CFD EXPERTS

Simulate the Future

WWW.CFDEXPERTS.NET

Ansys

2021 R2

Engineering What's Ahead.

DesignXplorer Optimization Tutorials

Ansys

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2021 R2
July 2021

Published in the U.S.A.

Table of Contents

Introduction 1
What is DesignXplorer? 1
DesignXplorer Features 2
"What If?" Studies 2
Parameters Correlation 3
Design of Experiments 5
Response Surface 5
Features that Use Response Surface Data 6
Six Sigma Analysis 7
Getting Started with DesignXplorer 8
DesignXplorer Workflow 8
Parametrization Basics 11
Parameterizing CAD Dimensions 12
Parameterizing Workbench Dimensions 12
Derived Parameters 16
DesignXplorer Workspaces 17
Parameter Set Workspace 19
DesignXplorer Workspaces 22
Parameters Correlation Workspace 23
Design of Experiments Workspace 24
Response Surface Workspace 25
Optimization Workspace 26
Six Sigma Analysis Workspace 27
Downloading Input Files for Tutorials 27
What Have You Learned? 28
Performing a Parametric "What If" Study 29
Getting Started 29
Promoting Dimensions and Solution Results to Input and Output Parameters 30
Promoting Dimensions in DesignModeler to Input Parameters 30
Promoting a Property and Solution Results in Mechanical to Output Parameters 32
Working with Design Points in Workbench 34
Adding Parameter Charts 35
Adding a Parameter Chart 35
Adding a Parameters Parallel Chart 36
Viewing the Solution 37
Performing a Parameters Correlation 41
What is a Parameters Correlation? 41
Getting Started 44
Defining Parameters 45
Generating Design Points 45
Performing the Parameters Correlation 45
Viewing the Correlation Matrix and Charts 47
Using a Design of Experiments 55
What is a Design of Experiments? 55
DOE Types 56
Getting Started 64
Working with the DOE 67
Working with the Response Surface 71
Solving for the Desired Design Point 76

Performing a Goal-Driven Optimization ... 79
What is Goal-Driven Optimization? ... 79
Getting Started ... 80
Promoting Properties and Results in Mechanical to Parameters ... 81
Performing the Optimization ... 83
Adding the Goal-Driven Optimization System ... 83
Configuring and Updating the Design of Experiments Cell ... 83
Configuring and Updating the Response Surface Cell ... 86
Configuring and Updating the Optimization Cell ... 88
Viewing Goal-Driven Optimization Charts ... 90
Viewing Candidate Points ... 90
Viewing the Solution ... 92
Viewing Results for the Most Viable Candidate Point ... 92
Viewing Results for the Response Point ... 95
Viewing Results in Mechanical ... 97
Using ACT to Expose MATLAB Optimizers in DesignXplorer ... 99
Using Adaptive Single-Objective Optimization ... 101
What is Adaptive Single-Objective Optimization? ... 101
Problem Definition ... 102
Problem Setup ... 102
Scenario 1: Kriging-NLPQL Response Surface Optimization to NLPQL Direct Optimization ... 103
Run the Response Surface Optimization ... 104
Run the Direct Optimization ... 111
How Effective was the Approach Used for Scenario 1? ... 113
Scenario 2: NLPQL Direct Optimization to NLPQL Direct Optimization ... 114
Run the First NLPQL Direct Optimization ... 114
Run the Second NLPQL Direct Optimization ... 116
How Effective was the Approach Used for Scenario 2? ... 119
Scenario 3: Screening Direct Optimization to NLPQL Direct Optimization ... 119
Run the Screening Direct Optimization ... 119
Run the NLPQL Direct Optimization ... 121
How Effective was the Approach Used for Scenario 3? ... 123
Scenario 4: Adaptive Single-Objective Direct Optimization ... 124
Run the ASO Direct Optimization System ... 124
How Effective was the Approach Used for Scenario 4? ... 128
Time to Spare? ... 129
What Have You Learned? ... 129
Different Types of Optimizations ... 129
Different Optimization Methods ... 130
Using Adaptive Multiple-Objective Optimization ... 131
What is Adaptive Multiple-Objective Optimization? ... 131
Problem 1 ... 131
Problem 1 Definition ... 131
Problem 1 Project Setup ... 132
Run the MOGA Direct Optimization ... 133
Run the AMO Direct Optimization ... 136
Problem 2 ... 139
Problem 2 Definition ... 139
Problem 2 Project Setup ... 140
Run the MOGA Direct Optimization ... 142
Run the AMO Direct Optimization ... 145

What Have You Learned? 149
Using Mixed-Integer Sequential Quadratic Programming (MISQP) Optimization 151
Getting Started 151
What is MISQP? 151
Problem Definition 152
Project Setup 152
Direct Optimization Using Screening 153
Response Surface Optimization to Direct Optimization Using Screening and MISQP 155
Run Screening in a Response Surface Optimization 156
Run MISQP in a Response Surface Optimization 159
Run MISQP in a Direct Optimization 162
Direct Optimization Using MISQP 164
Run MISQP in a Direct Optimization with a Maximum of 20 Iterations 165
Run MISQP in a Direct Optimization with a Maximum of 50 Iterations 167
What Have You Learned? 169

Introduction

This guide provides several tutorials for using Ansys DesignXplorer to analyze and optimize design spaces. The following topics introduce DesignXplorer and explain how to download the input files that you will need for the tutorials:

What is DesignXplorer?

DesignXplorer Features

Getting Started with DesignXplorer

DesignXplorer Workspaces

Downloading Input Files for Tutorials

What Have You Learned?

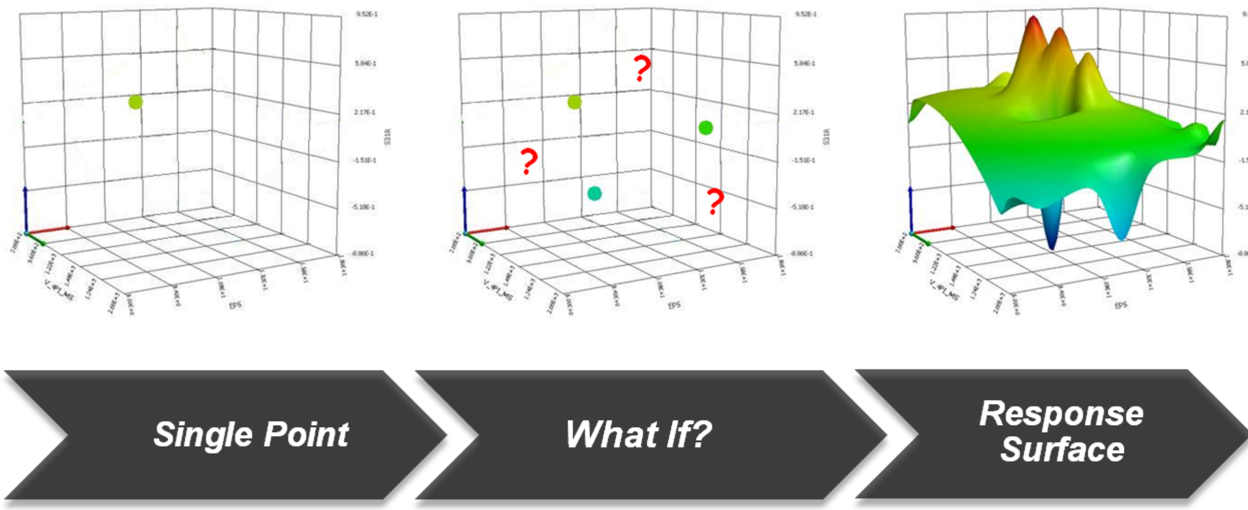
Note:

While this guide does not include tutorials for producing ROMs (reduced order models), you can find comprehensive ROM production and consumption information in "Using ROMs" in the *DesignXplorer User's Guide*.

What is DesignXplorer?

DesignXplorer is a tool that uses response surfaces to efficiently analyze a design space. With DesignXplorer, you can:

- Examine and understand the performance at other design or operating conditions
- Find the conditions which give the best performance
- Determine the key parameters influencing your design
- Evaluate the robustness of your design



DesignXplorer Features

DesignXplorer offers a variety of features to help you to examine and better understand your designs.

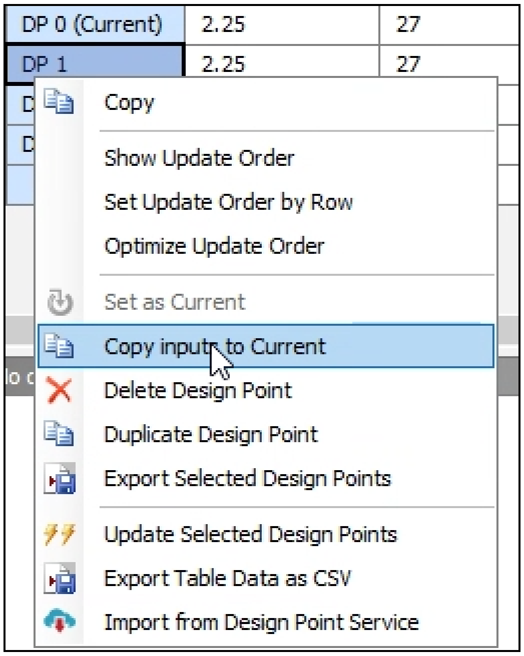
This section provides a brief introduction to these features:

"What If?" Studies
Parameters Correlation
Design of Experiments
Response Surface
Six Sigma Analysis

"What If?" Studies

Ansys Workbench offers the ability to perform a *"What if?"* study, which runs through a list of manually specified design points. You don't need a DesignXplorer license to perform such a study. A "What if?" study provides you with preliminary information that you can then use in DesignXplorer for more in-depth analysis.

When you have a Workbench project open, in the **Project Schematic**, double-clicking the **Parameter Set** bar opens it. In the **Table** pane, you can see the design points that have been generated. Right-clicking a design point allows you to select an option from the context menu.



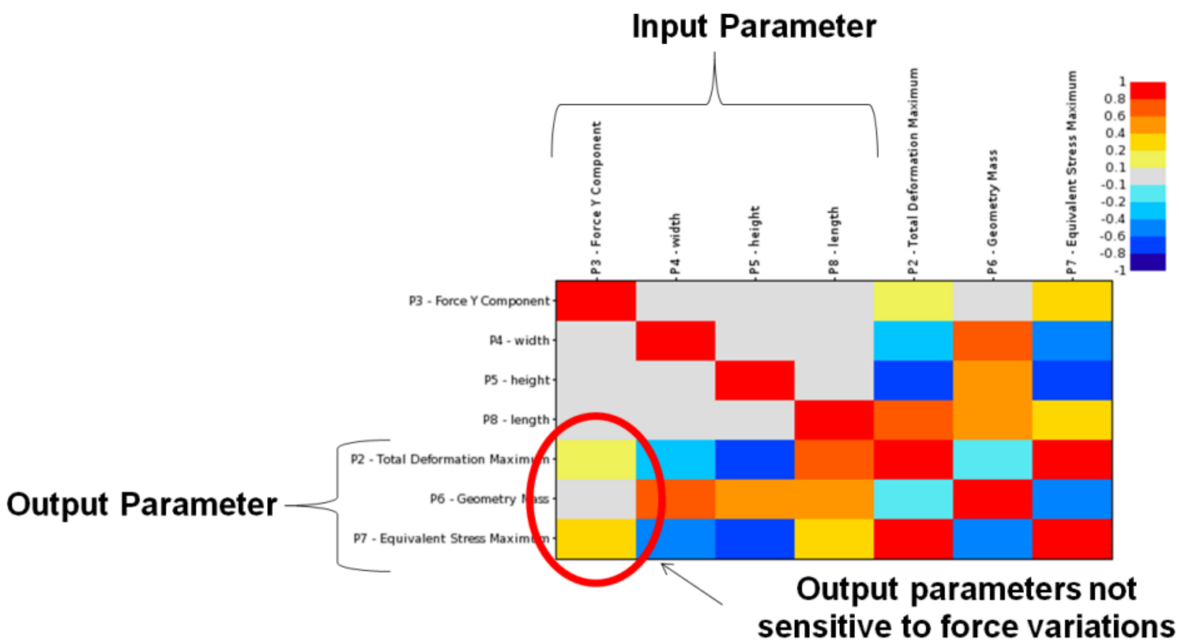
For more information, see the tutorial Performing a Parametric "What If" Study (p. 29).

