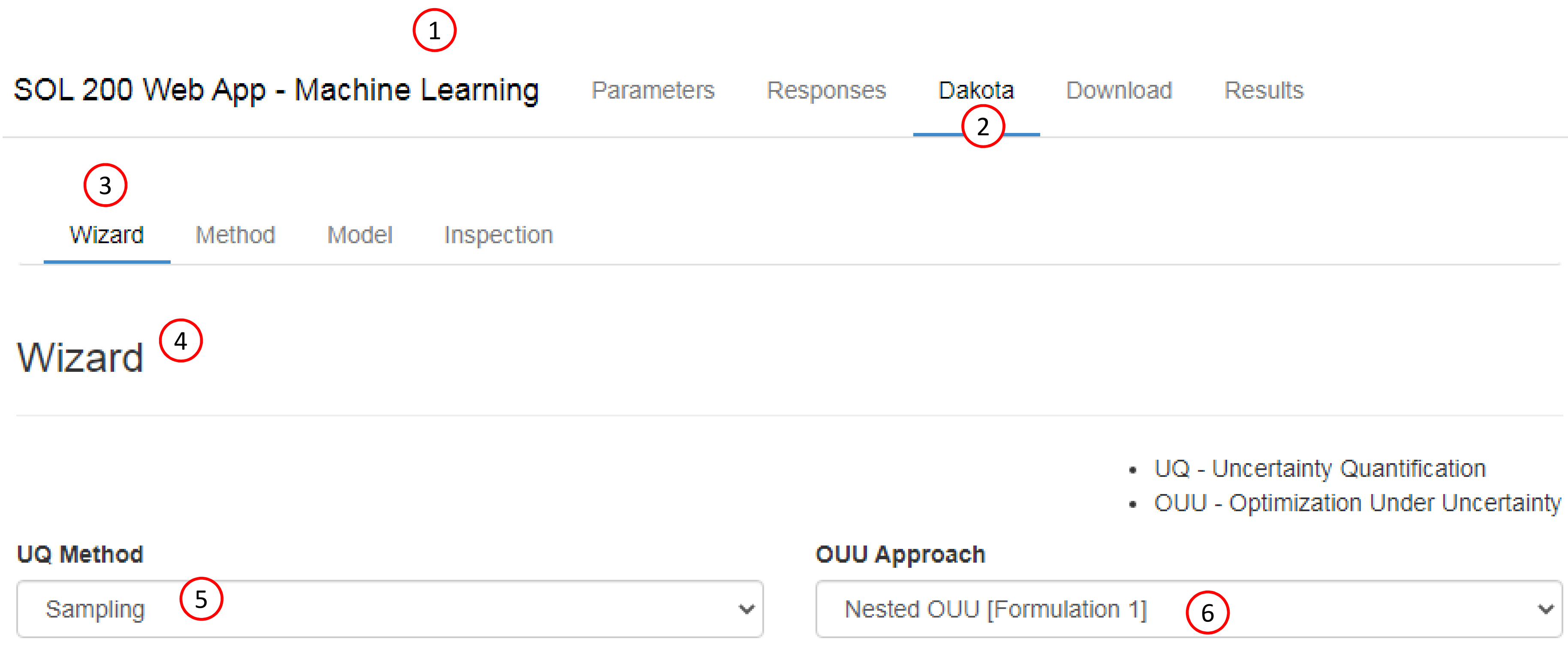
Part B - An LHS of size 200 (200 MSC Nastran runs) is evaluated and the tail probabilities are calculated. These tail probabilities are deemed the actual tail probabilities or actual probabilities of failure.

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

Uncertainty Quantification

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

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



Uncertainty Quantification

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

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

There are 53 variables to update, but this step only updates 4 variables. Later on, all the variables are updated using a faster procedure.

Configure OUU Variables 1

Reset Table

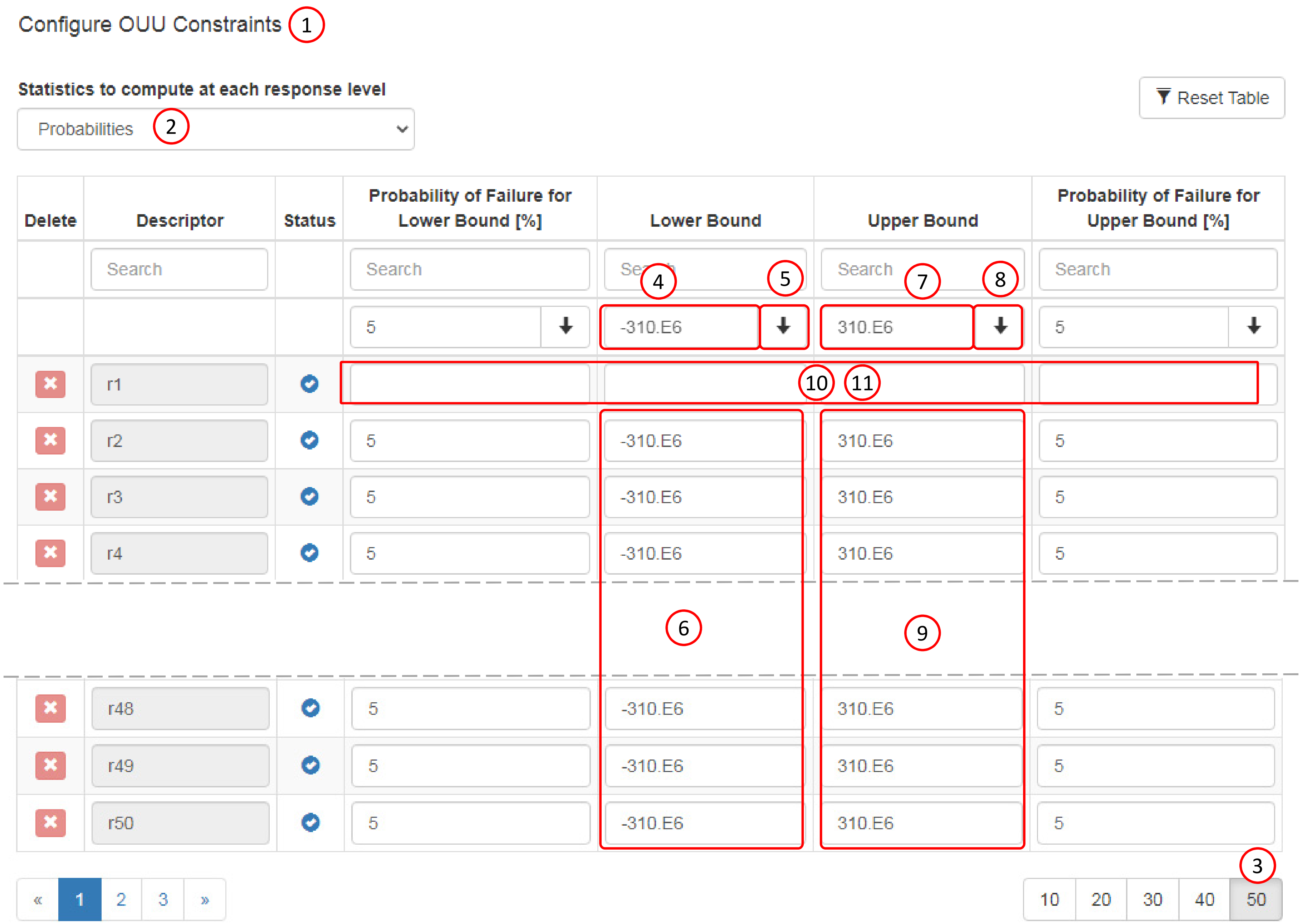
Delete	Descriptor	Status	Initial Value	Lower Bound	Upper Bound	Description
	Search		Search	Search	Search	Search
			Batch	.00001	1.0	
✖	x1_mean	✔	1.3179002125E-03	.00001	1.0	Mean - Field 4 of DESVAR
✖	x2_mean	✔	1.7973456563E-03	.00001	1.0	Mean - Field 4 of DESVAR
✖	x3_mean	✔	2.2591465096E-03	.00001	1.0	Mean - Field 4 of DESVAR
✖	x4_mean	✔	2.2591465096E-03	.00001	1.0	Mean - Field 4 of DESVAR
✖	x5_mean	✔	.01	.00001	1.0	Mean - Field 4 of DESVAR

2

```
<<<<< Function evaluation summary (UQ_I): 284 total (282 new, 2 duplicate)
<<<<< Best parameters          =
                      1.3179002125e-03 x1_mean
                      1.7973456563e-03 x2_mean
                      2.2591465096e-03 x3_mean
                      2.2591465096e-03 x4_mean
                       [...]
                      7.8231184718e-04 x51_mean
                      9.1467021248e-04 x52_mean
                      1.0674418472e-03 x53_mean
<<<<< Best objective function  =
                      1.8957040000e+02
<<<<< Best constraint values   =
                       [...]
```

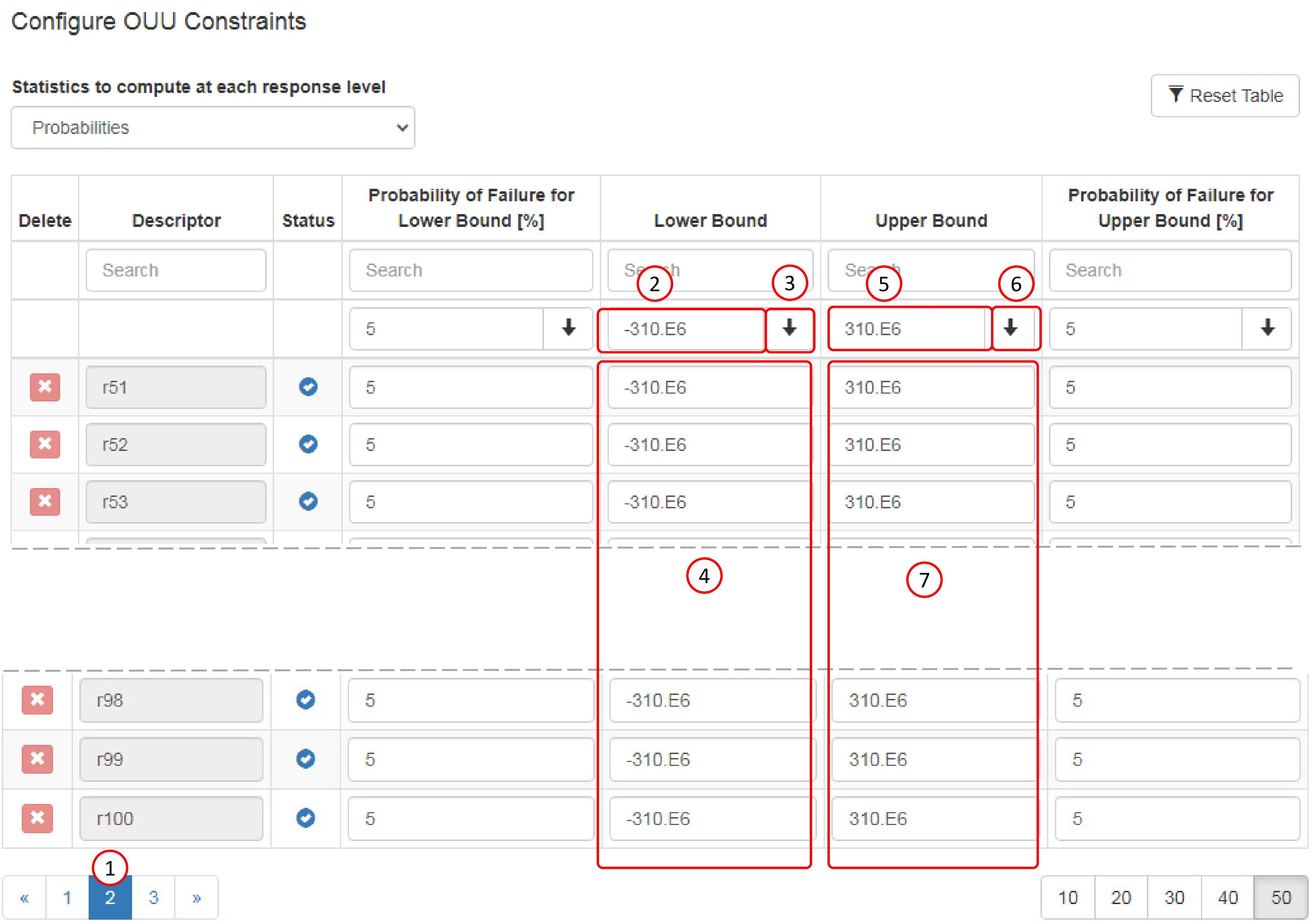
Dakota - Uncertainty Quantification (UQ)

1. Navigate to section Configure OUU Constraints
2. Set Statistics to compute at each response level to Probabilities
3. Set the number of visible rows to 50
4. Set the indicated input box to -310.E6
5. Click the indicated button
6. The lower bound for each visible response has been updated
7. Set the indicated input box to 310.E6
8. Click the indicated button
9. The upper bound for each visible response has been updated
10. Clear any constraints and bounds for response r1. The weight response r1 is not to have any constraints.
11. Verify that there are no constraints or bounds defined for response r1.



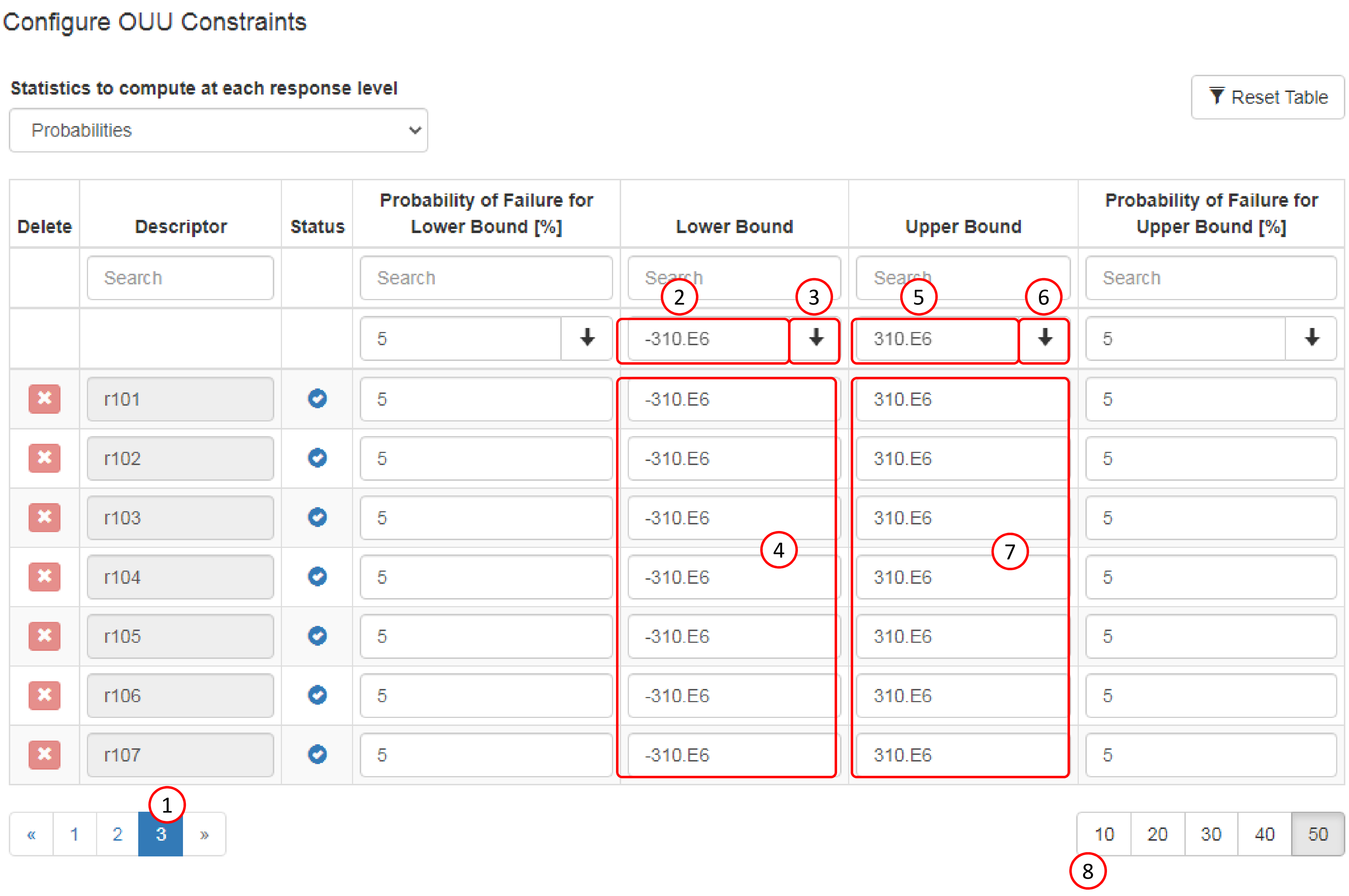
Dakota - Uncertainty Quantification (UQ)