Parameters Correlation

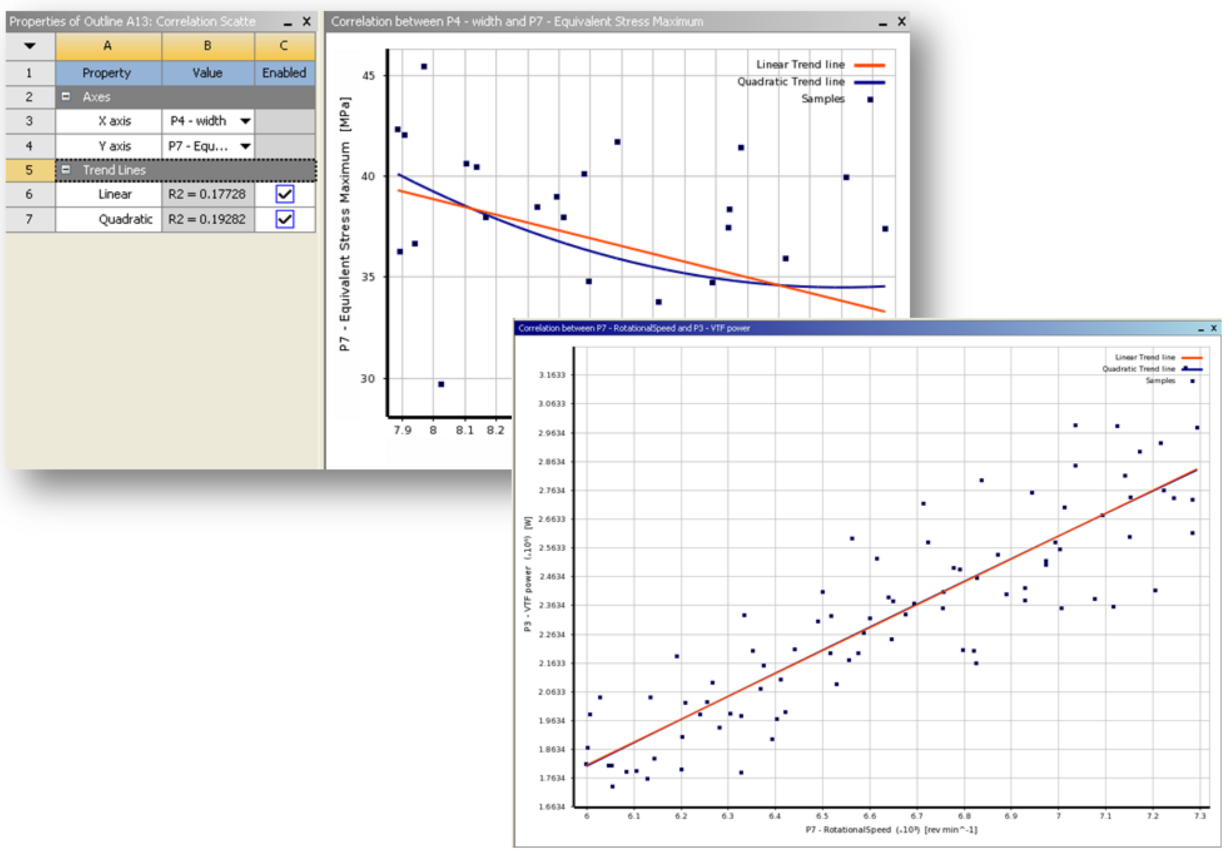
A *parameters correlation* performs simulations on a random sampling of the design space to identify the correlations between all parameters in the project. A linear association between parameters is evaluated using the Spearman's or Pearson's product-moment coefficient. A correlation or sensitivity matrix is generated to demonstrate the correlation between input and output parameters and the sensitivity of output parameters to input parameters.

A parameters correlation allows you to:

- Determine which input parameters have the most (and the least) impact on your design.



- Identify the degree to which the relationship is linear or quadratic.



For more information, see the tutorial Performing a Parameters Correlation (p. 41).

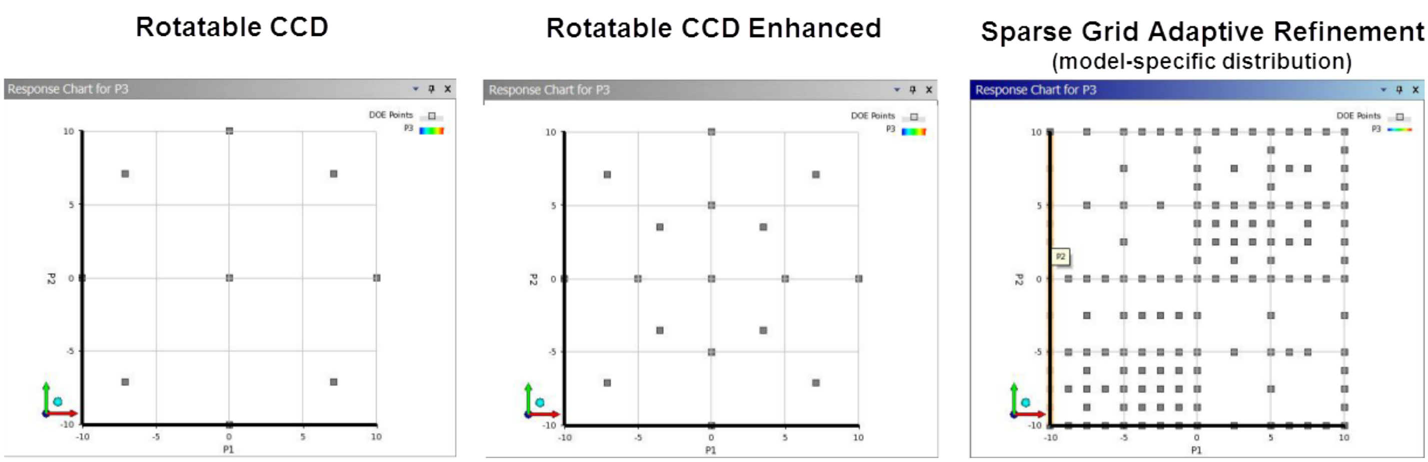
Design of Experiments

DesignXplorer uses a technique called a *Design of Experiments* (DOE) to identify design points with parameter combinations for exploring the solution space most efficiently. The most efficient solution uses the fewest number of design points. A DOE works best with fewer than 20 parameters.

DesignXplorer makes several DOE types available so that you can select the type best suited to your project and purposes. Available DOE types include:

- Central Composite Design (CCD)
- Optimal Space-Filling Design
- Box-Behnken Design
- Custom
- Custom + Sampling
- Sparse Grid Initialization
- Latin Hypercube Sampling Design

These images show examples of design points generated with various DOE types:



For more information, see the tutorial Using a Design of Experiments (p. 55).

Response Surface

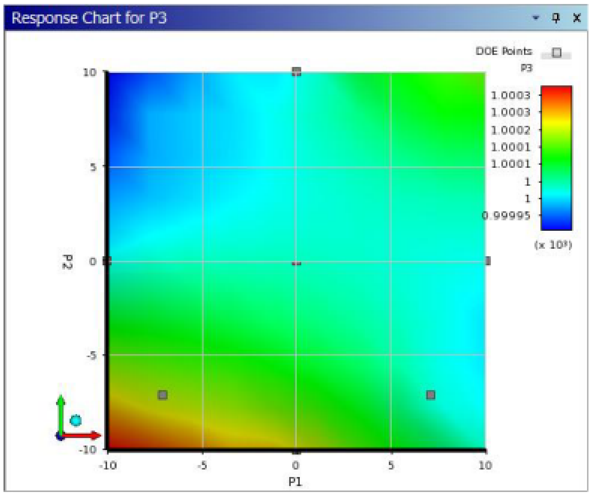
From the design points identified in a DOE, DesignXplorer builds a continuous *response surface*. Just as there are several DOE types, there are several response surface types:

- Genetic Aggregation
- Full 2nd Order Polynomials
- Kriging
- Non-Parametric Regression

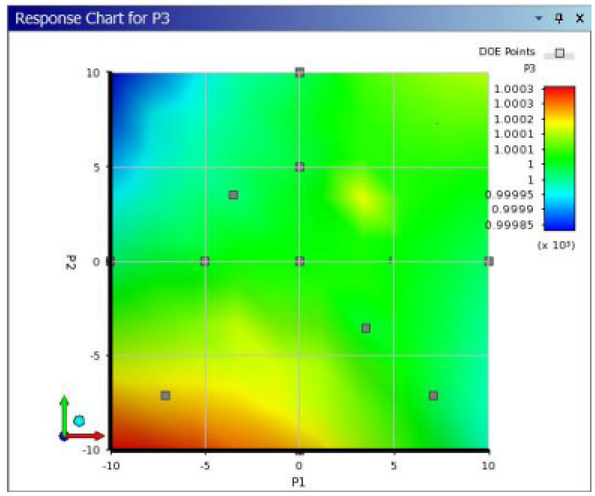
- Neural Network
- Sparse Grid

The following images are response surfaces generated from the design points shown in the previous topic:

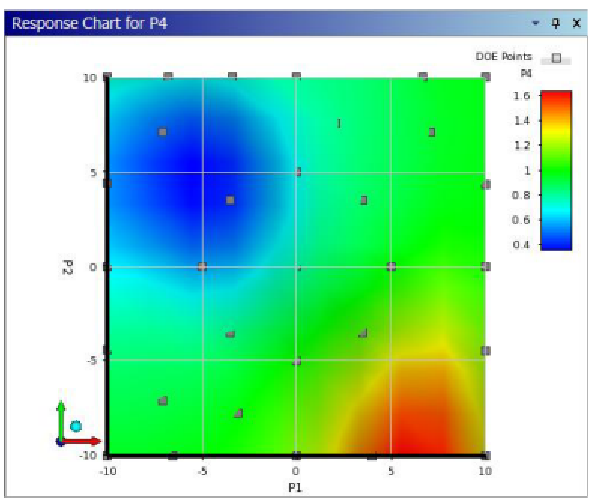
Rotatable CCD



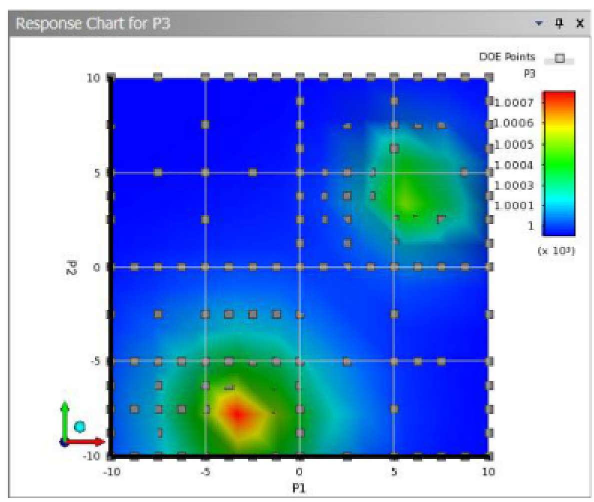
Rotatable CCD Enhanced



Rotatable CCD Enhanced: Auto-Kriging



Sparse Grid

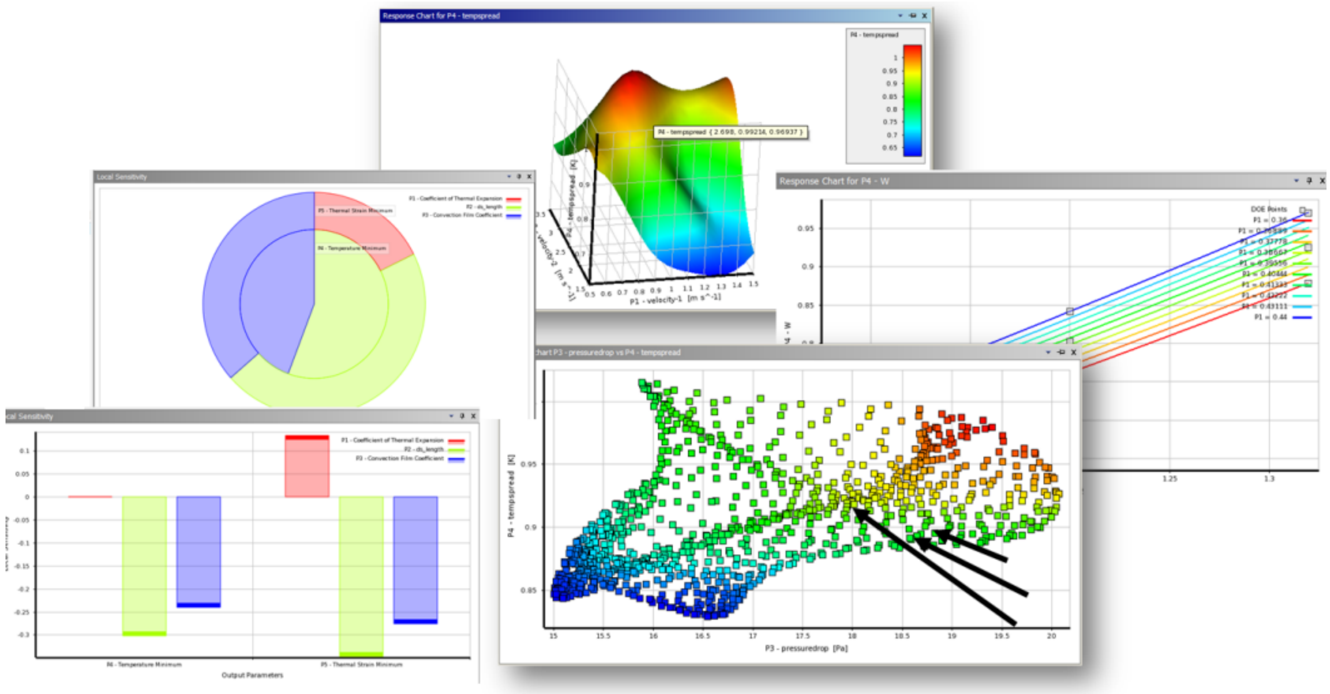


Features that Use Response Surface Data

Because the response surface provides approximated values for output parameters, other DesignXplorer features can quickly explore and use the data that it provides. Examples of features that make use of response surface data include:

- Min/Max Search
- Response surface charts

- Sensitivity charts
- Optimization Tradeoff charts
- Robustness of design analyses



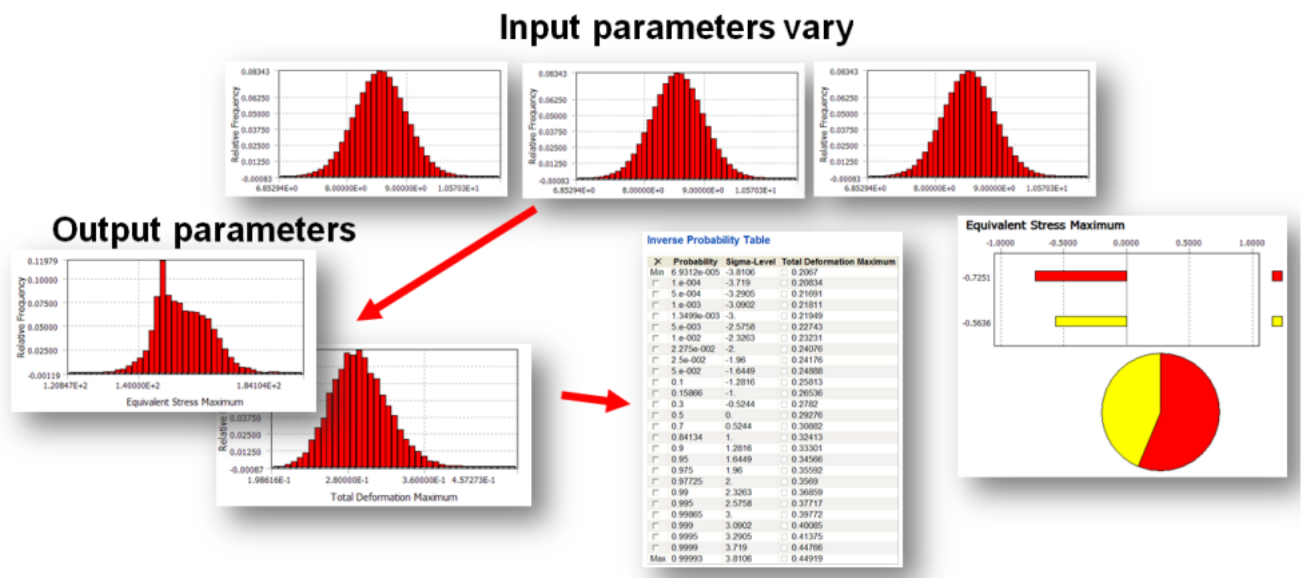
Six Sigma Analysis

The *Six Sigma Analysis* feature of DesignXplorer is an essential part of the "robust design" process. Robust design takes the variation of inputs into account and seeks a design with a probabilistic goal. In practice, dimensions (such as thickness), material properties (such as viscosity and density), and boundary conditions (such as resistance and flow rate) all have variation. Six Sigma Analysis factors in the variation and produces results that also have variation. By ensuring that results don't vary outside of the acceptable range, you can ensure that your design is robust.

With a Six Sigma Analysis, you can:

- Understand how your performance will vary with your design tolerances
- Determine how many and which parts are likely to fail
- Learn which inputs require the greatest control

A Six Sigma Analysis uses statistical distribution functions (such as Gaussian, normal, uniform, and so on) to describe uncertainty parameters.



Getting Started with DesignXplorer

This section describes the DesignXplorer workflow and parametrization basics:

DesignXplorer Workflow

Parametrization Basics

DesignXplorer Workflow

The workflow for using DesignXplorer basically consists of three steps:

1. Create parameters.

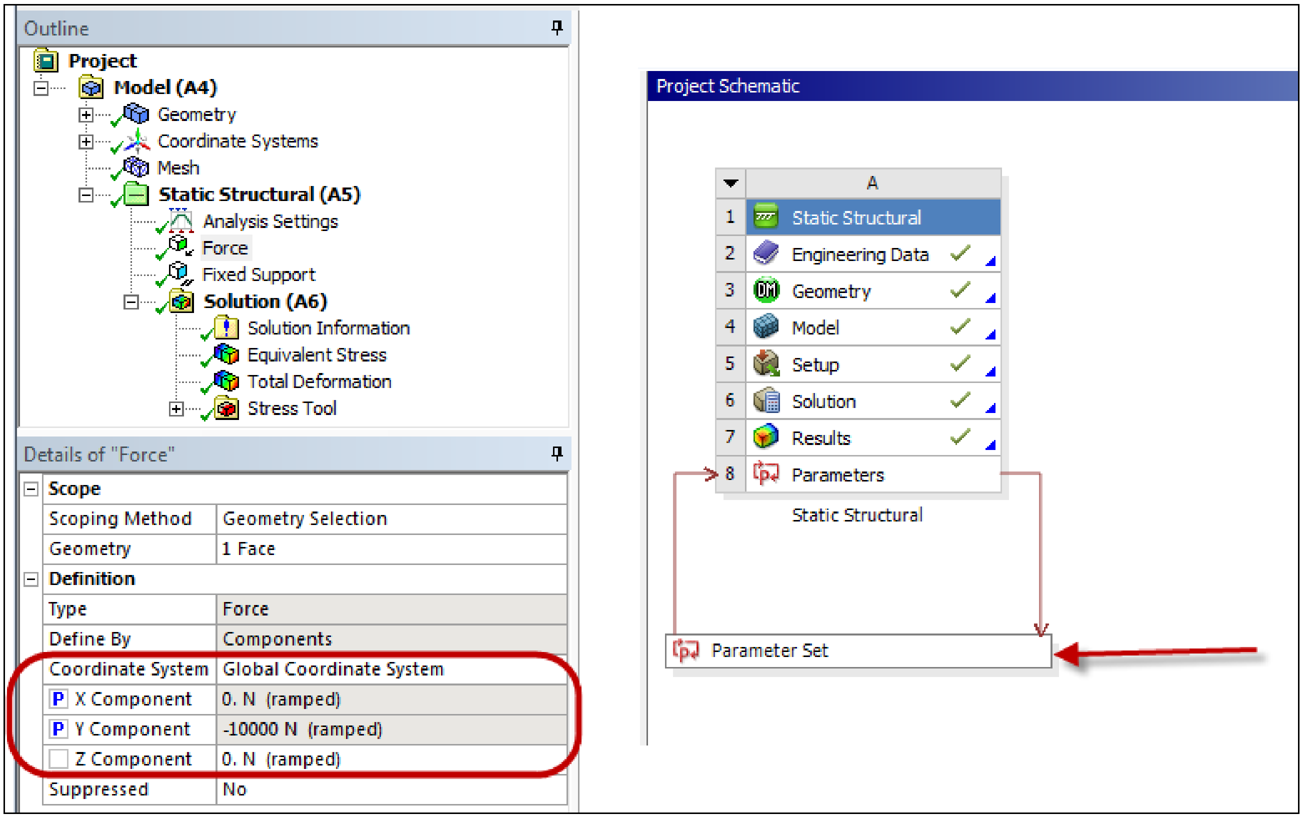
2. Add a DesignXplorer system to the Workbench project.