1. Click page 2 to view the next set of rows of the table
2. Set the indicated input box to -310.E6
3. Click the indicated button
4. The lower bound for each visible response has been updated
5. Set the indicated input box to 310.E6
6. Click the indicated button
7. The upper bound for each visible response has been updated



Dakota - Uncertainty Quantification (UQ)

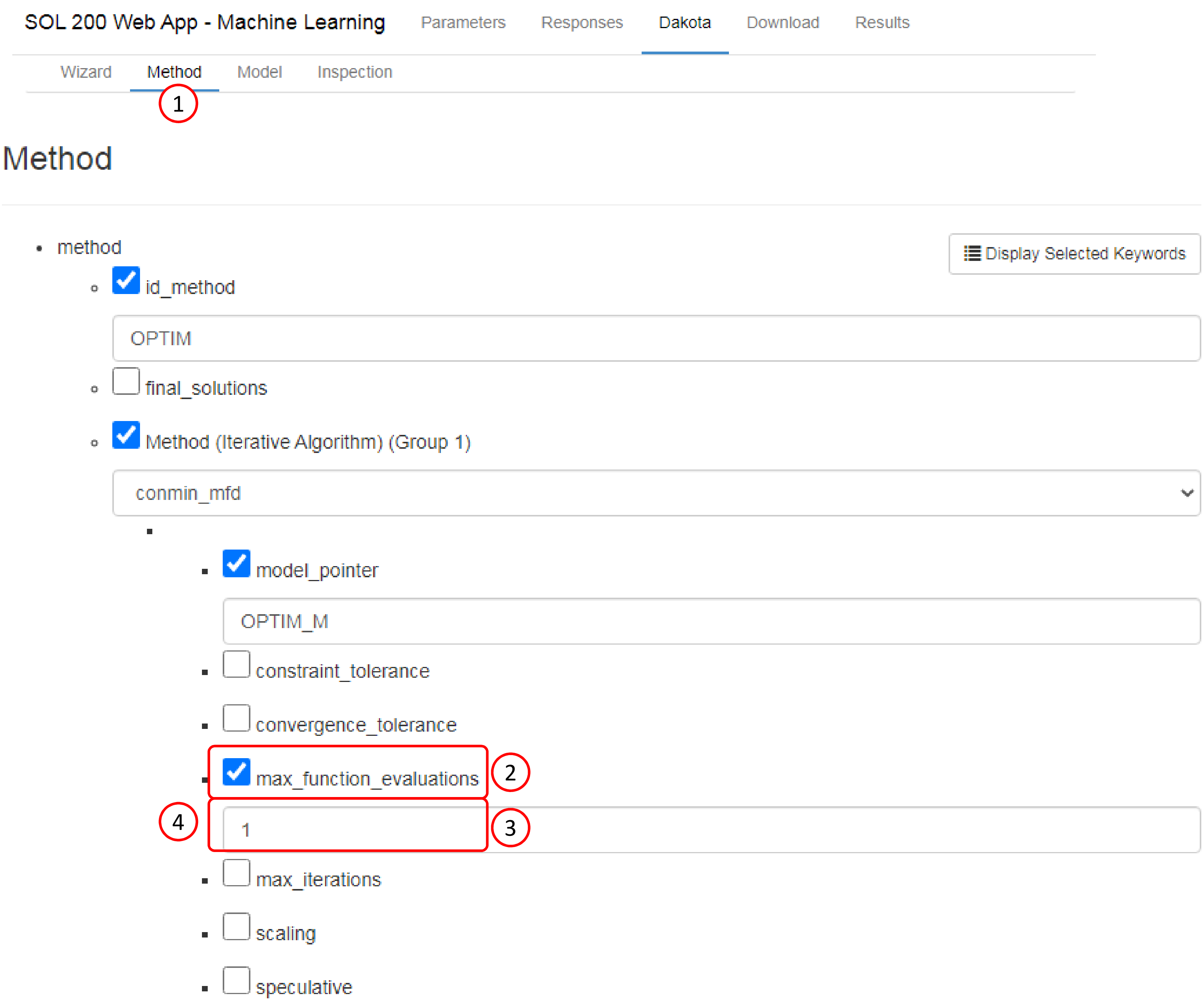
1. Click page 3 to view the next set of rows of the table
2. Set the indicated input box to -310.E6
3. Click the indicated button
4. The lower bound for each visible response has been updated
5. Set the indicated input box to 310.E6
6. Click the indicated button
7. The upper bound for each visible response has been updated
8. Click the indicated button to view at most 10 rows in the table. Reducing the number of visible rows helps improve the performance of the web app.



Uncertainty Quantification

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

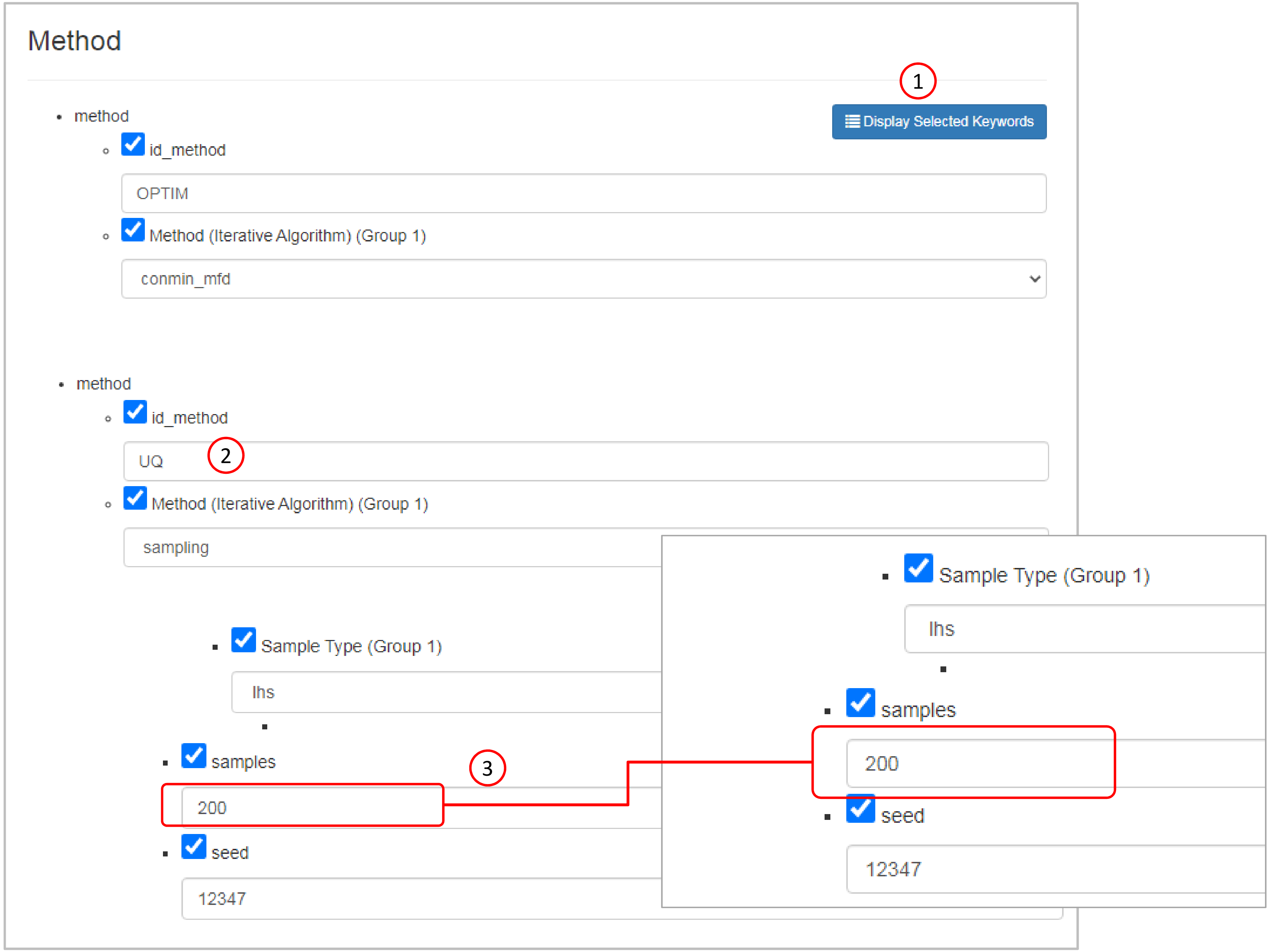
- The goal is to perform an uncertainty quantification and run the optimization procedure only to compute the constraint values, i.e. probabilities of failure. The keyword max_function_evaluations is set to 1 to allow the optimization routine to calculate only constraint values and terminate with zero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



Uncertainty Quantification

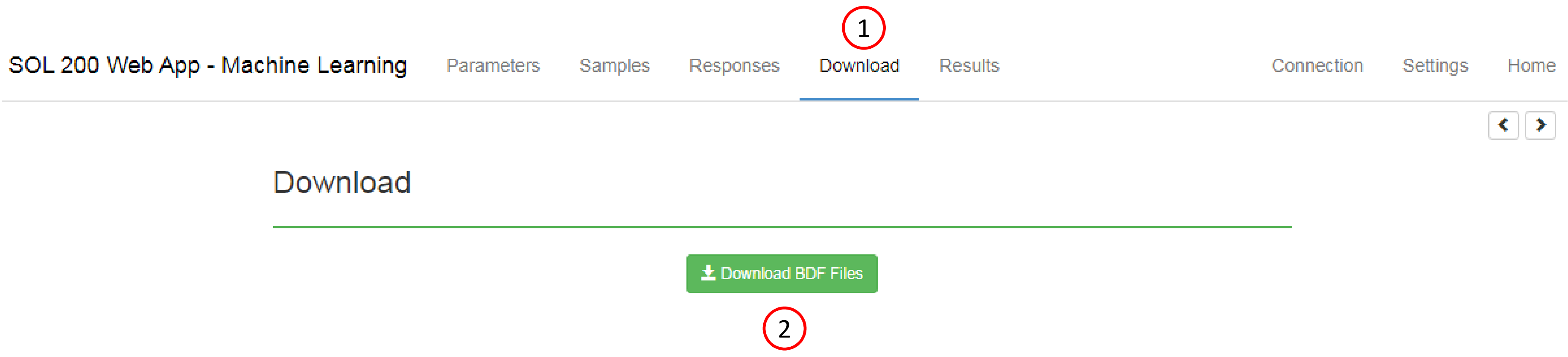
1. Click Display Selected Keywords
2. Scroll to the method keyword block with id_method=UQ
3. Change the number of samples to 200.

An LHS of size 100 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 in part B is to confirm actual probabilities, based on an LHS of size 200, are below the max probability of failure of 5%.



Download

1. Click Download
2. Click Download BDF Files

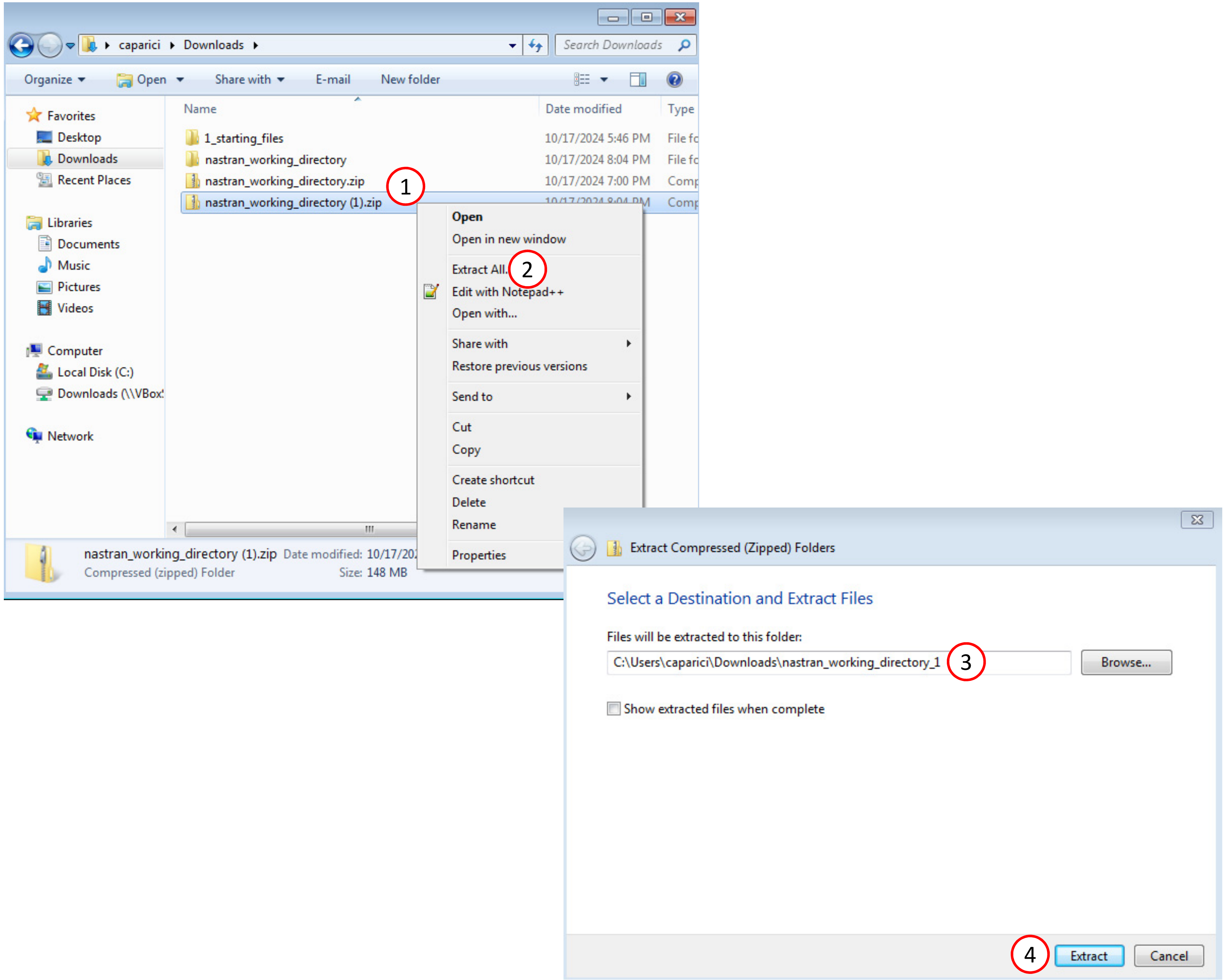


Start MSC Nastran

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.



Update the Initial Values for the Variables

1. Navigate to the following directory

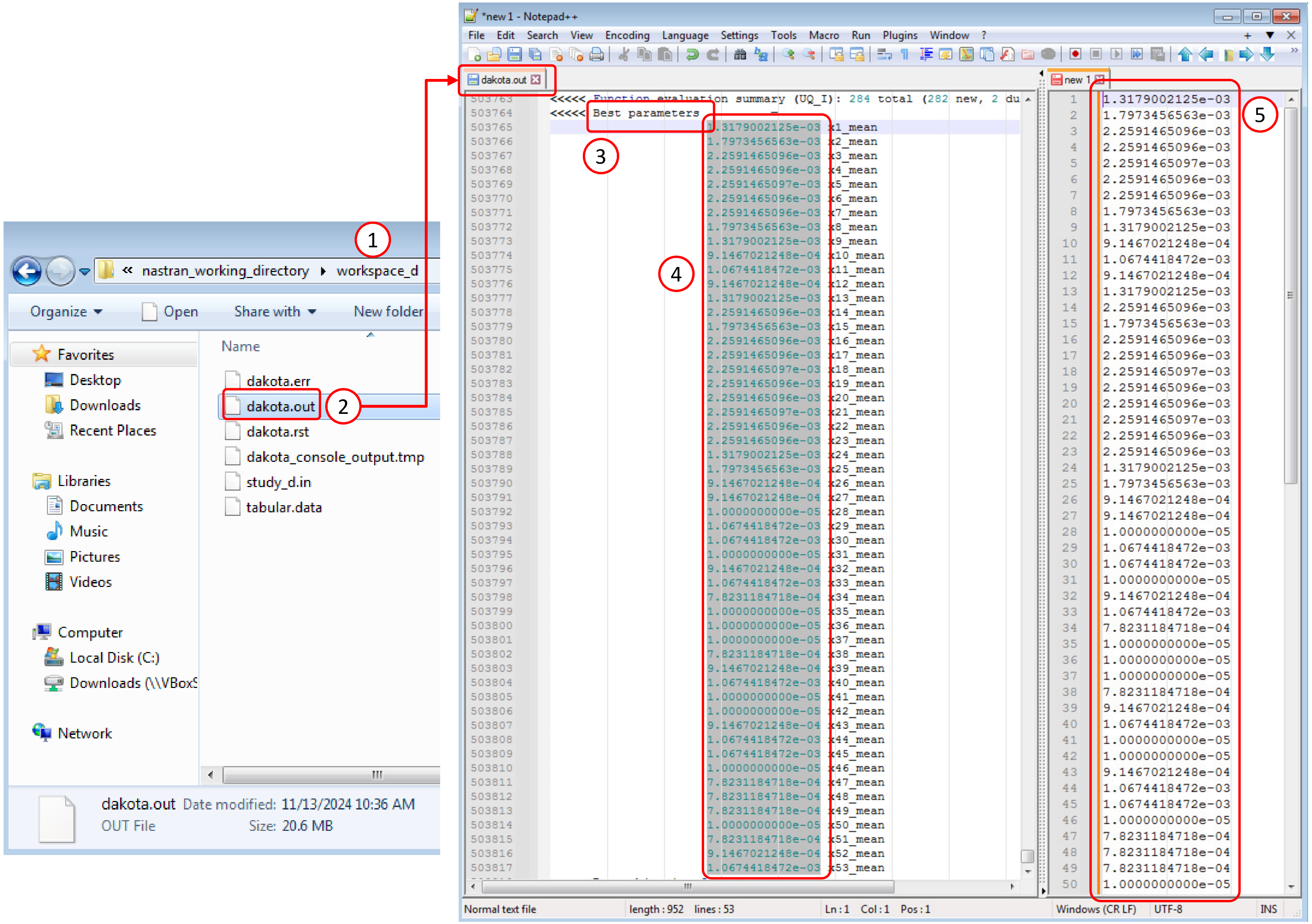
   `nastran_working_directory/workspace_d`

2. Open file dakota.out in Notepad++

3. Scroll to end of this file and find the optimal solution listed under section Best parameters. These are the optimal variable values from the previous OUU.