3. Work through the DesignXplorer system.

Step 1: Create Parameters

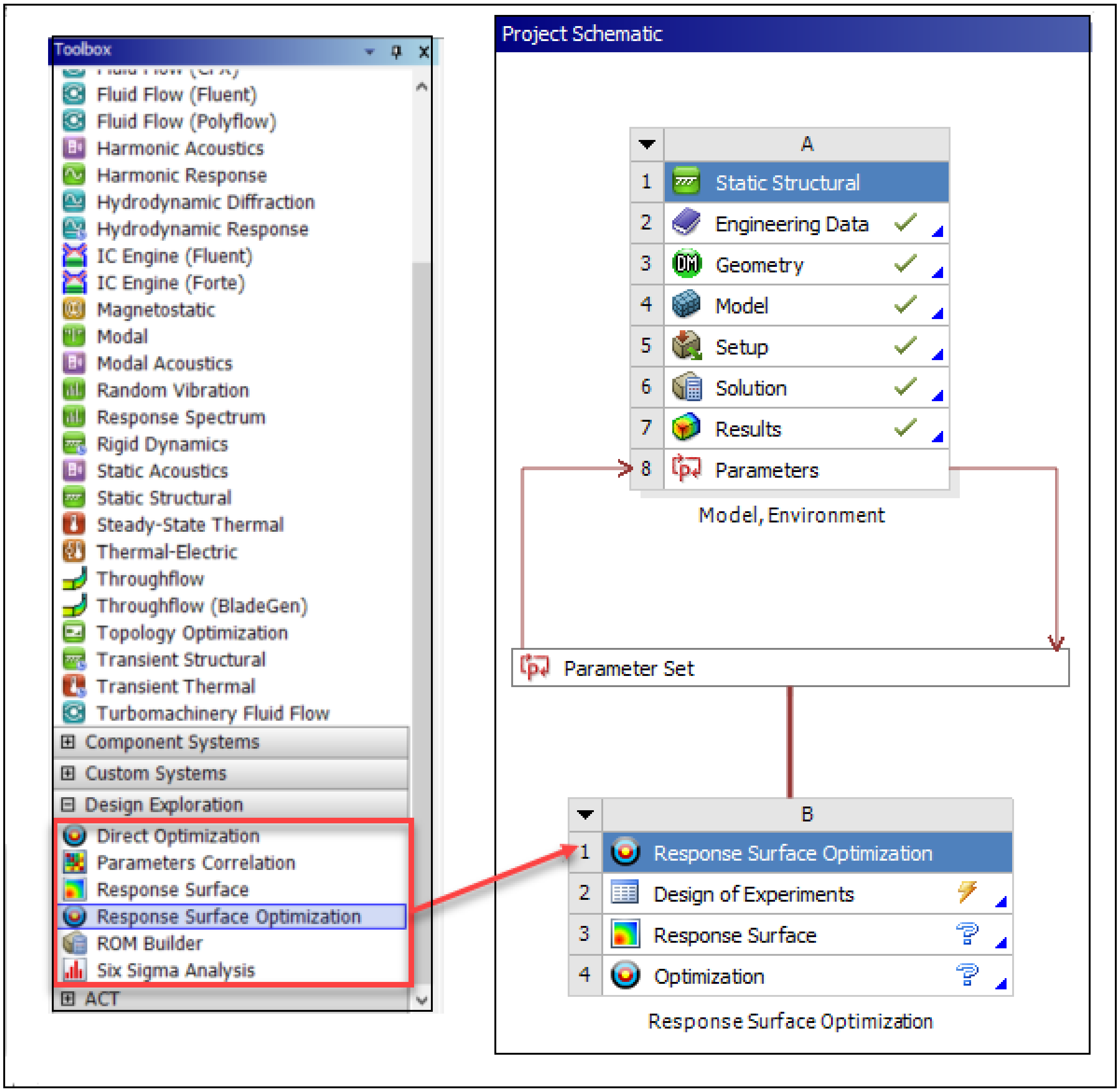
To work with DesignXplorer, you must have parameters in your project. Thus, your first step is to parameterize dimensions, boundary conditions, and such.

Once parameters are created, the **Parameter Set** bar is added to the Workbench **Project Schematic**. The following figure shows two values being parameterized in Mechanical and the subsequent inclusion of the **Parameter Set** bar in the **Project Schematic**.



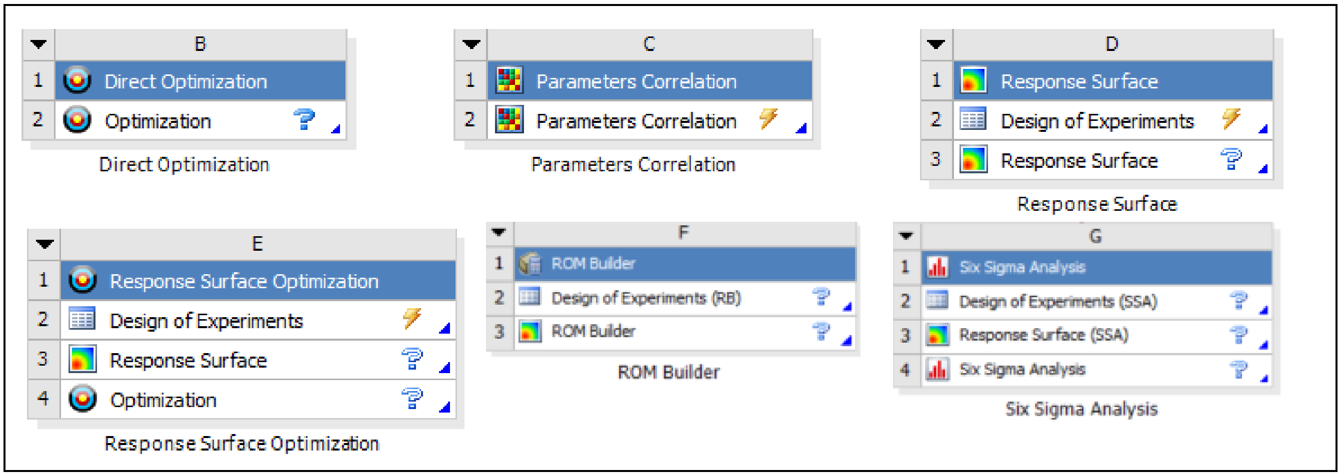
Step 2: Add a DesignXplorer System to the Project

Following parameter creation, you can add one or more DesignXplorer systems to the project. To do this, in the Workbench **Toolbox** under **Design Exploration**, locate the desired system and then drag it on the **Project Schematic**, dropping it under the **Parameter Set** bar. In the following figure, a **Response Surface Optimization** system has been added to the **Project Schematic**.



Step 3: Work Through the DesignXplorer System

After adding a DesignXplorer system to the **Project Schematic**, you must work through its components, which are the individual cells in the system, in sequence from top to bottom. Each type of design exploration system has a different workflow:



The **Direct Optimization** and **Parameters Correlation** systems are the only ones that do not have **Design of Experiments** cells. For all other DesignXplorer systems, the **Design of Experiments** cell is first in the sequence. In the following systems, the **Response Surface** cell is second: **Response Surface**, **Response Surface Optimization**, and **Six Sigma Analysis**. DesignXplorer systems can share and reuse **Design of Experiments** and **Response Surface** cells.

Parametrization Basics

Parameters can be defined not only in Workbench applications but also in numerous external CAD packages. Applications that allow for parametrization include the following:

- DesignModeler
- Mechanical
- Mechanical APDL
- Fluent
- CFX
- Polyflow
- SpaceClaim
- ExplicitSTR
- CFD-Post
- Vista TF
- Meshing
- Engineering Data
- Microsoft Excel

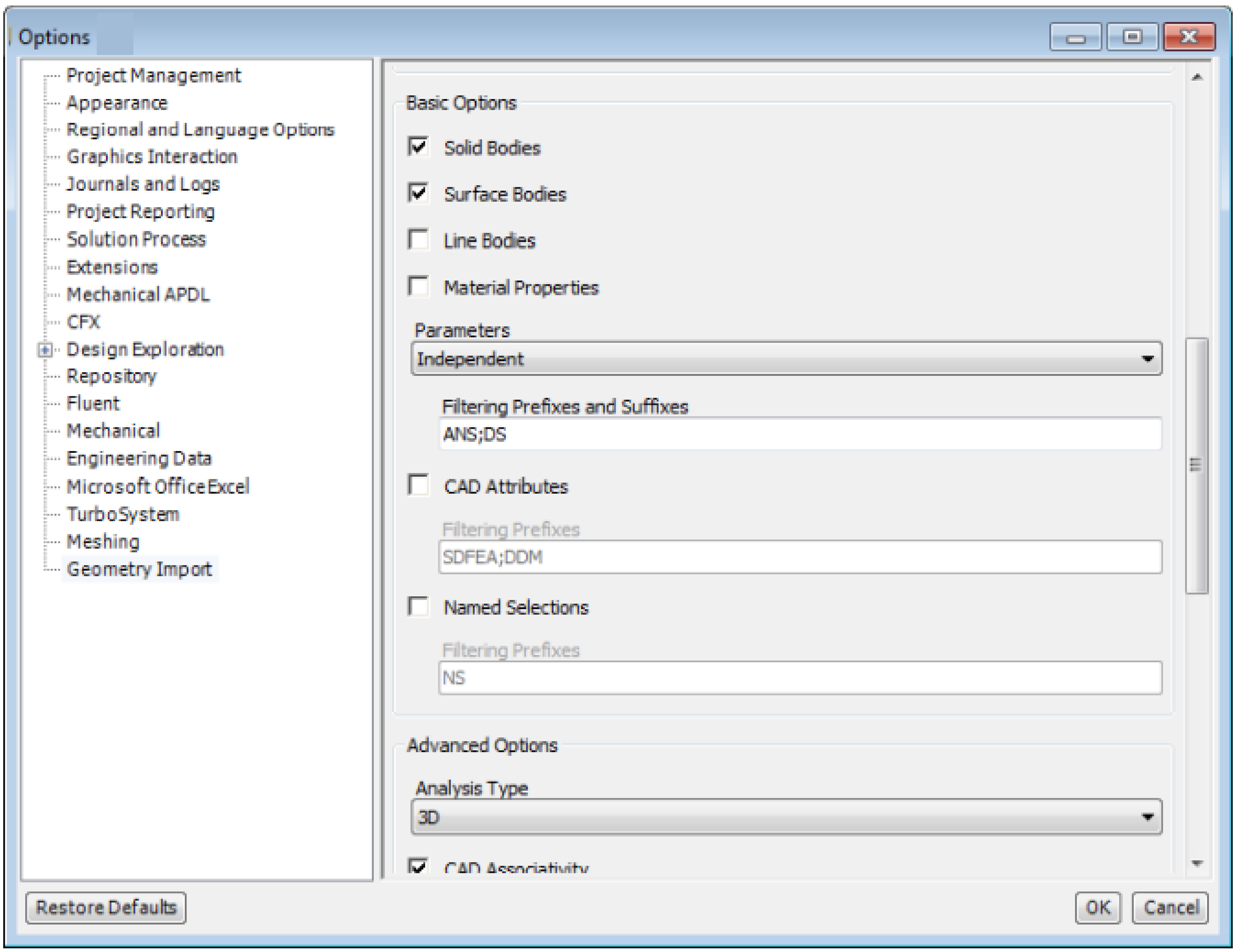
Parameterizing CAD Dimensions

To parameterize CAD dimensions, you must first enable parameters in Workbench:

1. From the Workbench menu, select **Tools** → **Options** → **Geometry Import**.

2. For **Filtering Prefixes and Suffixes**, specify any keys that must appear at the beginning or end of a CAD parameter name for it to display in Workbench.

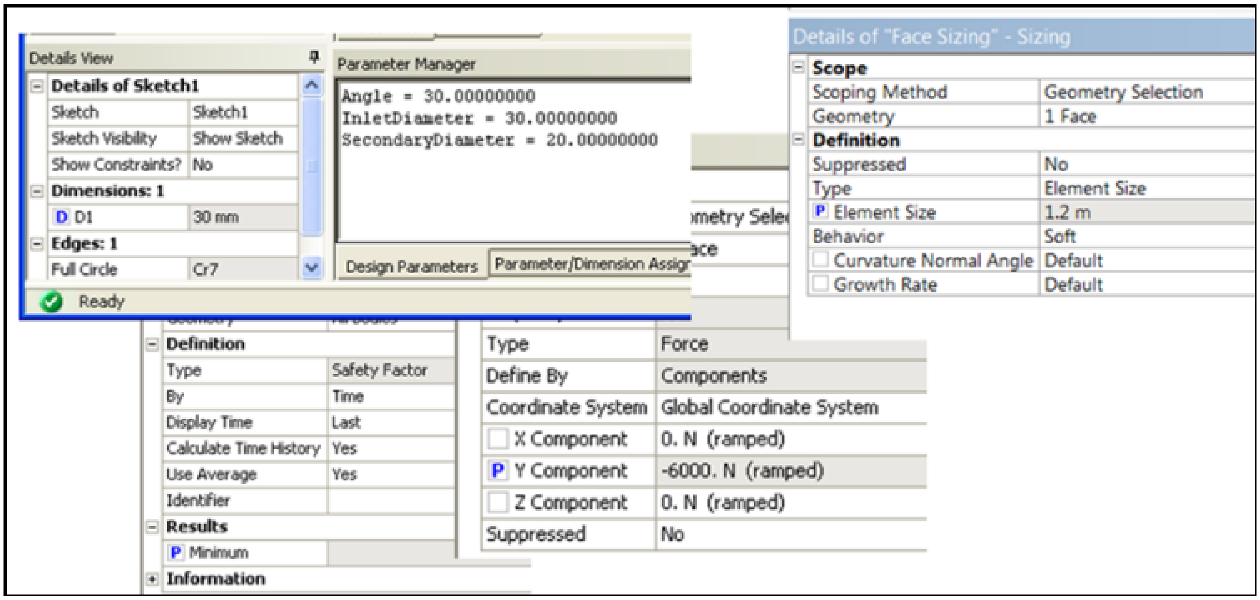
   The default is **ANS;DS**. However, you can specify any number of prefixes or suffixes as keys. For example, to expose a CAD parameter named **Length** in Workbench, you could rename it to something like **dslength**, **Lengthds**, **ds_Length**, or **Length_ds**. The order, underscore, and case do not matter. All of these would be recognized as matching the key **DS**.



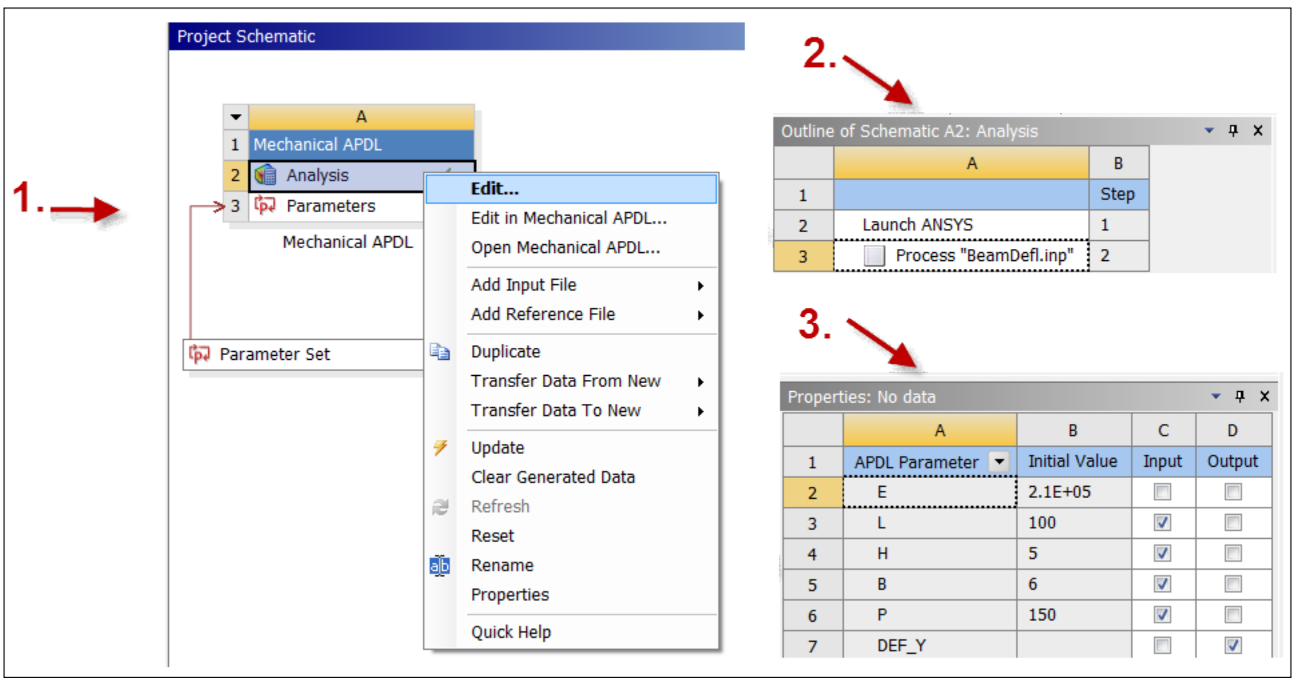
Parameterizing Workbench Dimensions

The process for parameterizing Workbench dimensions depends on the application. The following figures show different methods for parameterizing model data. For more information, see the documentation for the application that you are using.

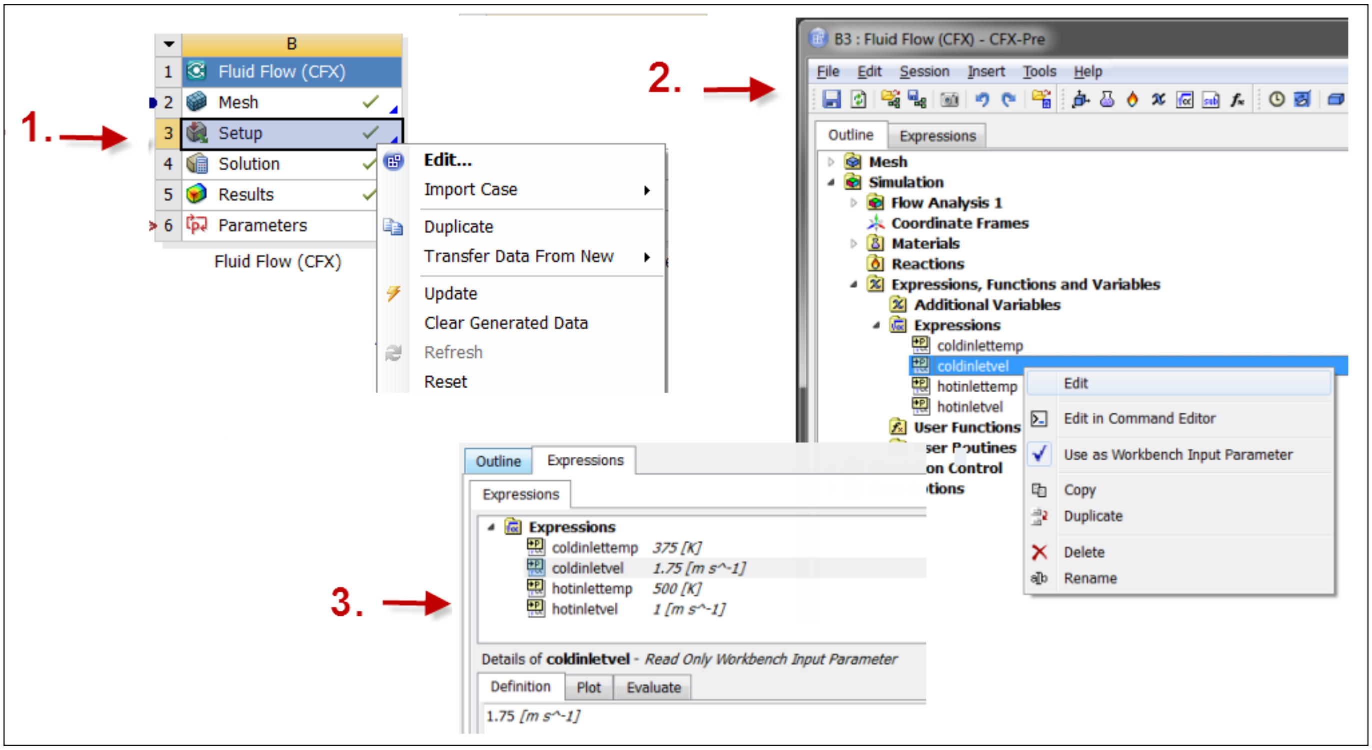
Input and Output Parameters Created in DesignModeler and Mechanical