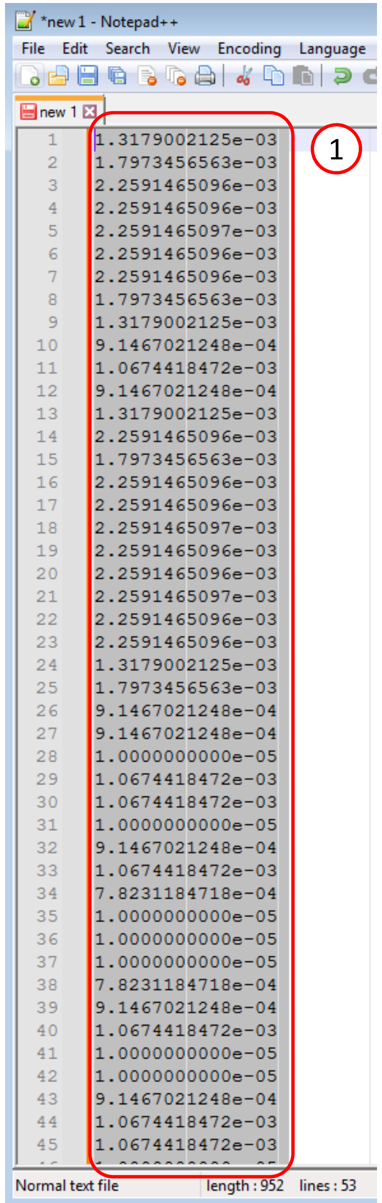
4. Press and hold CTRL+ALT on the keyboard and use the left mouse button to select the indicated values for all 53 variables.

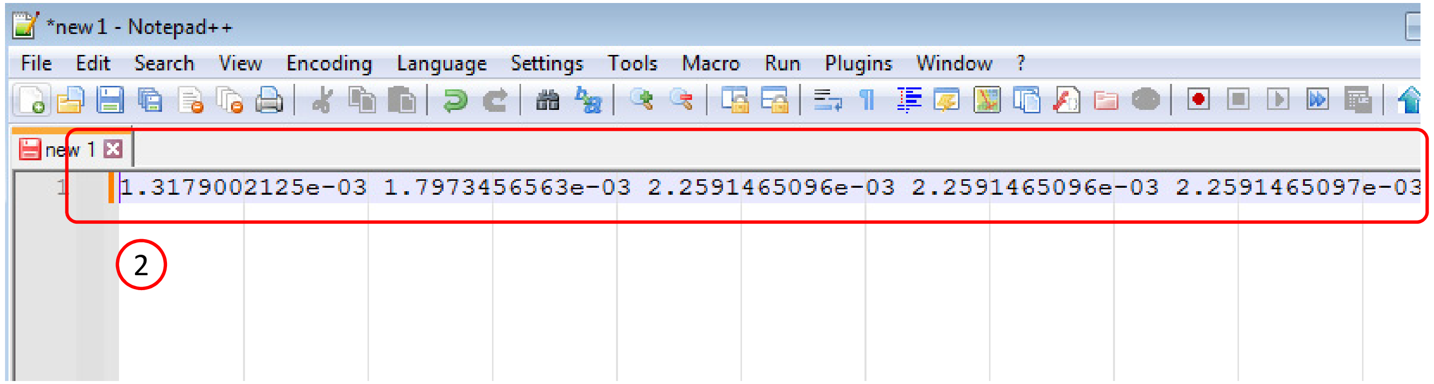
5. Open a new Notepad++ window and paste the selected values to the new window.



Update the Initial Values for the Variables

1. In the new window select all the values with the mouse or with CTRL+A
2. Press CTRL+J to join all the lines





Update the Initial Values for the Variables

1. Add the keyword *initial_point* to the beginning of the line and include a space. Example:

   ```
   initial_point 1.3179002125e-03
   […]
   ```

2. Open file nastran_working_directory/workspace_d/study_d.in

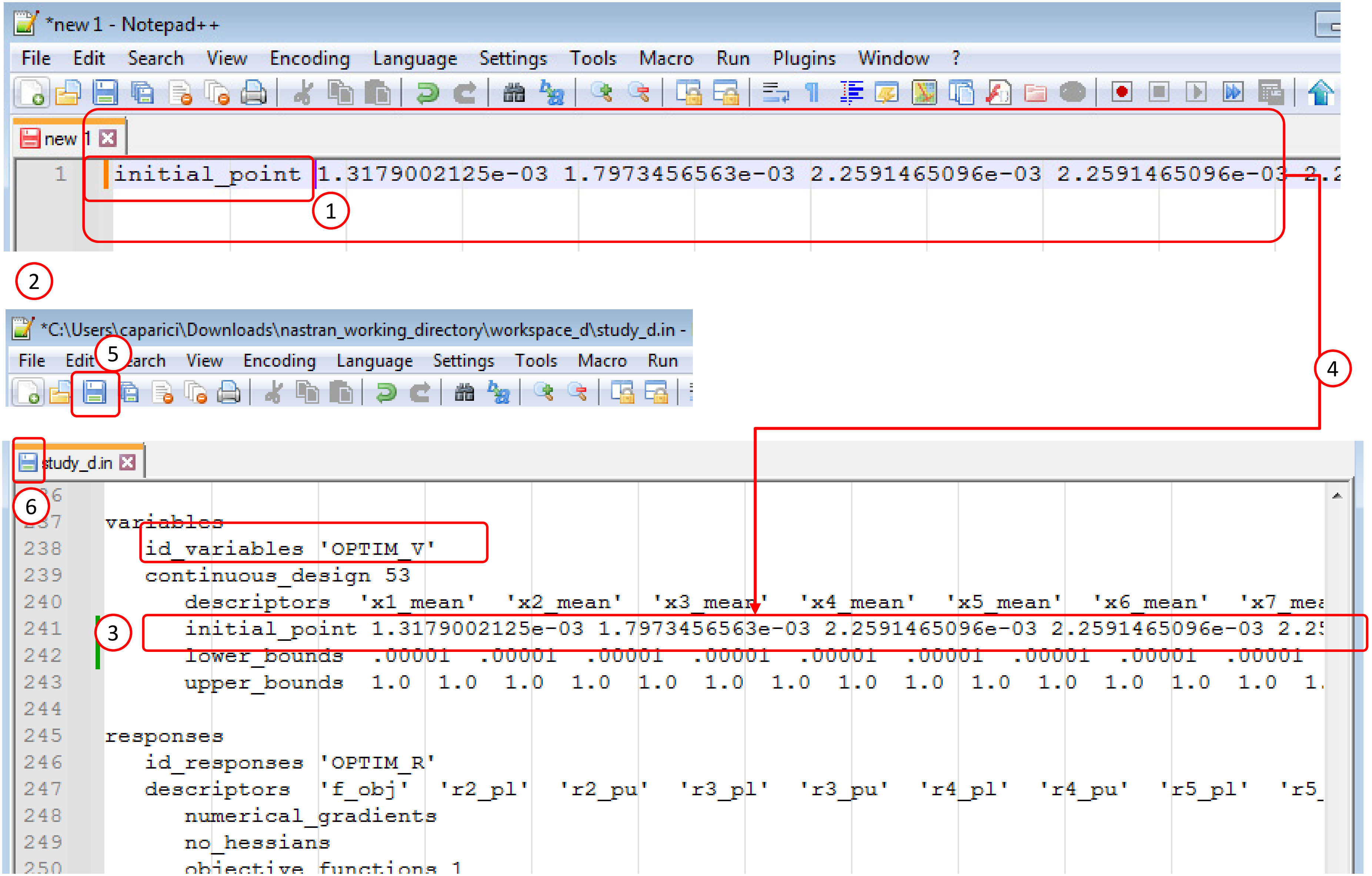
3. Search for the keywordblock with id_variables='OPTIM_V'

4. Copy the new line and replace the old line found in study_d.in

5. Click save

6. Ensure the save icon is blue, not red. A blue color indicates the file has been saved. A red icon indicates the file has not been saved.

The initial values of the variables has been updated.



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

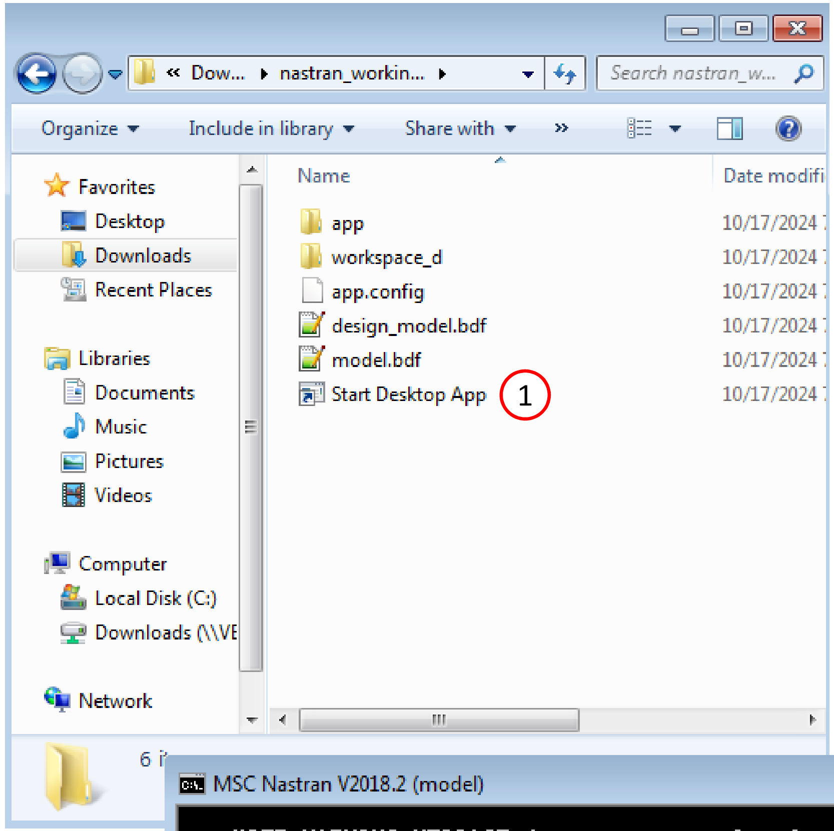
- One can run the Nastran job on a remote machine as follows:
  1) Copy the BDF files and the INCLUDE files to a remote machine. 2) Run the MSC Nastran job on the remote machine. 3) After completion, copy the BDF, F06, LOG, H5 files to the local machine. 4) Click "Start Desktop App" to display the results.

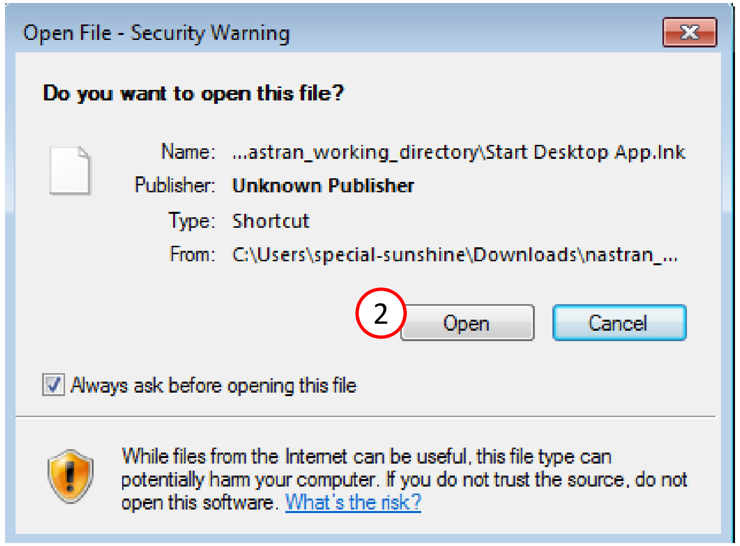
Using Linux?

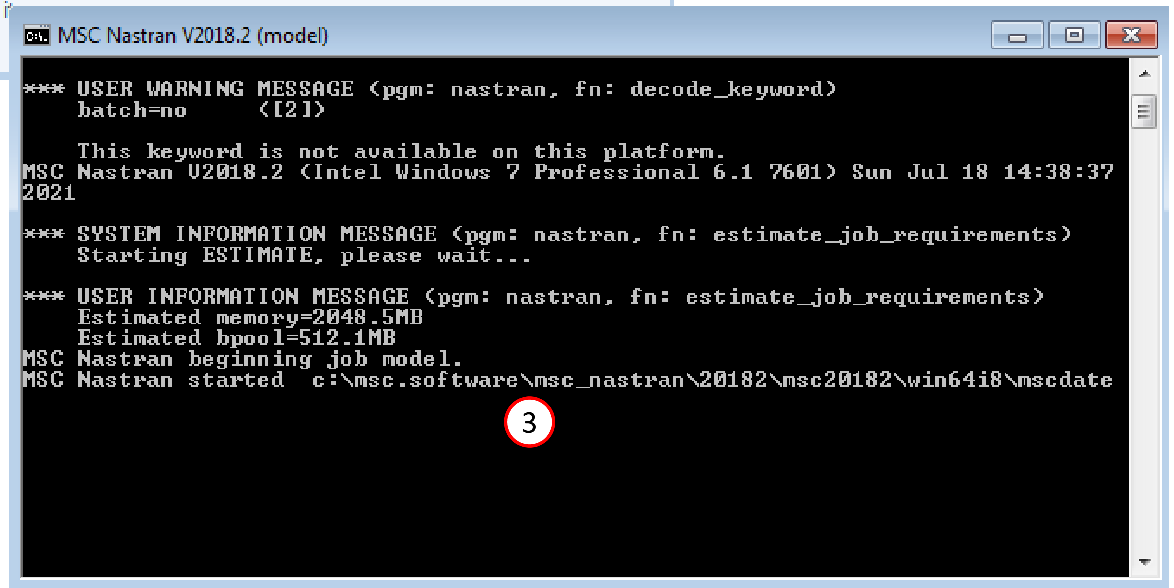
Follow these instructions:
1) Open Terminal
2) Navigate to the nastran_working_directory
   cd ./nastran_working_directory
3) Use this command to start the process
   ./Start_MSC_Nastran.sh

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

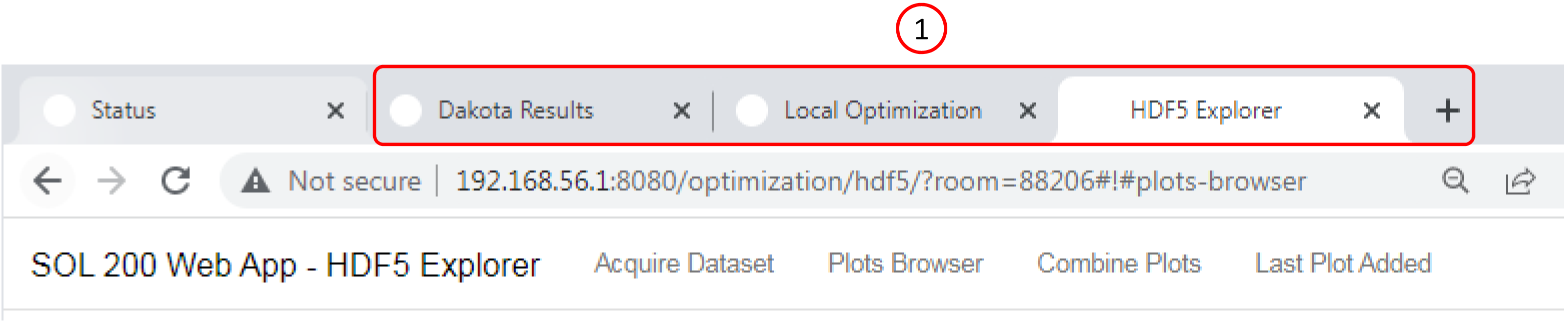
Python MSC Nastran

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 200 (200 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 200 runs are complete and zero optimization iterations are performed. Recall the goal is to just run the optimization procedure to calculate the constraint values.

1

2

```
------------------------------
Begin     UQ_I Evaluation  200
------------------------------
Parameters for evaluation 200:
                      1.0209139396e-03 x1
                      2.1493266355e-03 x2
                      2.0693108677e-03 x3
                      [...]
blocking fork:
Active response data for UQ_I evaluation 200:
Active set vector = { 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                      1.9927034000e+02 r1
                      9.7951450000e+07 r2
                     -6.9789300000e+07 r3
                     [...]
                     -9.2117390000e+07 r107
Active response data from sub_iterator:
Active set vector = { 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
                      1.8948053825e+02 mean_r1
                      1.0000000000e+00 ccdf_plev1_r2
                      0.0000000000e+00 ccdf_plev2_r2
                      [...]
---------------------------
NestedModel Evaluation    1 results:
---------------------------
Active response data from nested mapping:
Active set vector = { 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                      1.8948053825e+02 f_obj
                      1.0000000000e+00 r2_pl
                      [...]
                      0.0000000000e+00 r107_pu

Iteration terminated: max_function_evaluations limit has been met.
<<<<< Function evaluation summary (UQ_I): 200 total (200 new, 0 duplicate)
<<<<< Best parameters          =
                      1.3179002125e-03 x1_mean
                      1.7973456563e-03 x2_mean
                      2.2591465096e-03 x3_mean
                      [...]
                      1.0674418472e-03 x53_mean
<<<<< Best objective function  =
                      1.8948053825e+02
<<<<< Best constraint values   =
                      1.0000000000e+00
                      [...]
```

3

Results

1. The OUU was configured to constraint reliability indices. This verification was configured such that the constraints are for probabilities of failure. The reported values are probabilities of failure.

```
------------------------------
Begin     UQ_I Evaluation  200
------------------------------
Parameters for evaluation 200:
                      1.0209139396e-03 x1
                      2.1493266355e-03 x2
                      2.0693108677e-03 x3
                      [...]
blocking fork:
Active response data for UQ_I evaluation 200:
Active set vector = { 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                      1.9927034000e+02 r1
                      9.7951450000e+07 r2
                     -6.9789300000e+07 r3
                     [...]
                     -9.2117390000e+07 r107
Active response data from sub_iterator:
Active set vector = { 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
                      1.8948053825e+02 mean_r1
                      1.0000000000e+00 ccdf_plev1_r2
                      0.0000000000e+00 ccdf_plev2_r2
                      [...]
---------------------------
NestedModel Evaluation    1 results:
---------------------------
Active response data from nested mapping:
Active set vector = { 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                      1.8948053825e+02 f_obj
                      1.0000000000e+00 r2_pl
                      [...]
                      0.0000000000e+00 r107_pu

Iteration terminated: max_function_evaluations limit has been met.
<<<<< Function evaluation summary (UQ_I): 200 total (200 new, 0 duplicate)
<<<<< Best parameters          =
                      1.3179002125e-03 x1_mean
                      1.7973456563e-03 x2_mean
                      2.2591465096e-03 x3_mean
                      [...]
                      1.0674418472e-03 x53_mean
<<<<< Best objective function  =
                      1.8948053825e+02
<<<<< Best constraint values   =
                      1.0000000000e+00
                      [...]
```

1

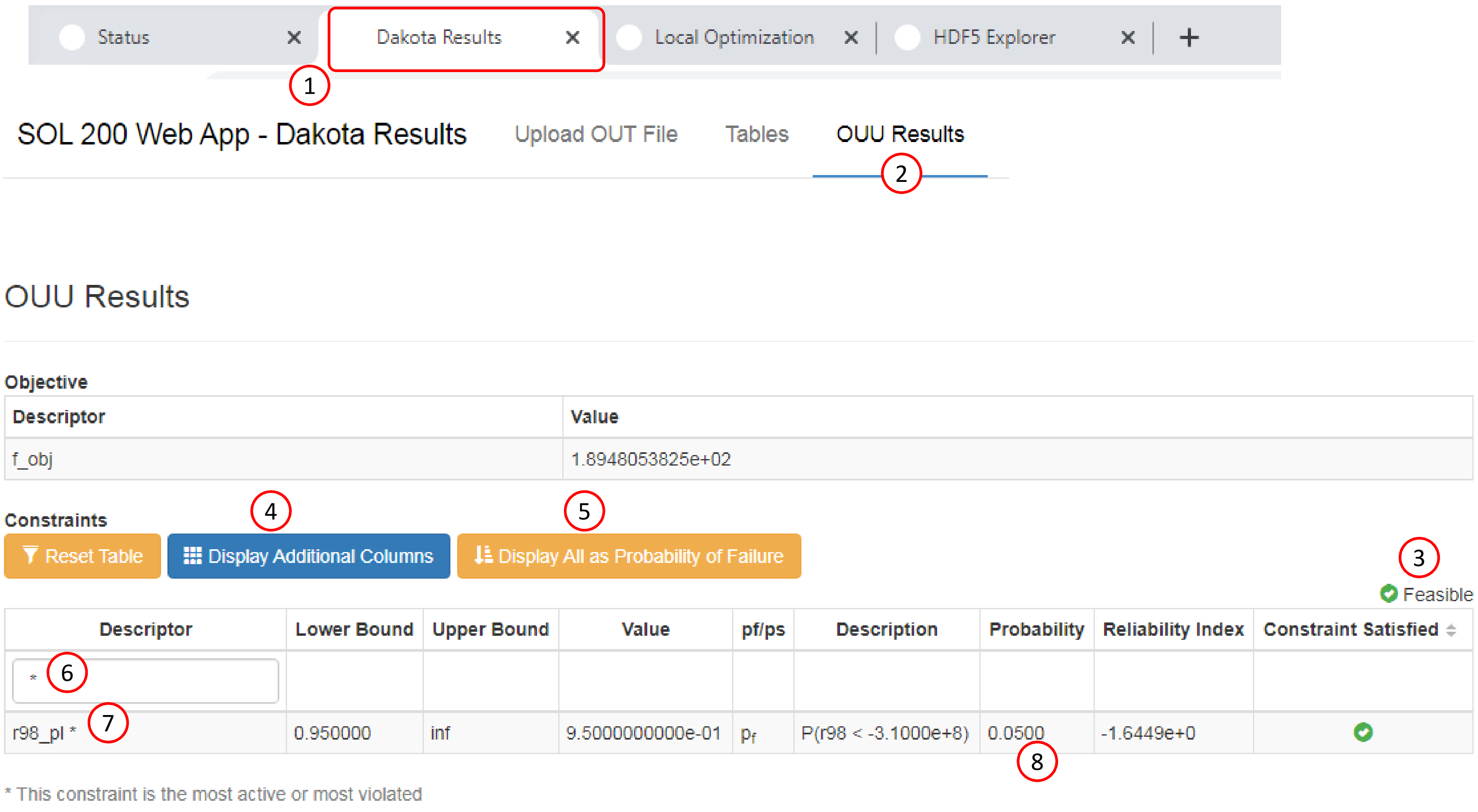
Discussion of Final Probabilities of Failure

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

1. Select the Dakota Results tab or window
2. Click OUU Results
3. Notice the final design is deemed feasible and all the individual constraints are satisfied
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. In the indicated search bar, search for character *
7. The search reveals the following constraints are the most active or most violated
8. Note the probability of failure is 5%, which is on the limit of 5% specified in this exercise.

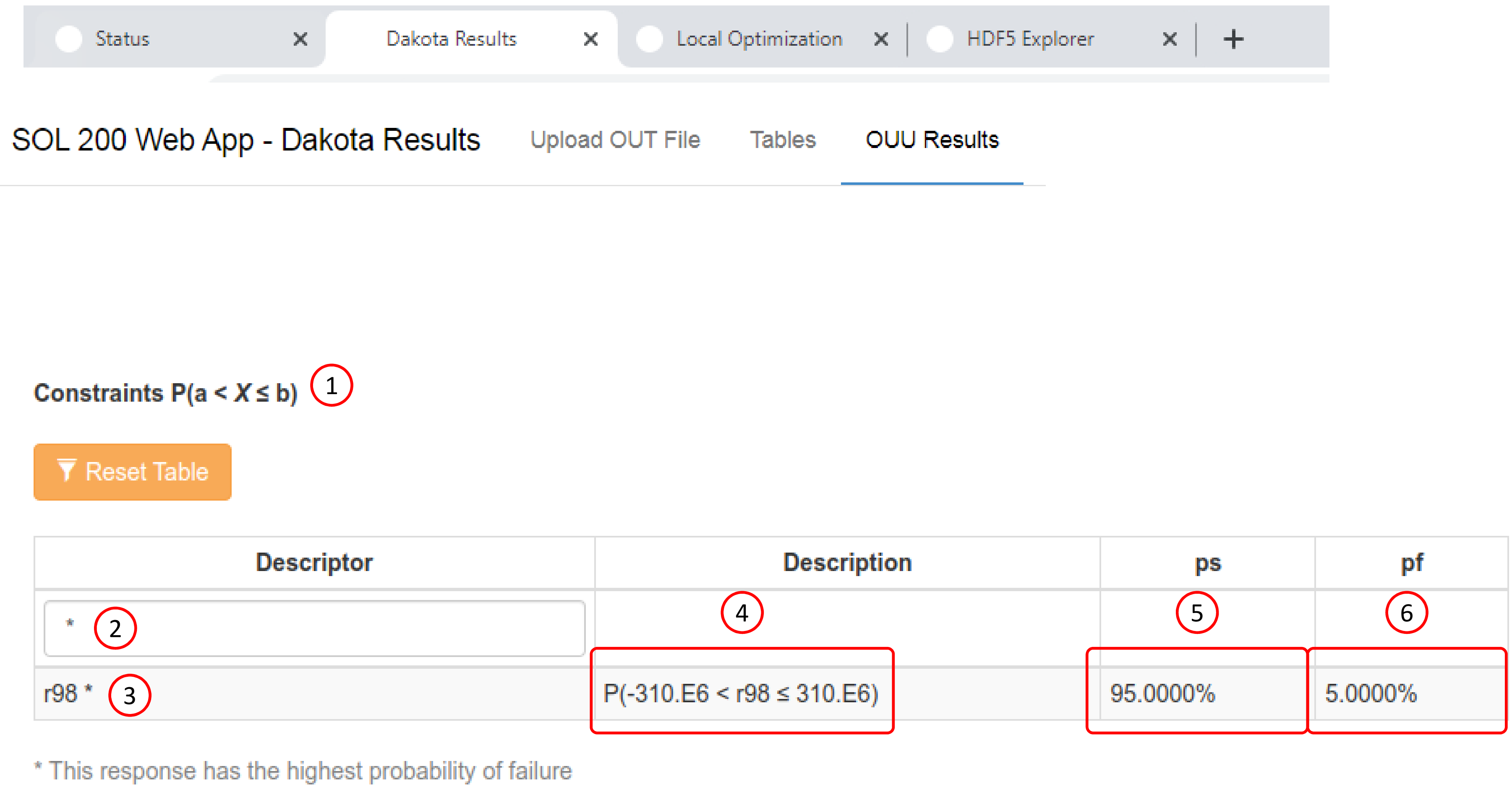


Discussion of Final Probabilities of Failure

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

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

The highest probability of failure is 5%.



Descriptor	Description	ps	pf
*			
r98 *	P(-310.E6 < r98 ≤ 310.E6)	95.0000%	5.0000%

Comparison of Approximate and Actual p_f

- It is seen the actual and worst case p_f of 5% is on the edge of the desired 5% p_f.
- The OUU in part A and the verification of probabilities in part B has been a success.

Response	Part A - OUU Approximated Reliability Index (Equivalent Probability of Failure)	Part B - UQ Generated (Actual Probability of failure)
Comments	The OUU considered and reported reliability indices	These are the probabilities after an LHS of size 200
Maximum Probability of Failure	1.6716146649e+00 (4.73%) for responses r55_pu, r63_pu, r67_pu, r78_pu	(5%) for response r98_pl

Comparison of Approximate and Actual p_f

This exercise used a 3% constraint on p_f. If one is more daring, a 4% constraint may be used to yield a more optimal solution. Just note that the final probabilities should be verified with an LHS of size N.

Configure UQ Responses

Statistics to compute at each response level

Probabilities

Delete	Descriptor	Status	Response Levels	Probability of Failure for Lower Bound [%]	Lower Bound	Upper Bound	Probability of Failure for Upper Bound [%]	Description
	r1							
	r2		-20000 2000(	4.0	-20000	20000	4.0	
	r3		-20000 2000(	4.0	-20000	20000	4.0	

1

End of Tutorial

Appendix

Appendix Contents

- 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

Interpreting the Dakota Input File

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

study_d.in

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


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

Interpreting the Dakota Input File

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

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

Interpreting the Dakota Input File

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

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

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

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

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

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

```
method
   id_method 'UQ'
2  sampling
      model_pointer 'UQ_M'
      distribution
            complementary      3                              4
      response_levels  -20000  20000  -20000  20000
         num_response_levels  0  2  2
      sample_type                                             5
            lhs             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.

1