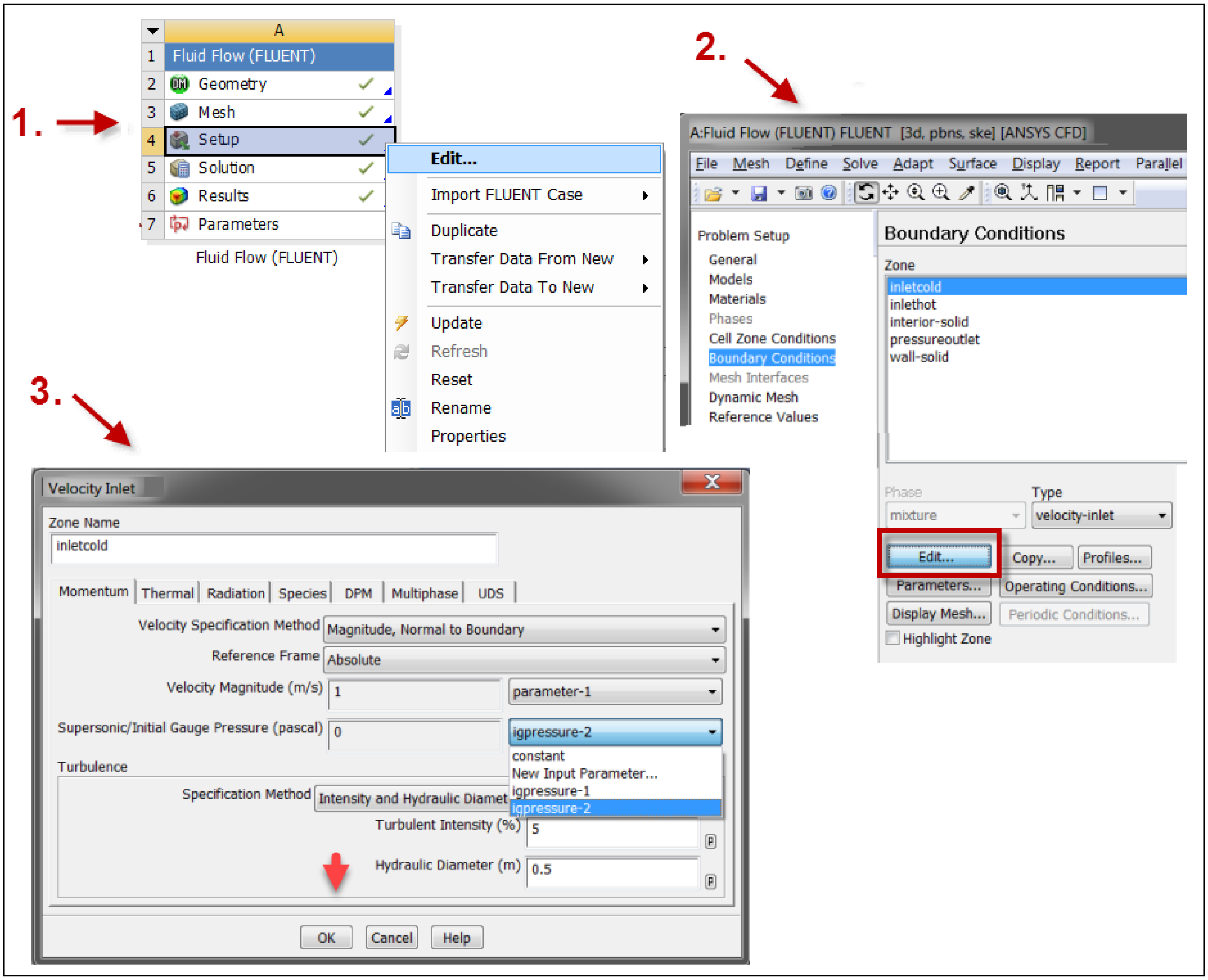
Parameter Definition Based on a Mechanical APDL Input File



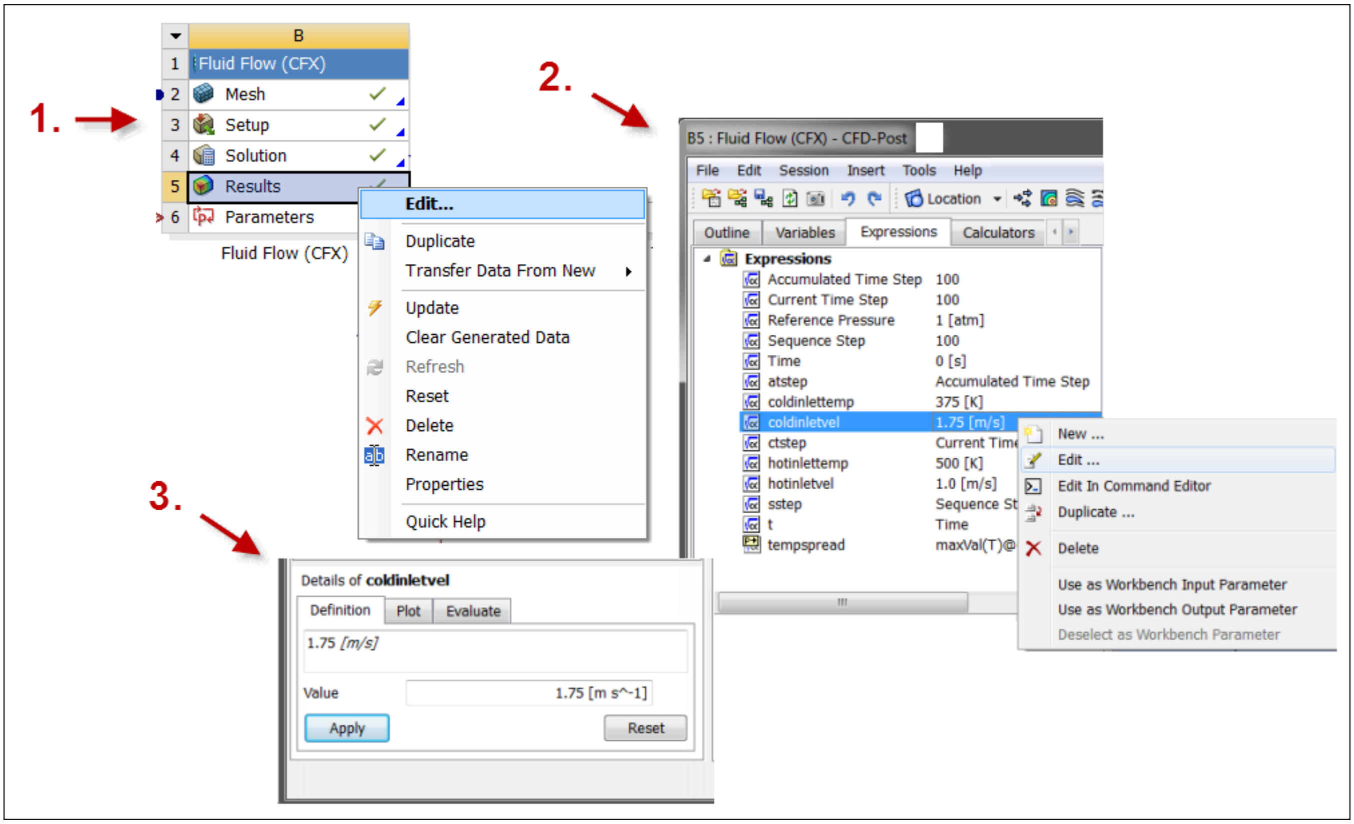
Input Parameters Created from an Expression in CFX



Input Parameter Created from an Expression in Fluent



Output Parameters Created from an Expression in CFD-Post



Derived Parameters

Derived parameters are created using analytical expressions composed of input parameters, output parameters, or both input and output parameters. As the definition suggests, derived parameters are calculated from other parameters by using equations that you provide.

Some examples of derived parameters include:

- **Cost Function**: Product of mass and cost per mass
- **Normalized Stress**: Stress response divided by an applied stress
- **Average Value**: Average of the first three frequencies
- **Mesh Sizing**: Mesh parameter set as a function of a geometric parameter

---

Note:

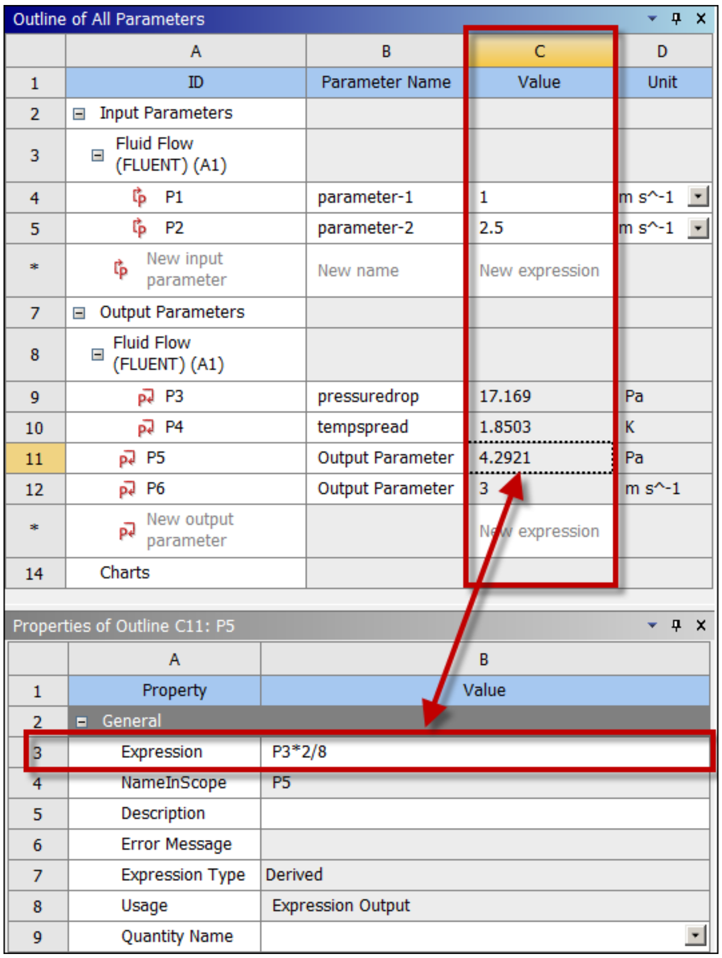
Derived parameters cannot reference other derived parameters.

Defining Derived Parameters

You can define derived parameters using various built-in arithmetic, trigonometric, and statistical functions. They are created in the analysis system and passed into DesignXplorer as output parameters.

To create a derived parameter:

1. Double-click the **Parameter Set** bar to open it.
2. Enter the expression in either of these locations:
    - **Value** field in the **Outline** pane
    - **Expression** field in the **Properties** pane (after selecting the parameter in the **Outline** pane)



Note:

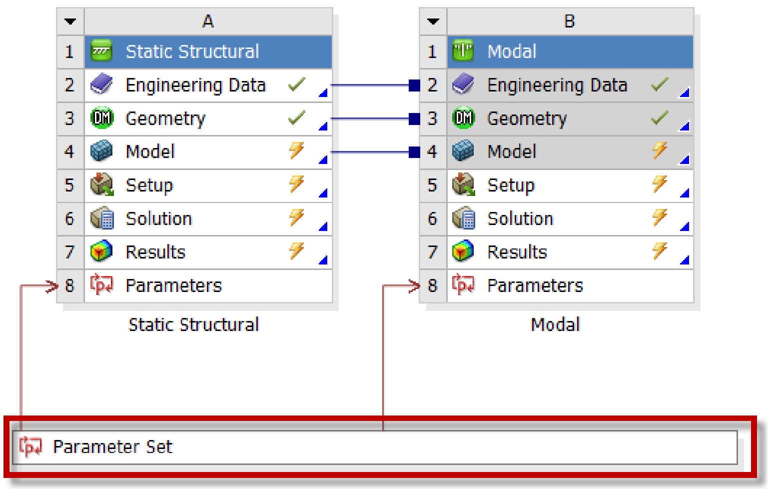
You can add both derived parameters and constant values in the **Outline** pane. However, once derived parameters are added, they become read-only in the **Outline** pane. To edit them, you must use the **Properties** pane.

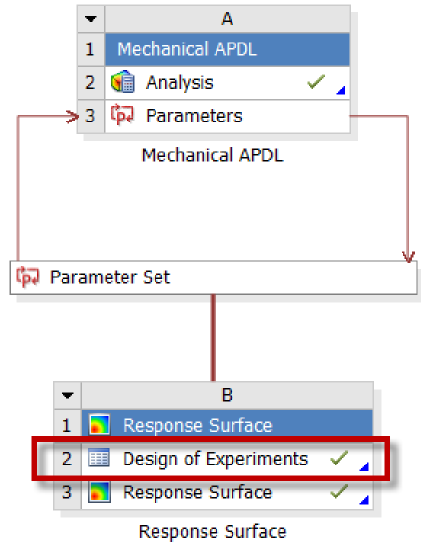
DesignXplorer Workspaces

When you open the **Parameter Set** bar or a DesignXplorer system cell, you will see that the layouts of their workspaces are very similar.