```
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
```

2

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

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

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

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.

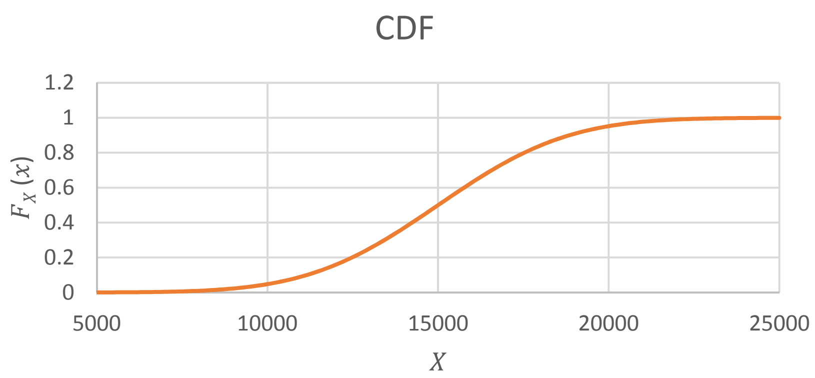
It must be decided if CDF or CCDF values are used throughout the UQ or OUU.

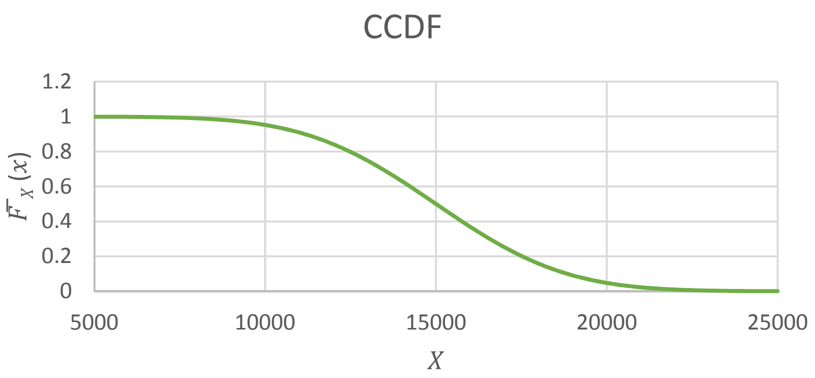
The CDF and CCDF are related by the following relationships

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

$$p_s = \mathrm{P}(X \leq 20000).$$

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

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

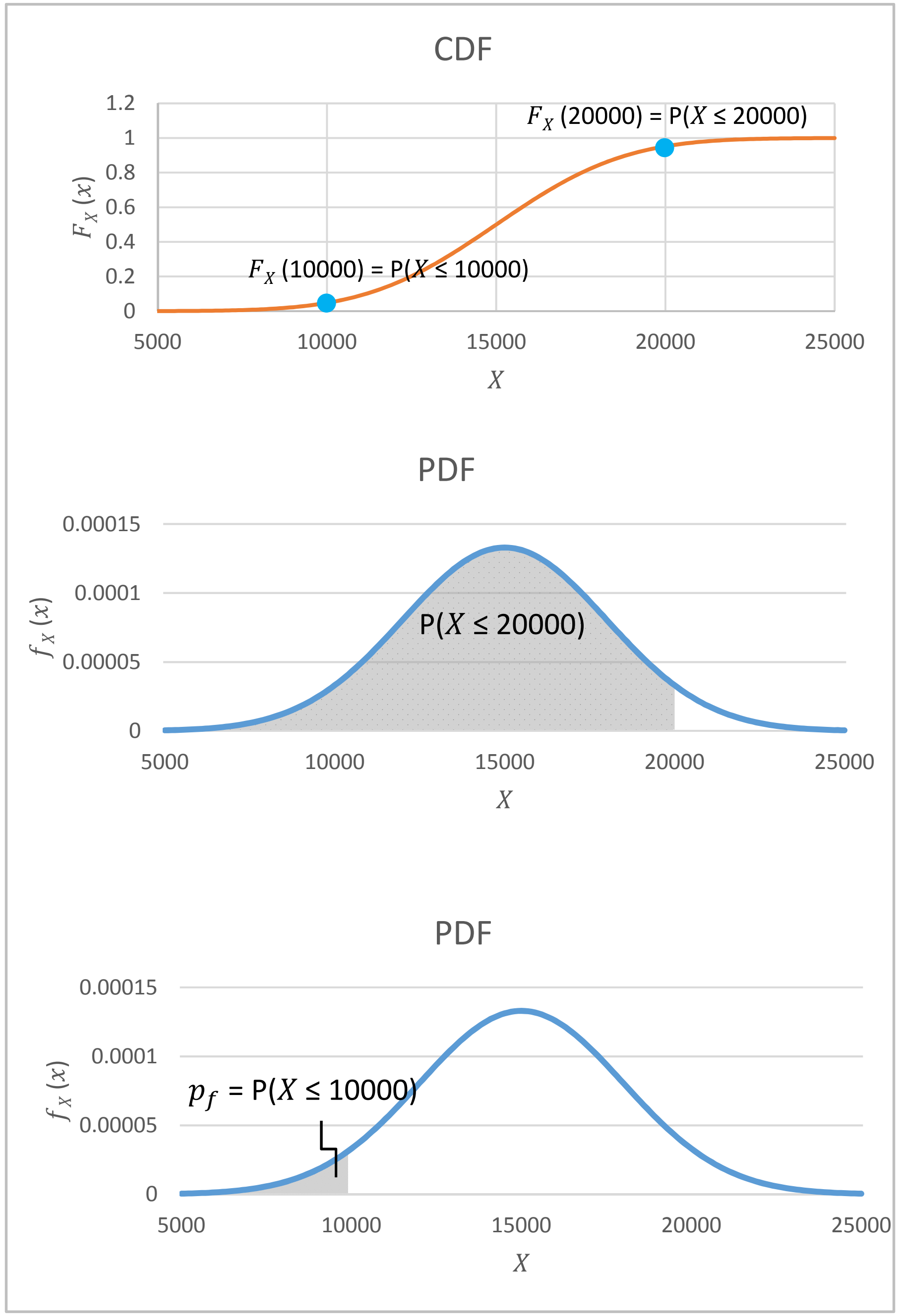
- For the lower bound, if CDF values are used, the probability of failure is

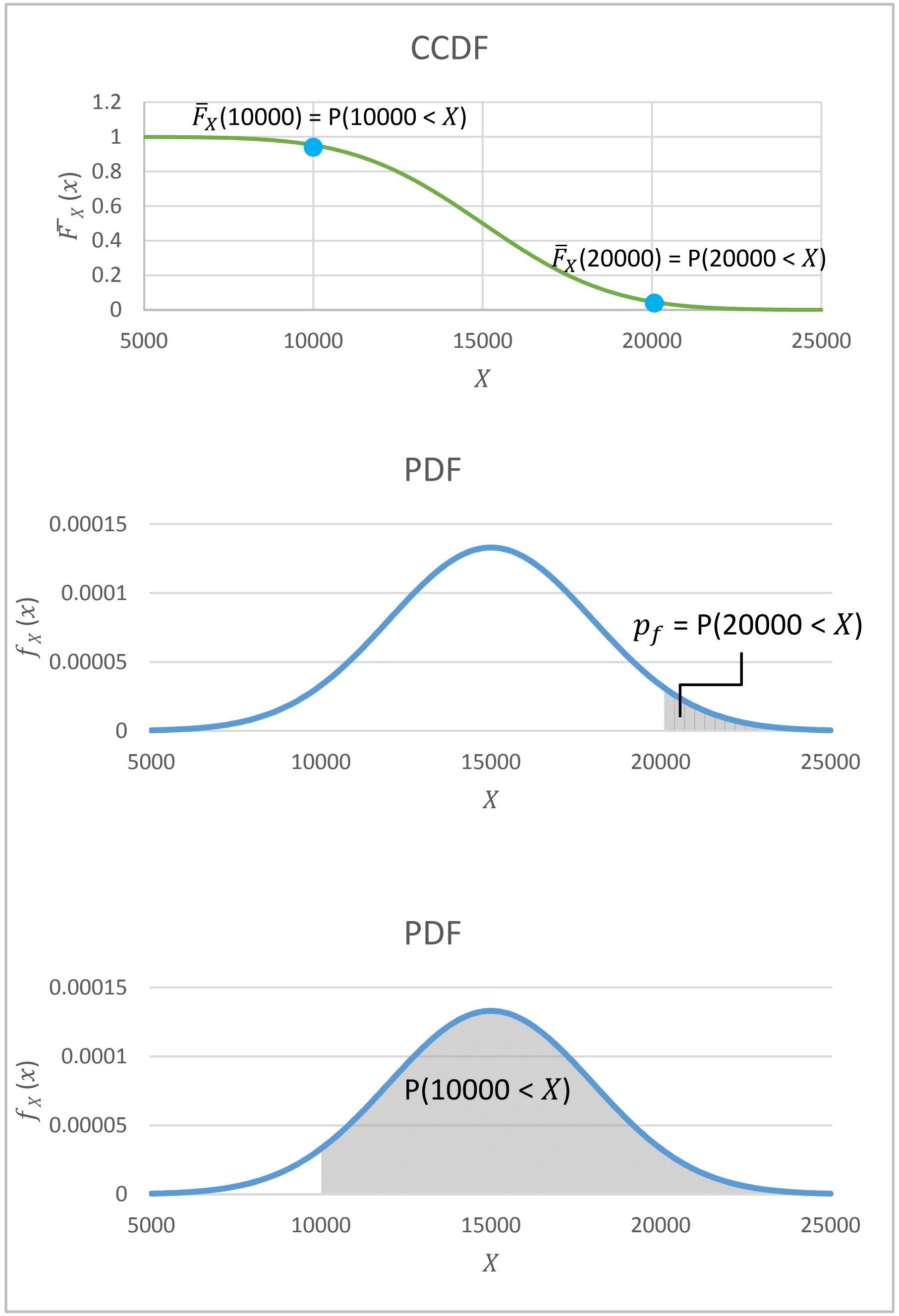
$$p_f = \mathrm{P}(X \leq 10000).$$

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

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

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





Dakota Output

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

Dakota Input File

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

Dakota Output File

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

p_f = P($X \le 10000$)

p_s = P($X \le 20000$)

Dakota Input File

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

Dakota Output File

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

p_s = P($10000 < X$)

p_f = P($20000 < X$)

CDF and CCDF Values

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

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

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

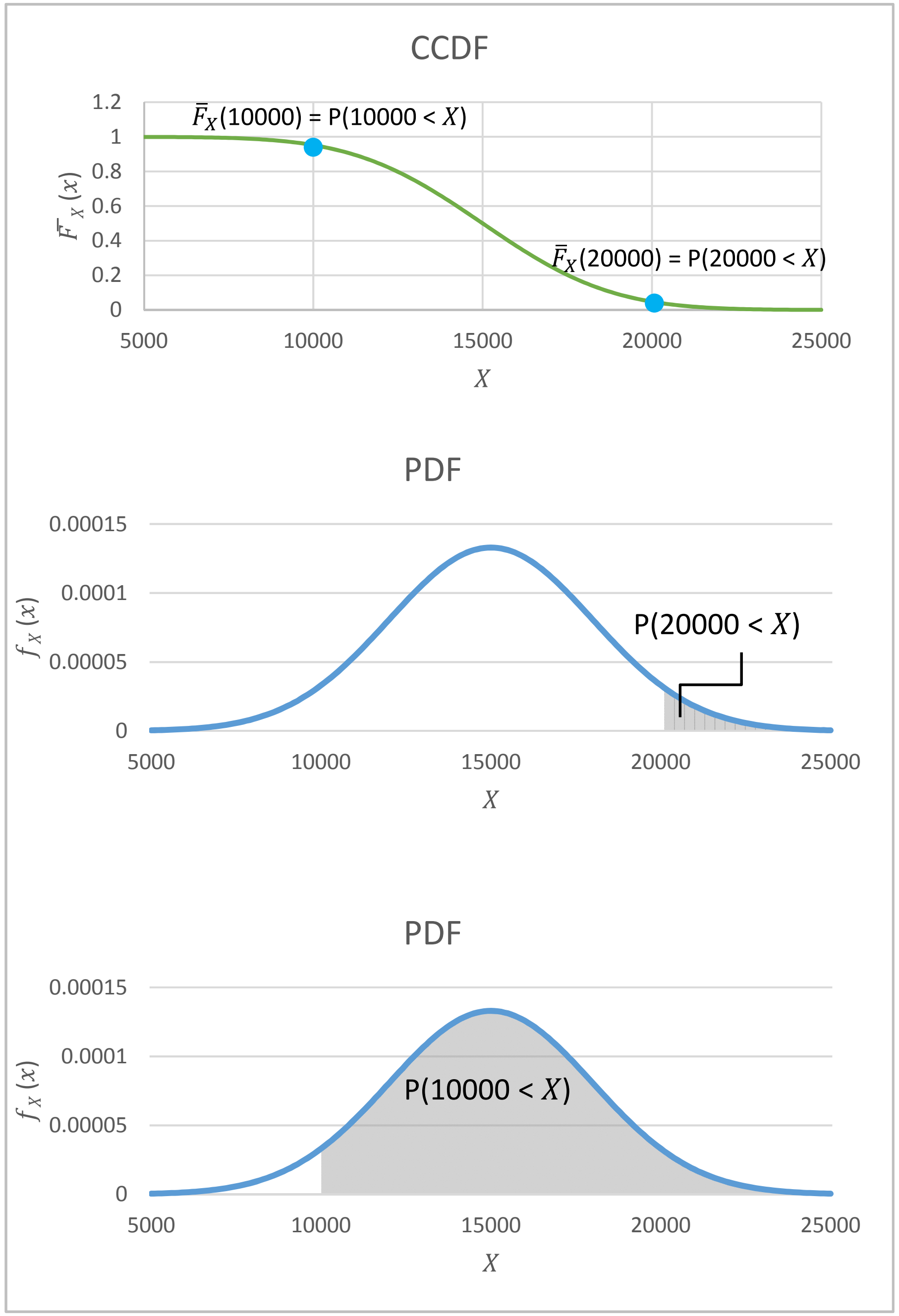
$$p_f = \mathrm{P}(20000 < X) < 0.05$$

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

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

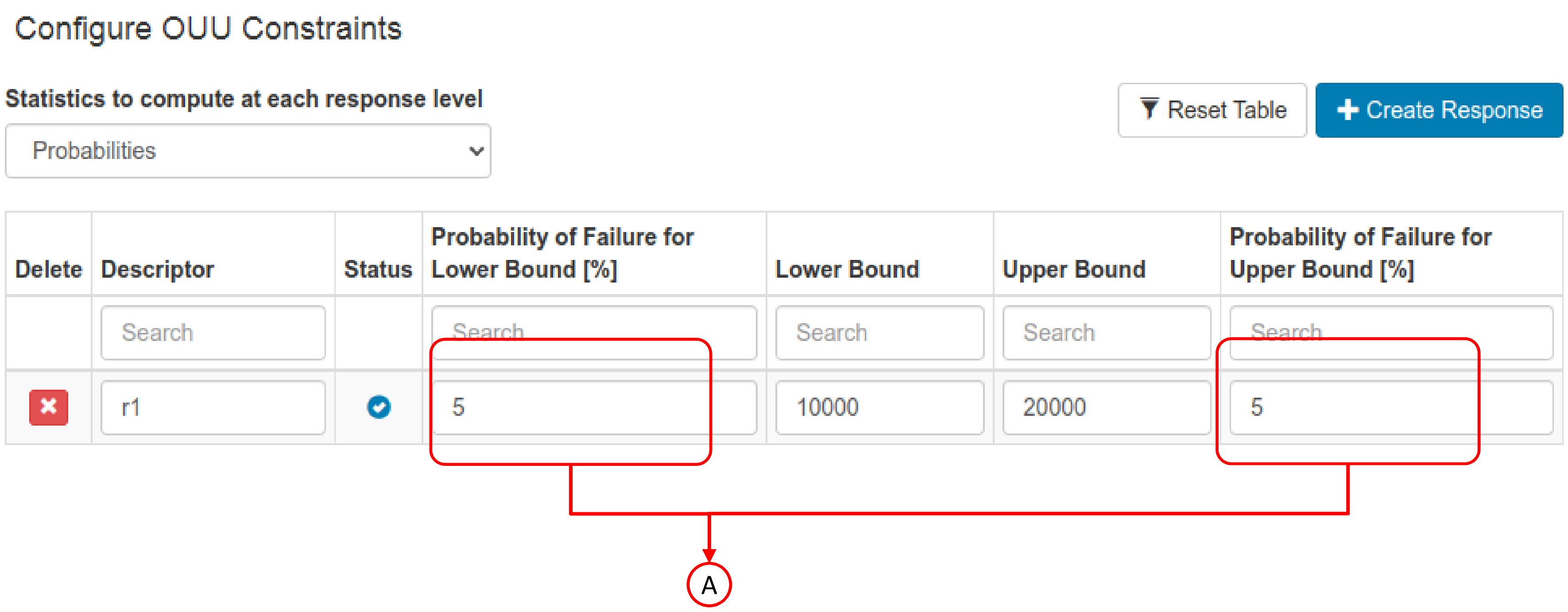
Dakota Input File

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



CDF and CCDF Values

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



CDF and CCDF Values

Assume probabilities have been selected and constrained.

Configure UQ Responses

Statistics to compute at each response level

Probabilities

- Probabilities
- Reliabilities
- Generalized Reliabilities

Let

r1_pl = P(10000 < r2)

r1_pu = P(20000 < r2)

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

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

Configure OUU Objective and Additional Constraints A

Create Constraint

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

CDF and CCDF Values

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

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

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

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

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.

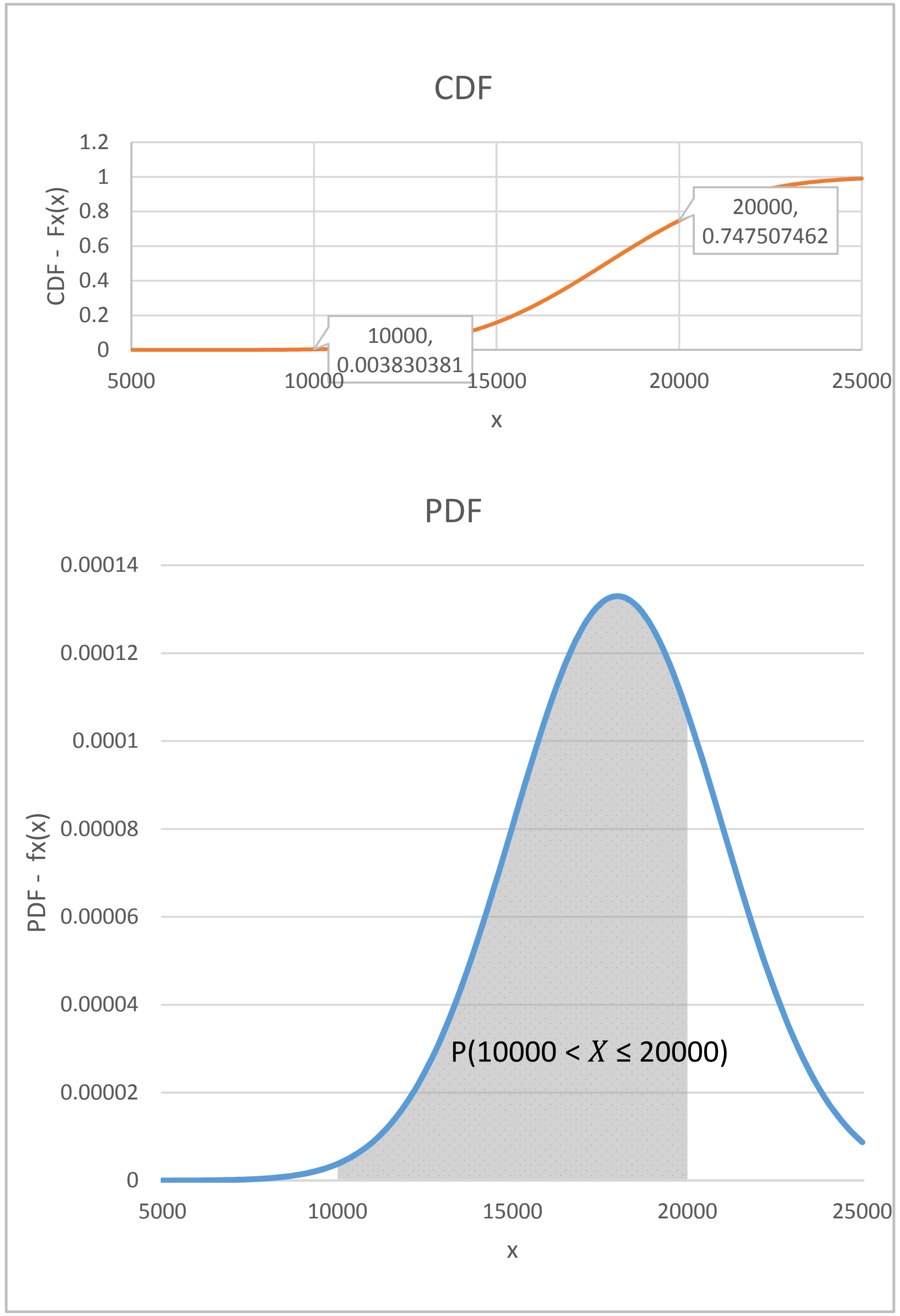
For example, suppose the following single probability is used: $P(10000 < X \le 20000) = 0.74$ (74% survival). Since this single probability is less than the desired 95%, failure is expected. With a single probability, it is not known if the distribution is violating the lower or upper bound.

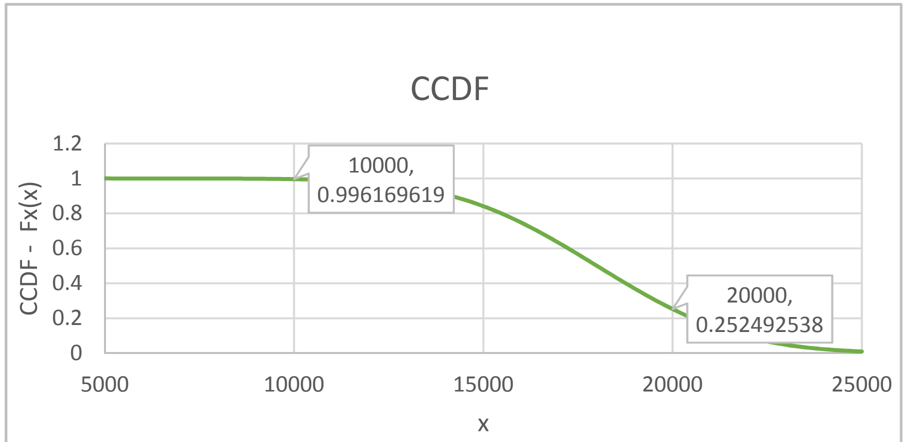
If separate probabilities are constrained, one for the lower and upper bounds, it makes it simpler to identify which of the bounds is being violated.

Consider the distribution shown on the right.

- For the upper bound (20000), the probability of failure is 25.25%. Since the maximum probability of failure is 5%, the probability of failure of the upper bound is violated.

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





CDF and CCDF Values

Final Comments

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

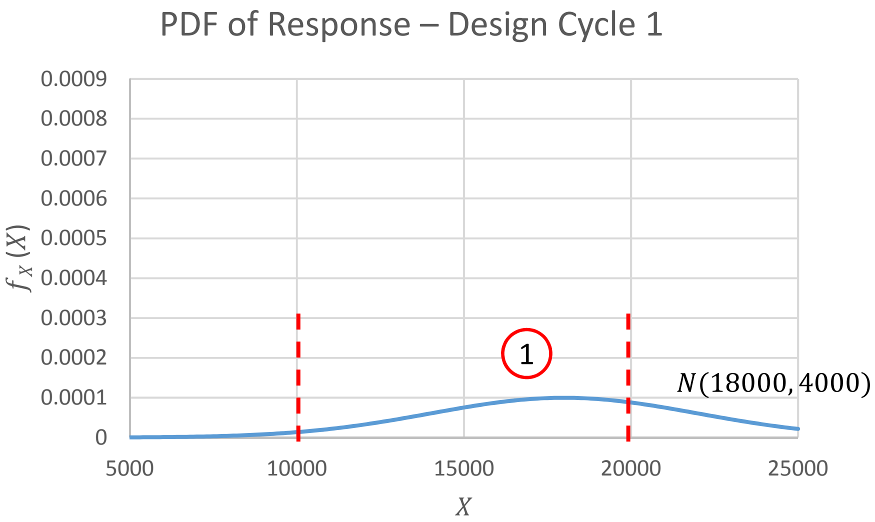
To the right is an example of the distribution of a response during an OUU.

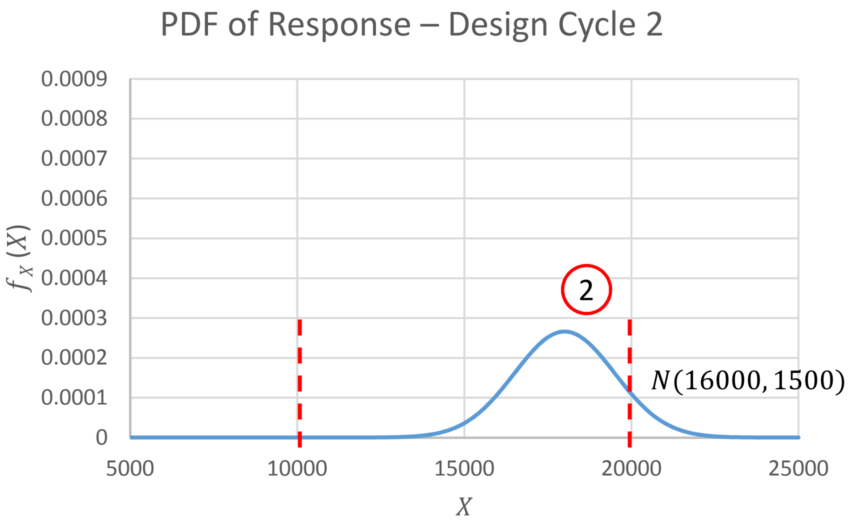
1. The standard deviation is too large and the probabilities of failure for both the lower and upper bound are greater than 5%. The design is infeasible.

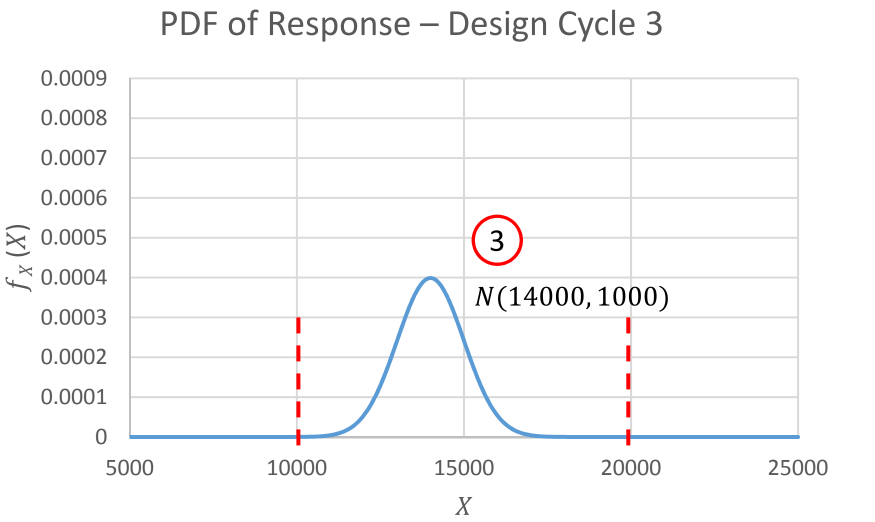
2. The mean has moved far enough to the right such that the probability of failure for the upper bound is greater than 5%. The design is infeasible.

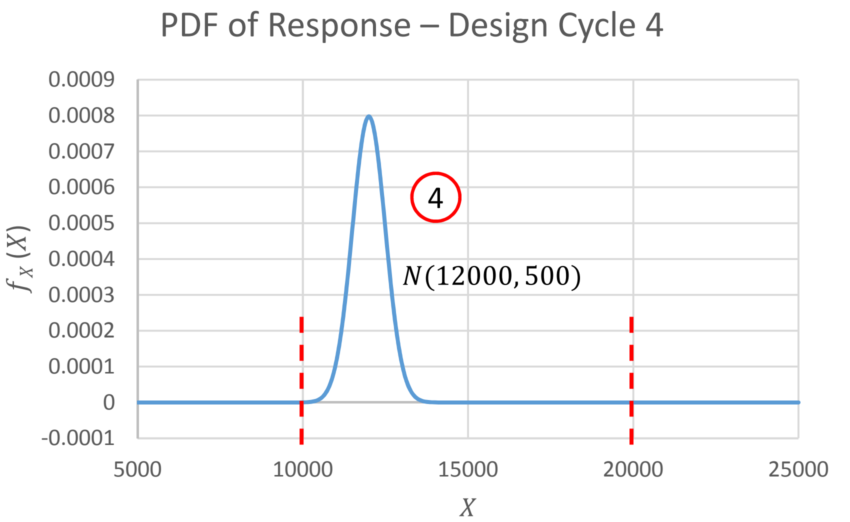
3. The mean is approximately half way between the lower and upper bound and yields a probability of failure within 5% for both lower and upper bounds. The design is feasible.

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





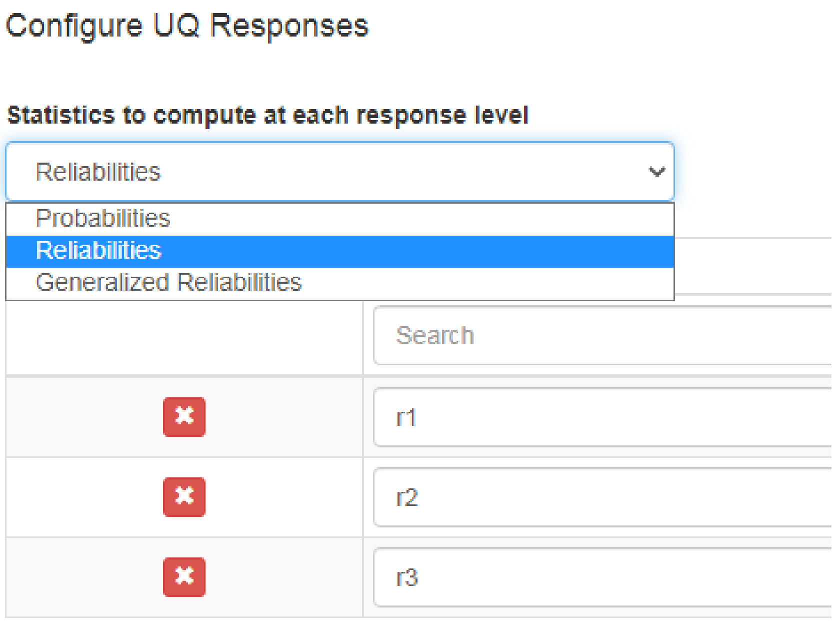




Probabilities, Reliability Index and Generalized Reliability Index

Probabilities, Reliability Index and Generalized Reliability Index

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



What is probability?

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

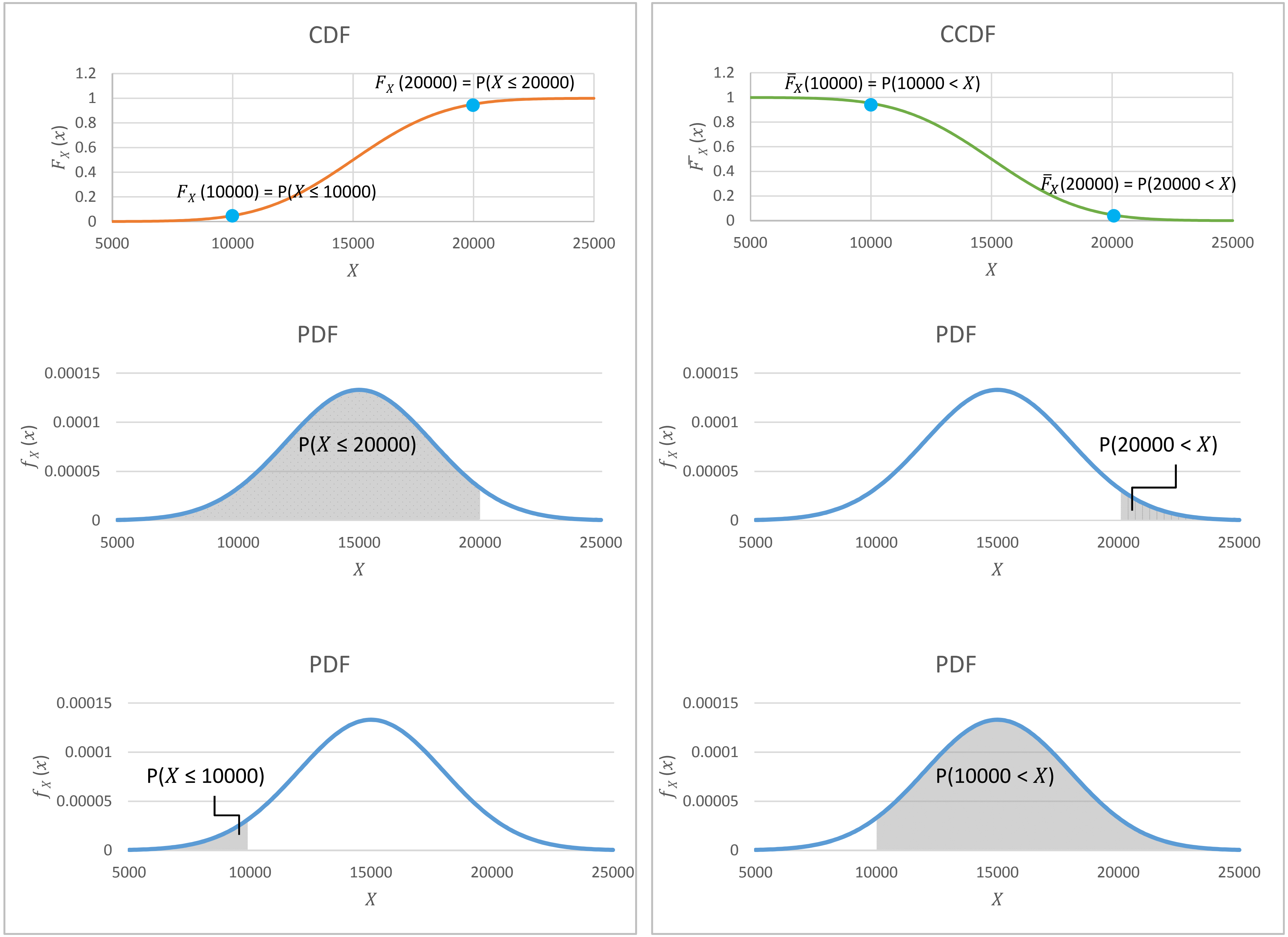
- P($X \leq 20000$)
- P($X \leq 10000$)

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.

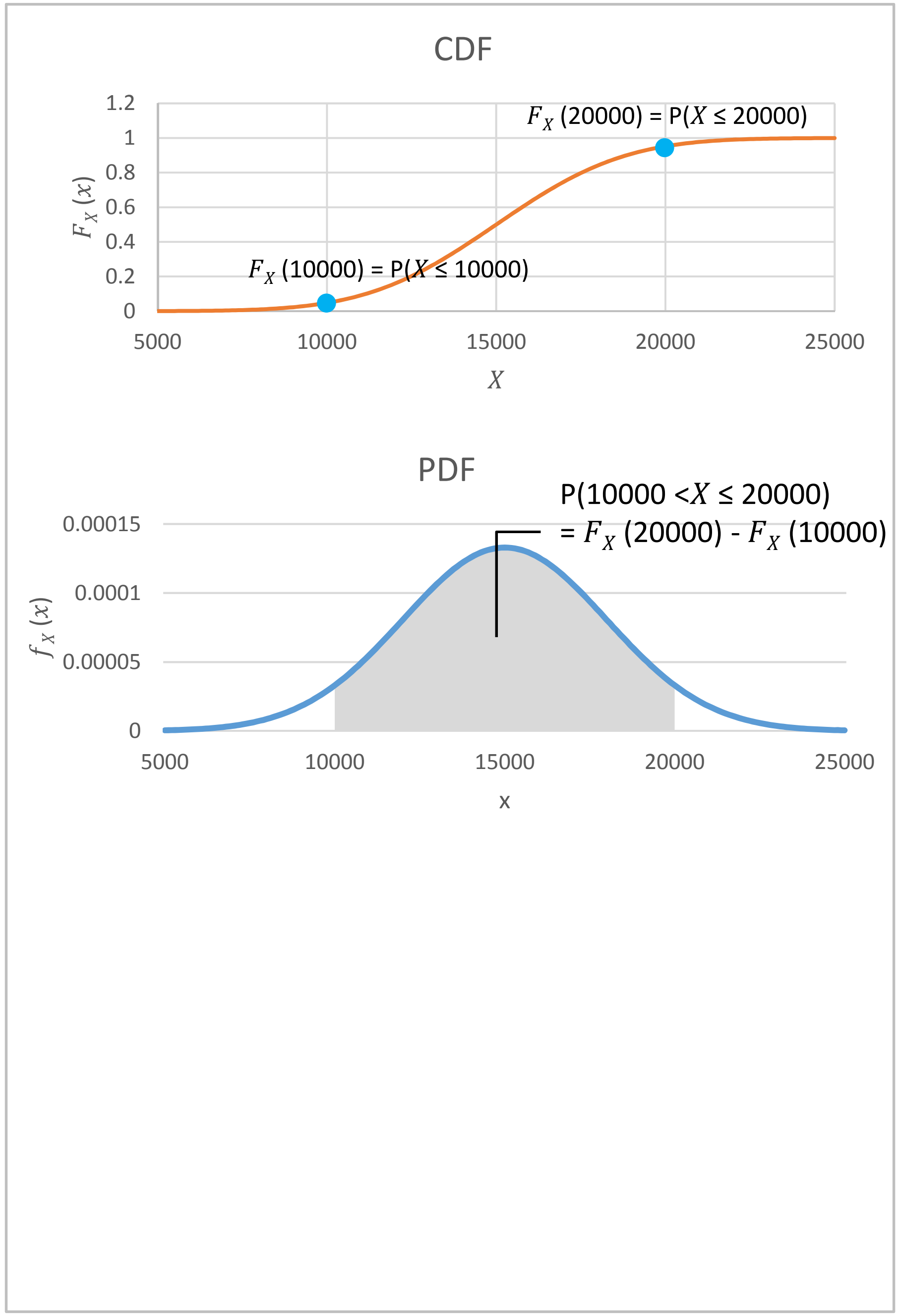
- P(10000 < X ≤ 20000)

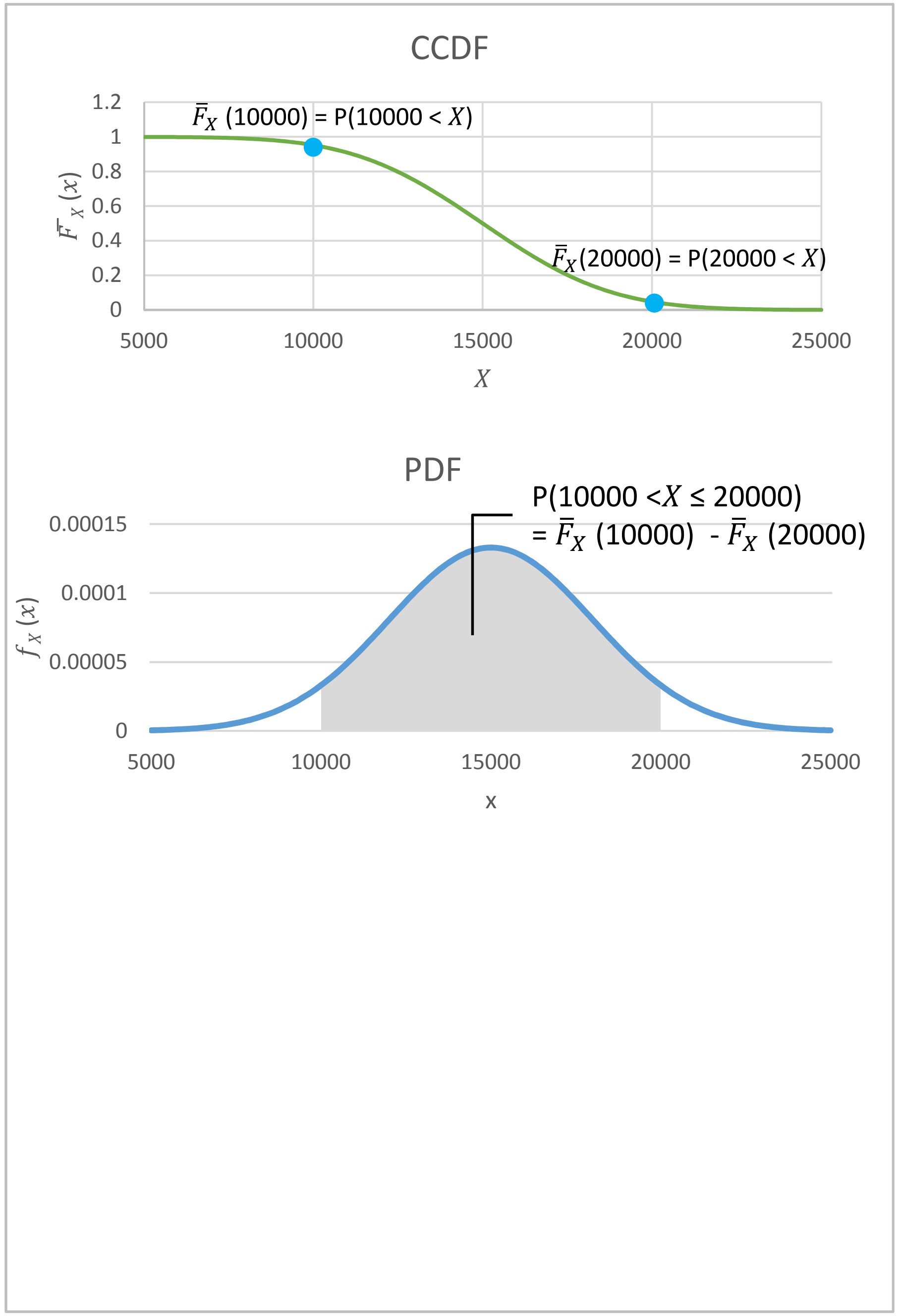
If cumulative distribution function (CDF) values are available, this probability may be determined as follows.

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

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

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

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

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:

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

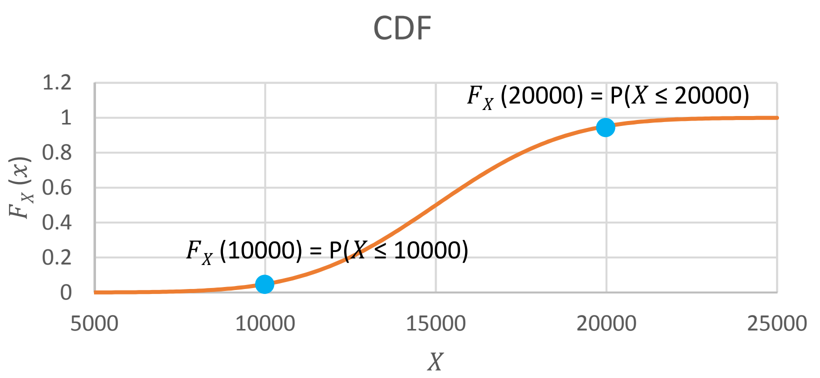
When the CCDF option is used, the probability and reliability index $\bar{\beta}$ are related via the following expression:

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

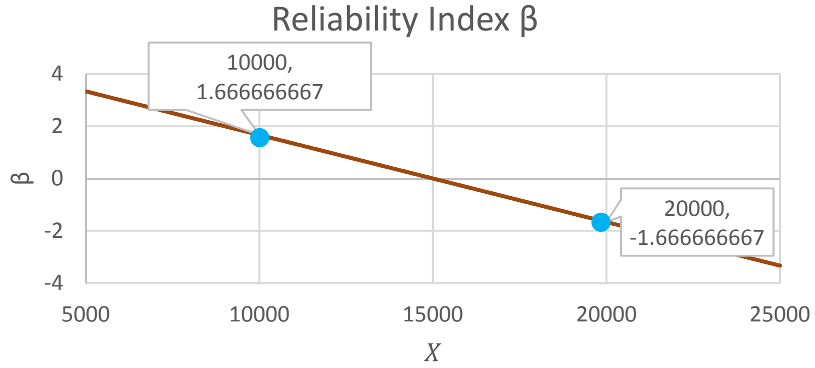
Constraining reliability indices is equivalent to constraining probabilities.

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.

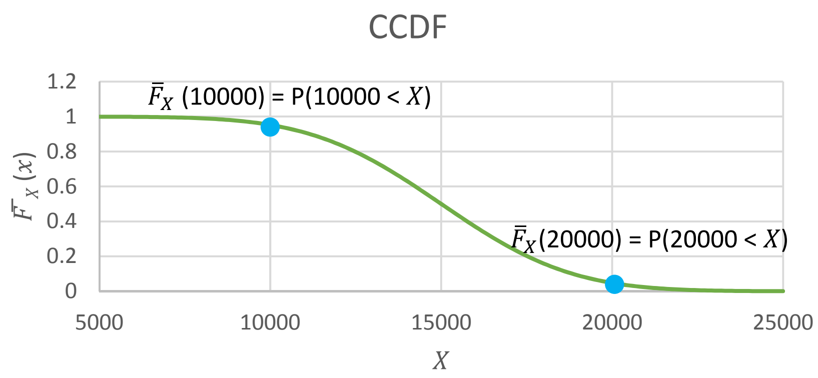


$$p(X \leq x) = F_X(x) = \Phi(-\beta)$$

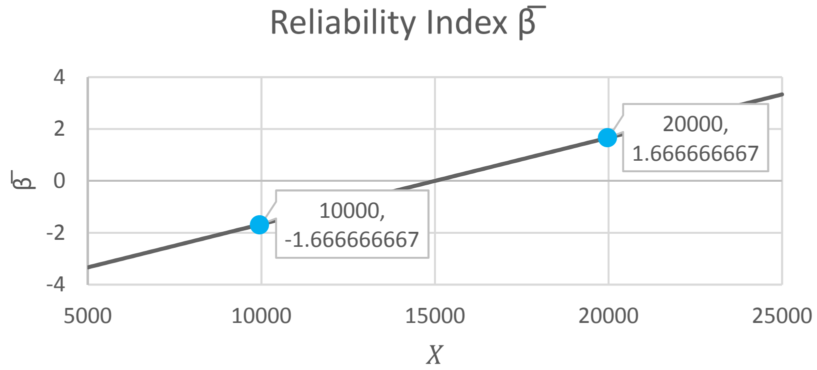


$$\beta = \frac{15000 - X}{3000}$$

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($X \le 10000$) <0.05

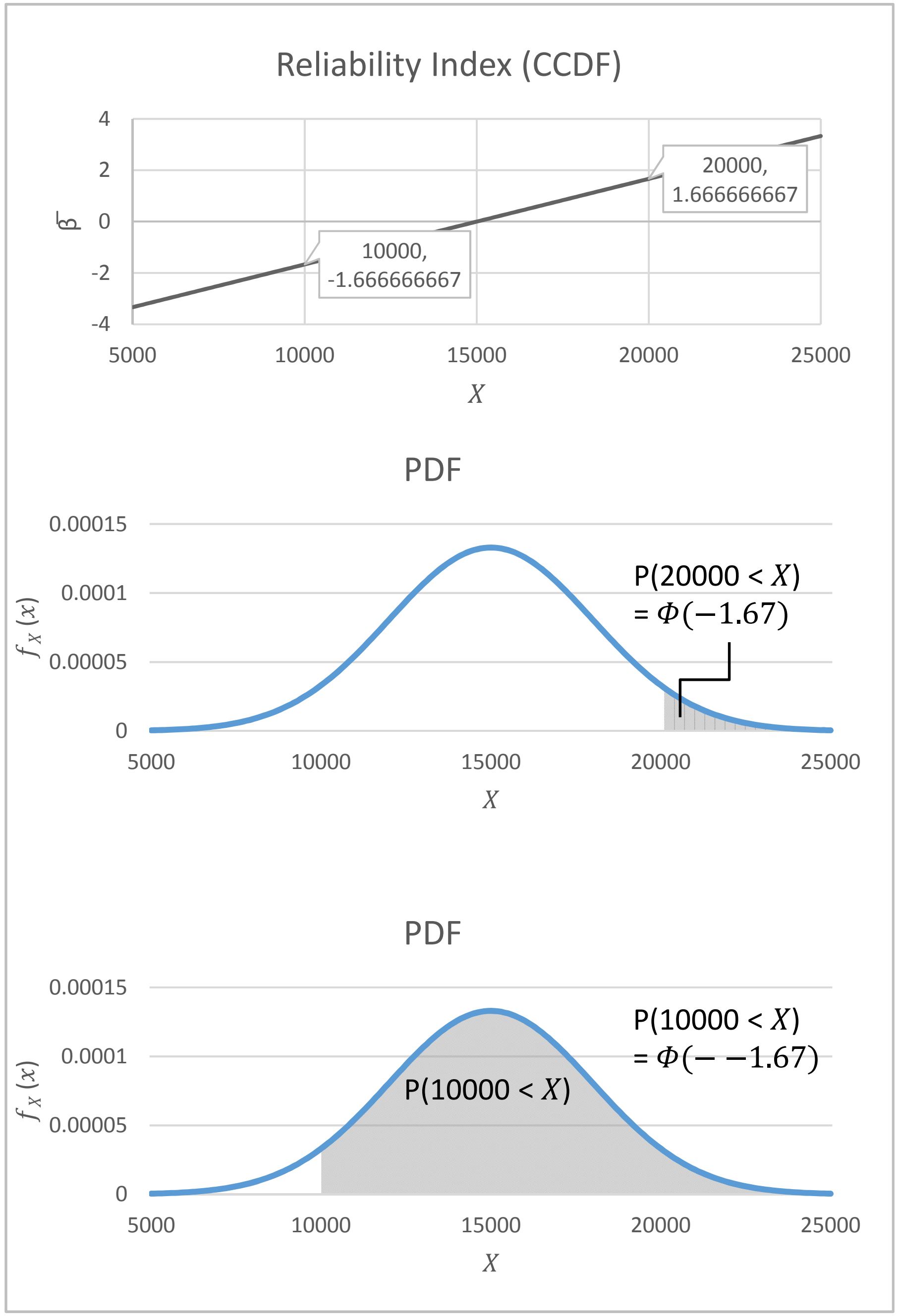
$p_{f,\,upper}$ = P($20000 < X$) < 0.05

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($X \le 10000$) = 1 - P($10000 < X$)	$p_{f,\,upper}$ < 0.05	$\bar{\beta}_{upper} < -1.67$



What is a generalized reliability index?

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

What is a limit state function?

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

What are generalized reliabilities?

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

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

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

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

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

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

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

References

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

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

A

```
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.

3

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


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

1 p_s = P(-20000 < X)

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

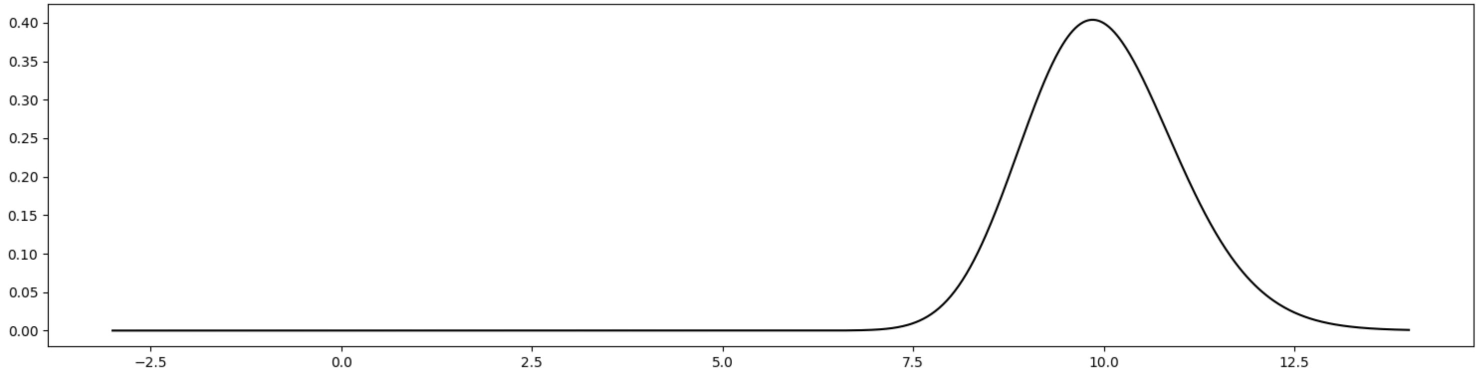
The following applies if uncertain variables have a normal or lognormal distribution.

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.

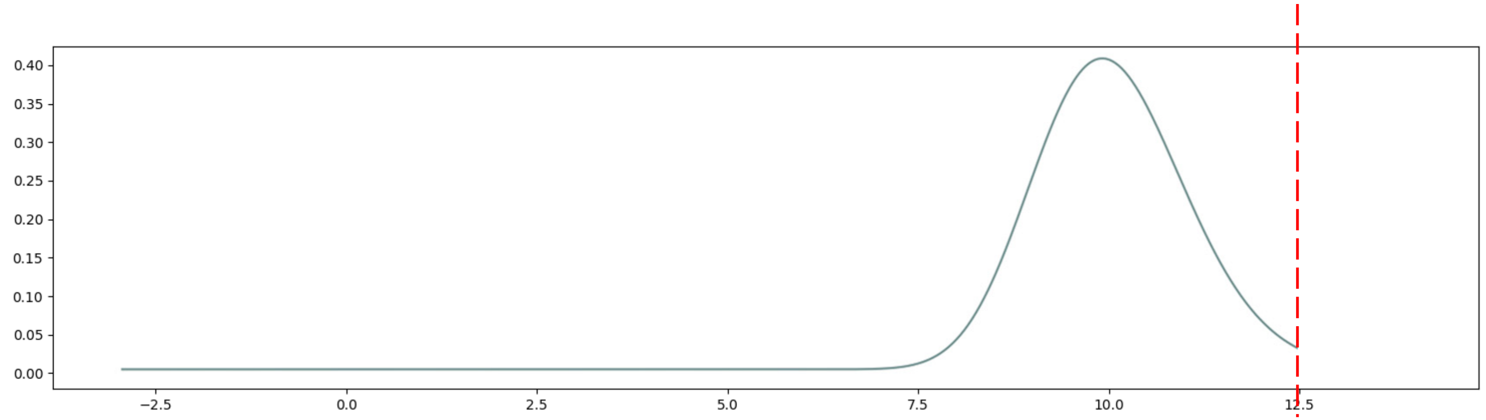
```
File LHS.ERR

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.

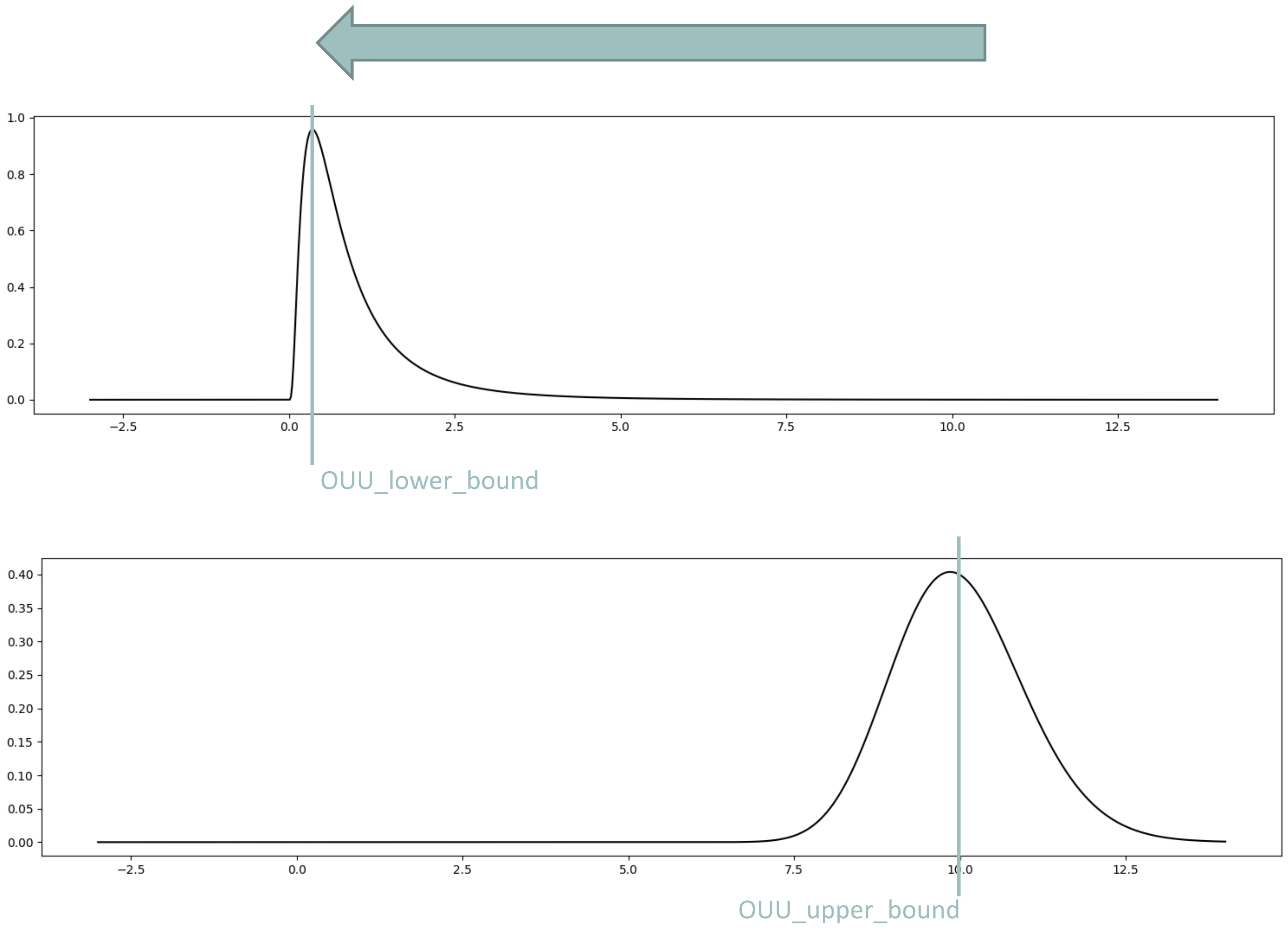


Suppose an upper bound on the distribution was equal to 12.5. No draws or samples will exceed the value of 12.5.

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



Direction of variable’s mean during the optimization.



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

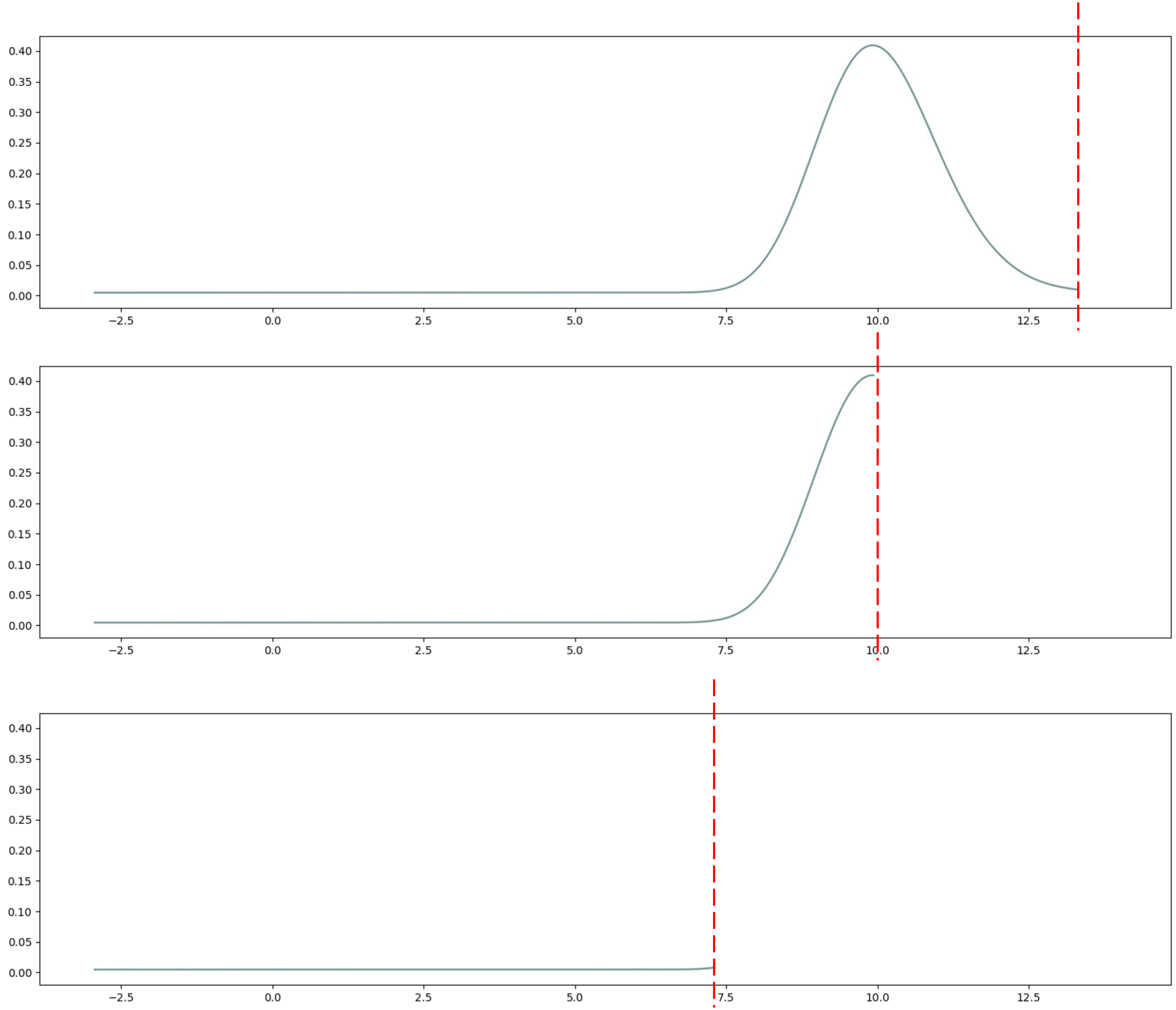
Consequently, the distribution for each variable will change as the mean varies during the optimization.

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

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

Three different UQ upper bounds are displayed.

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



P(xi < UQ_upper_bound) = 99.9%

P(xi < UQ_upper_bound) = 50%

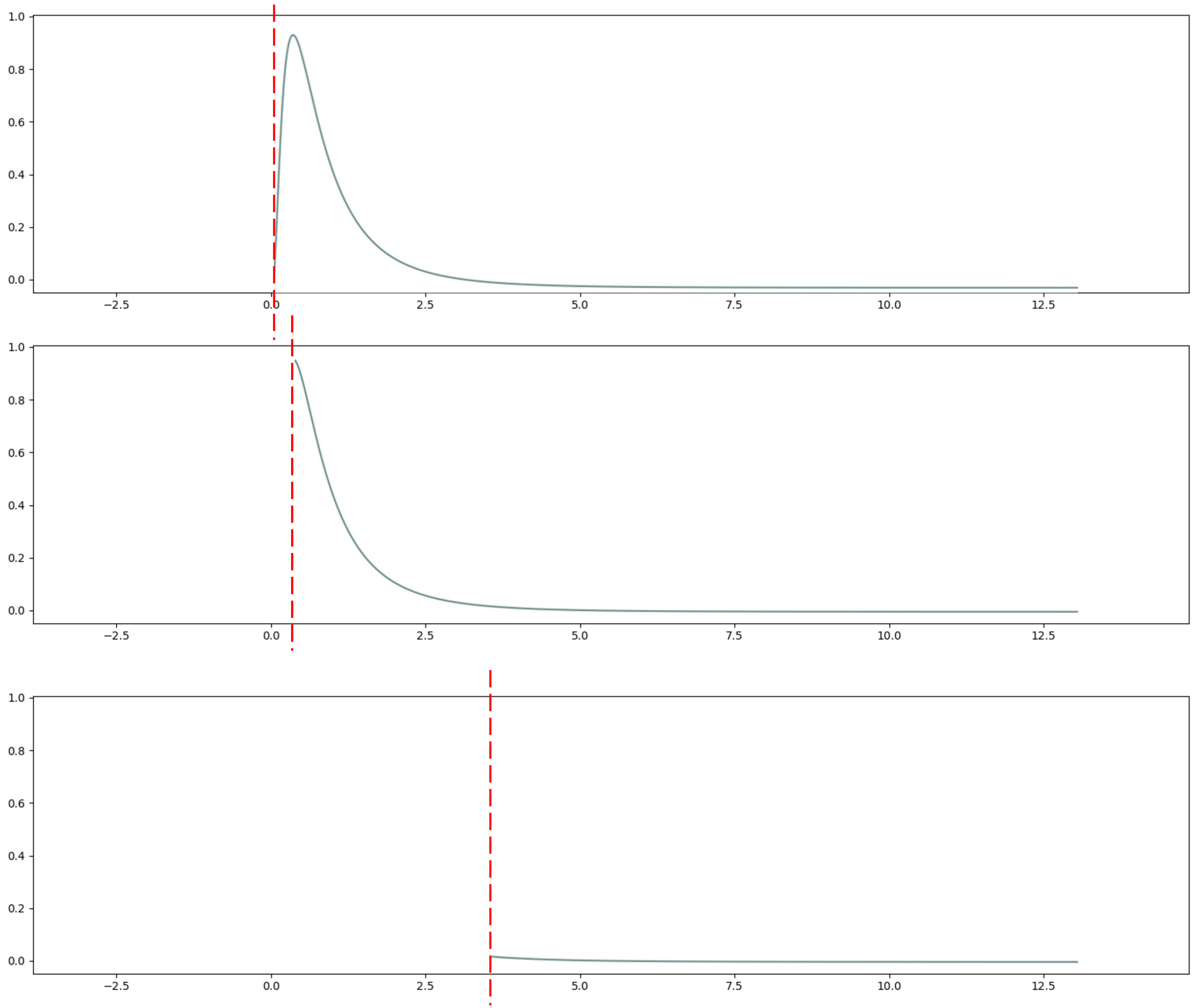
P(xi < UQ_upper_bound) = .1%

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

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

Three different UQ lower bounds are displayed.

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



P(xi > UQ_lower_bound) = 99.9%

P(xi > UQ_lower_bound) = 50%

P(xi > UQ_lower_bound) = .1%

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

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

```
LHS.ERR

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


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

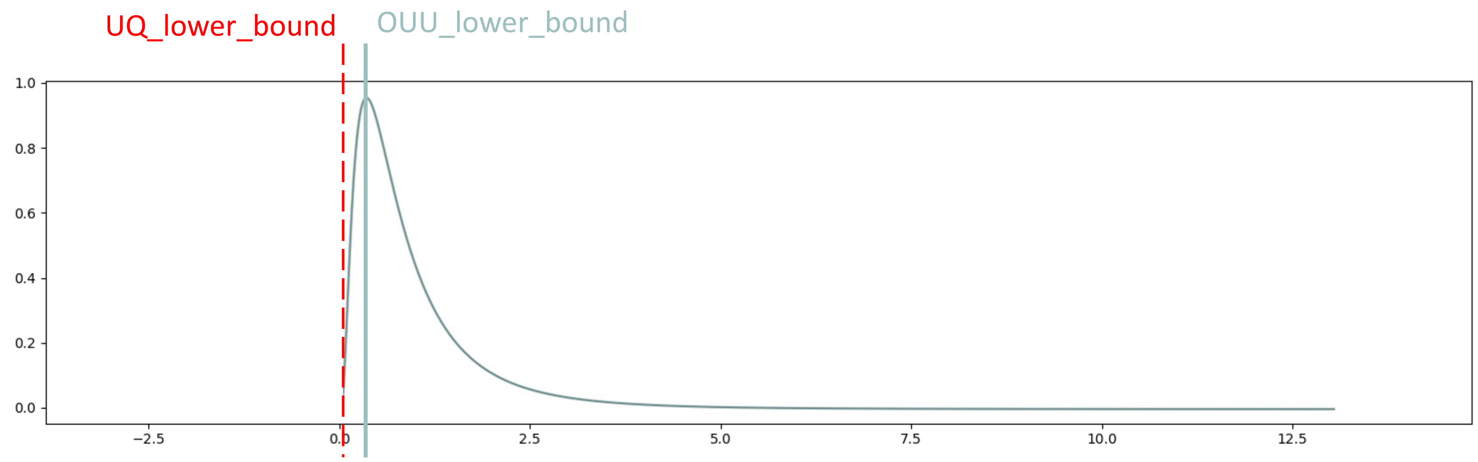
For first time users, the best practice is to ensure the following

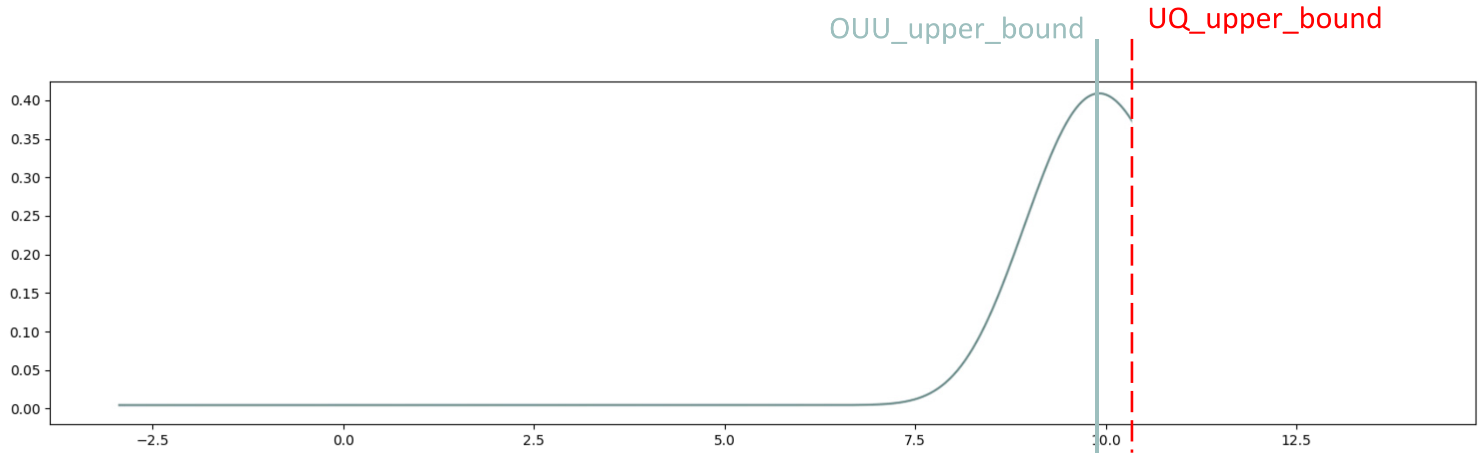
UQ_lower_bound < OUU_lower_bound

And

OUU_upper_bound < UQ_upper_bound.

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





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

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

And

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