Workspace Access

To open the **Parameter Set** bar or a DesignXplorer cell, double-click it.





Workspace Layout

Each workspace for a DesignXplorer cell includes the **Toolbox** with chart options and the following panes:

- **Outline**
- **Properties**
- **Table**
- **Chart**

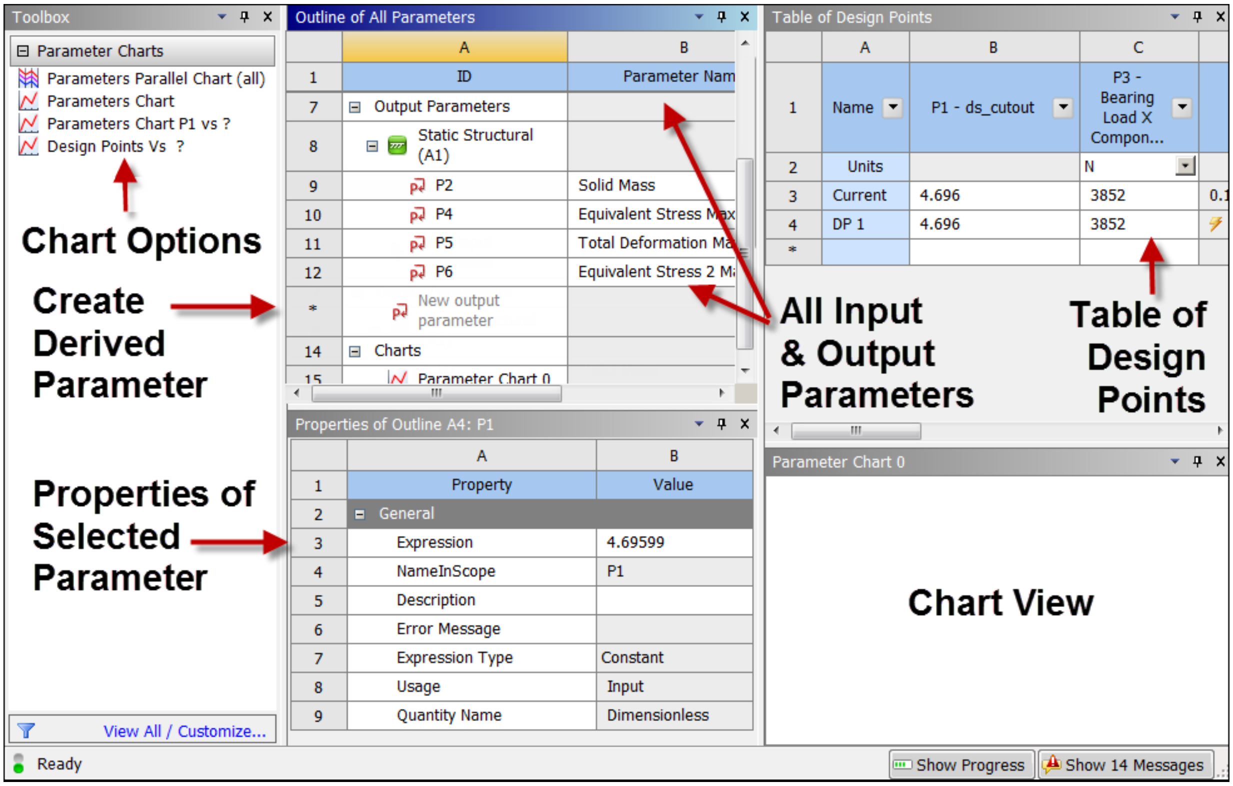
Workspaces are connected between the various Workbench native applications. For example, if you close the chart on one workspace, it is also closed on the other workspaces. To reset the workspace to the default layout, select **View** → **Reset Workspace**.

The following topics show the various workspaces you'll use when working in DesignXplorer:

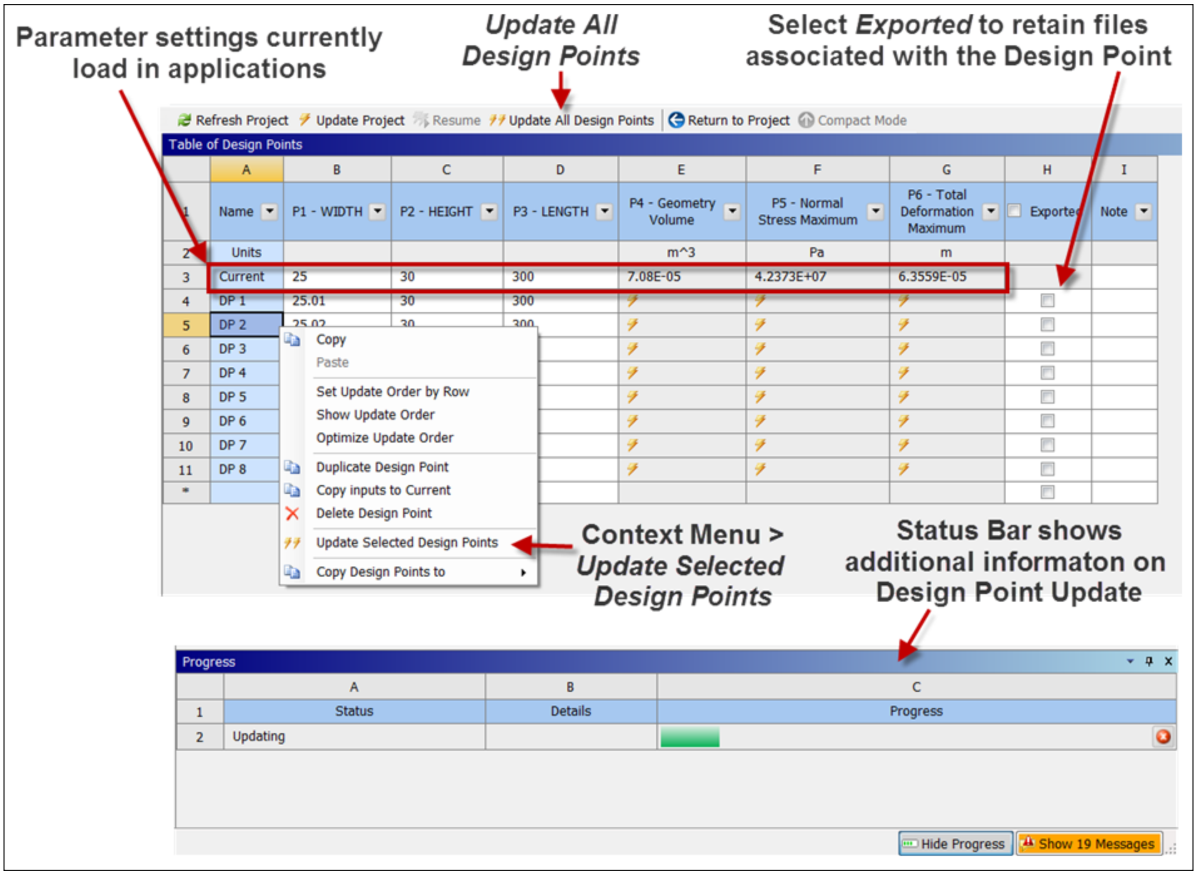
Parameter Set Workspace

DesignXplorer Workspaces

Parameter Set Workspace



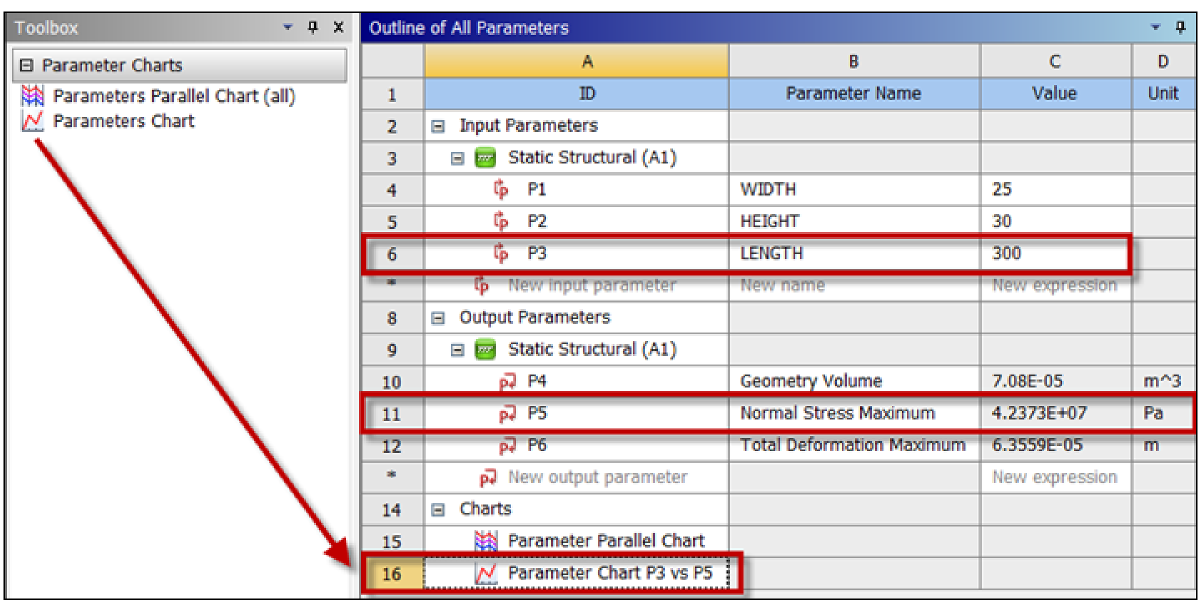
Parameter Set Table of Design Points



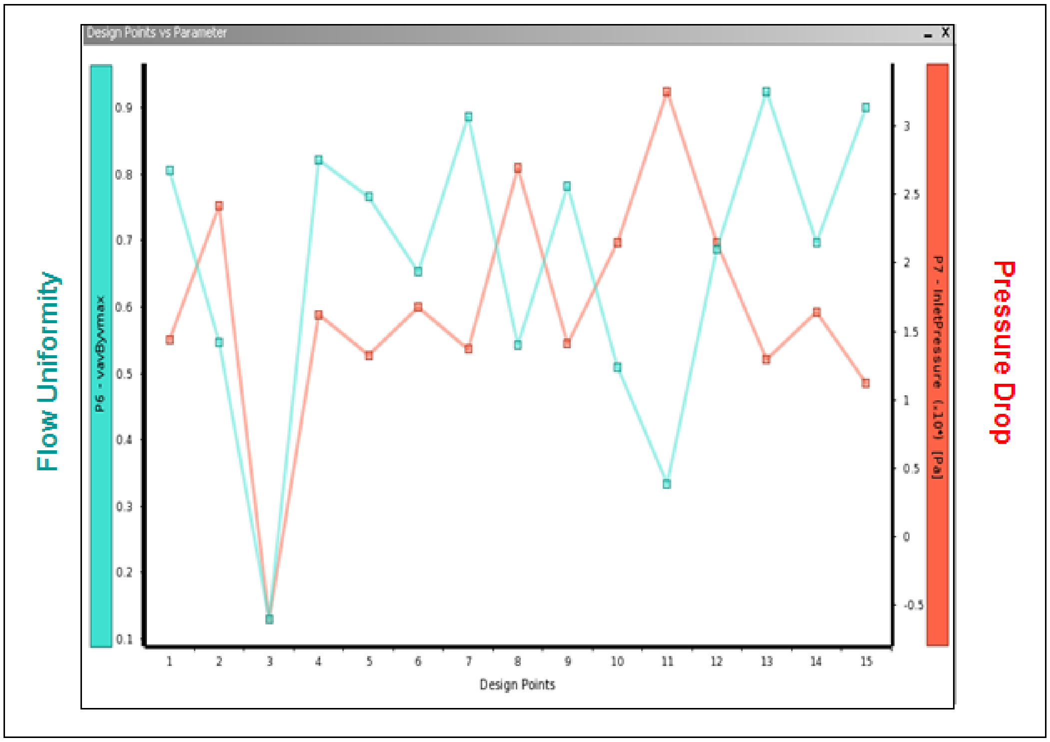
Parameter Set Charts

The charts available from the **Parameter Set** bar allow you to do the following things:

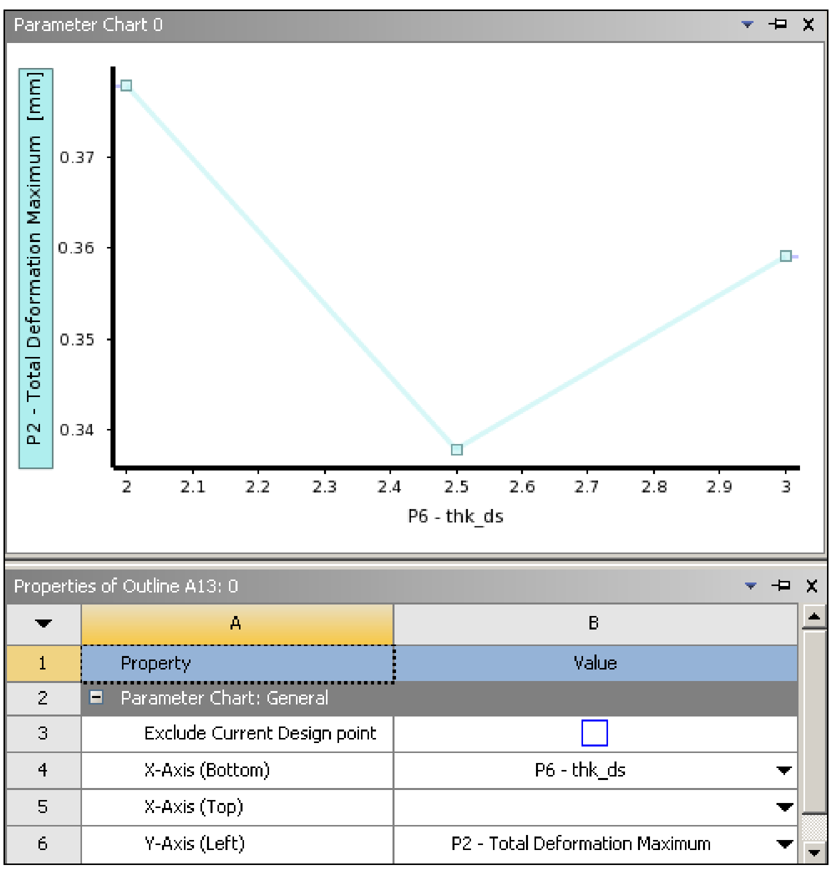
- Configure and plot an input parameter versus an output parameter in XY plots



- See how output parameters vary for design point combinations



- See the chart output versus the input parameter

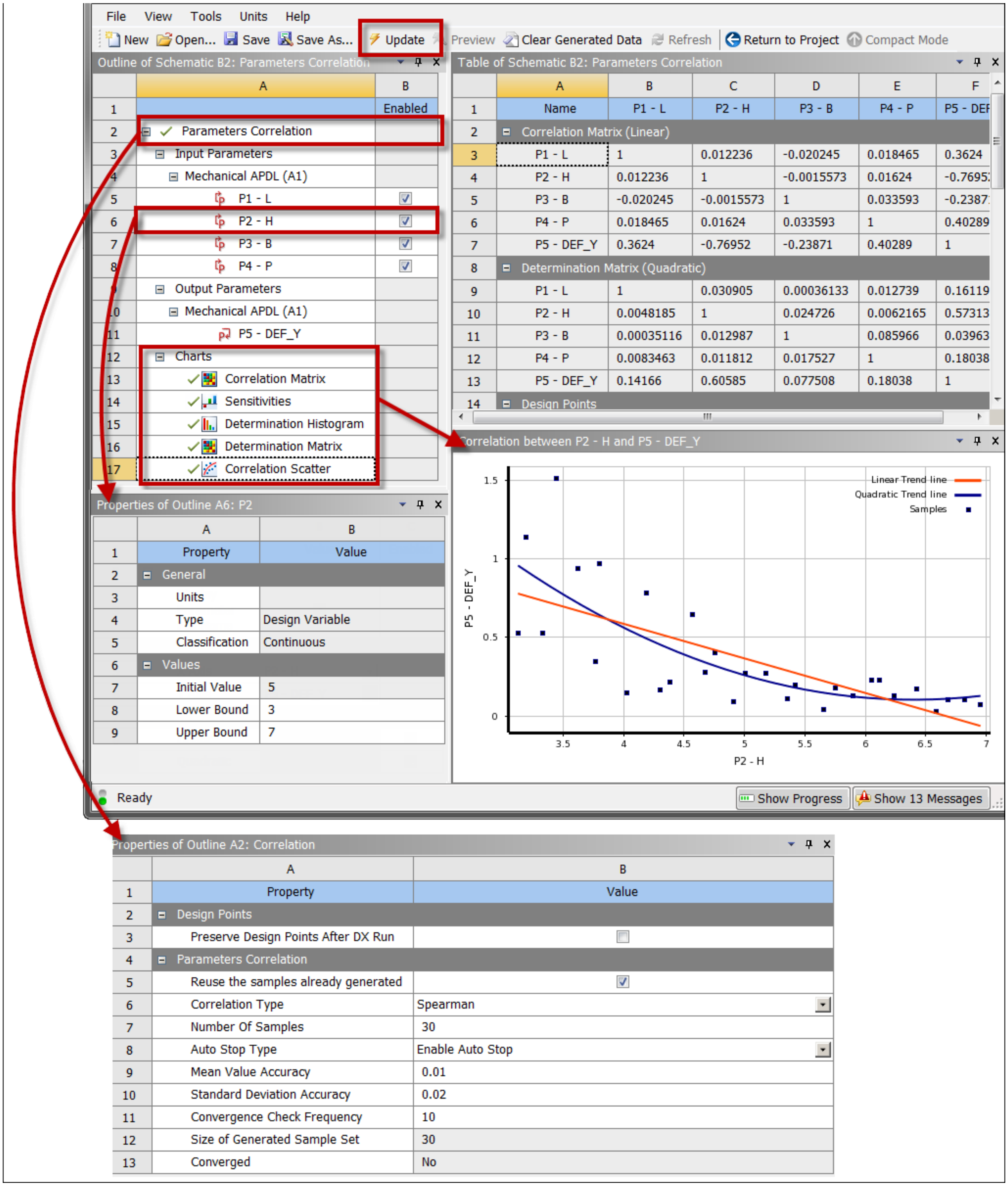


DesignXplorer Workspaces

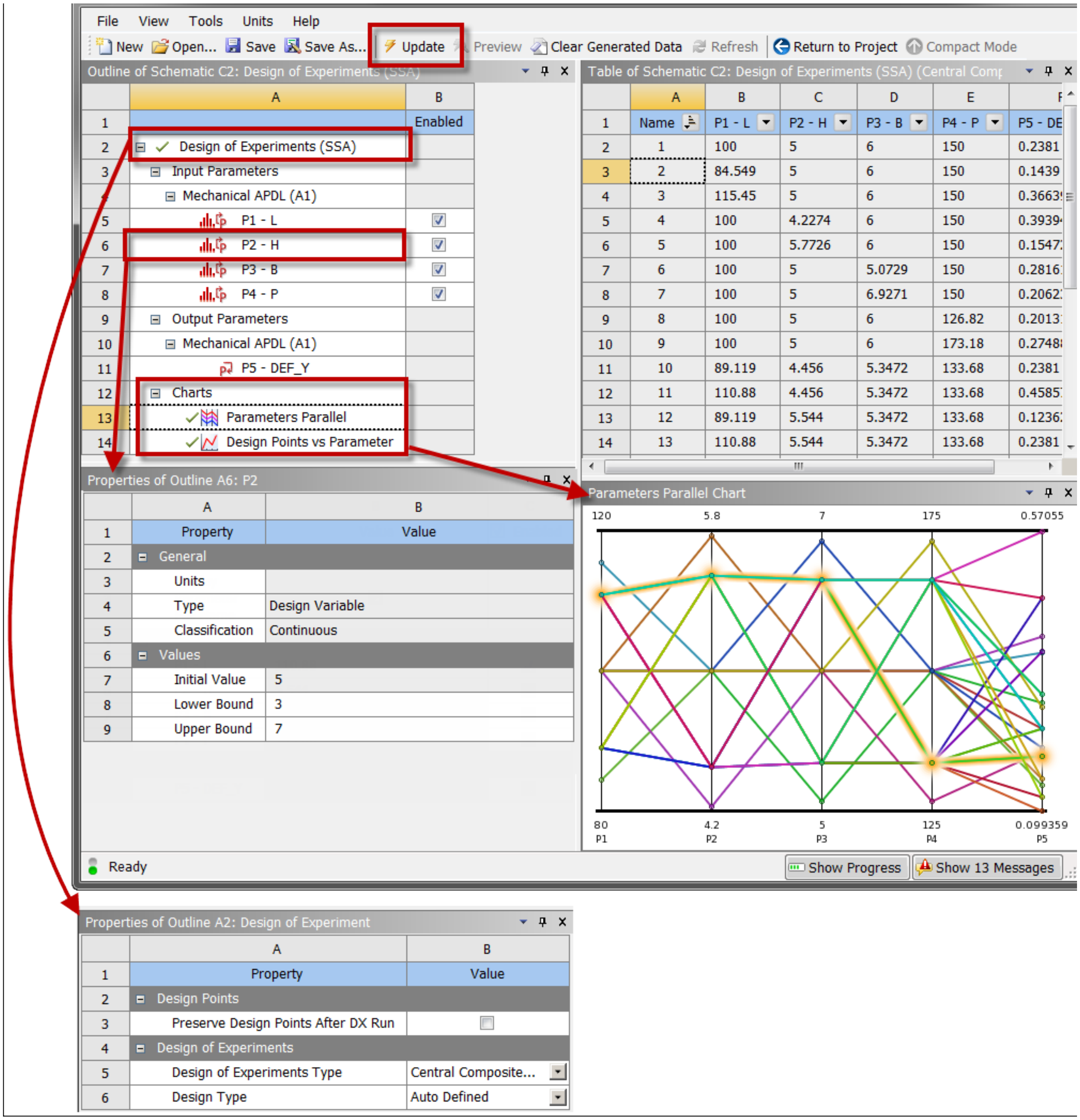
This section includes images of the following DesignXplorer workspaces:

- Parameters Correlation Workspace
- Design of Experiments Workspace
- Response Surface Workspace
- Optimization Workspace
- Six Sigma Analysis Workspace

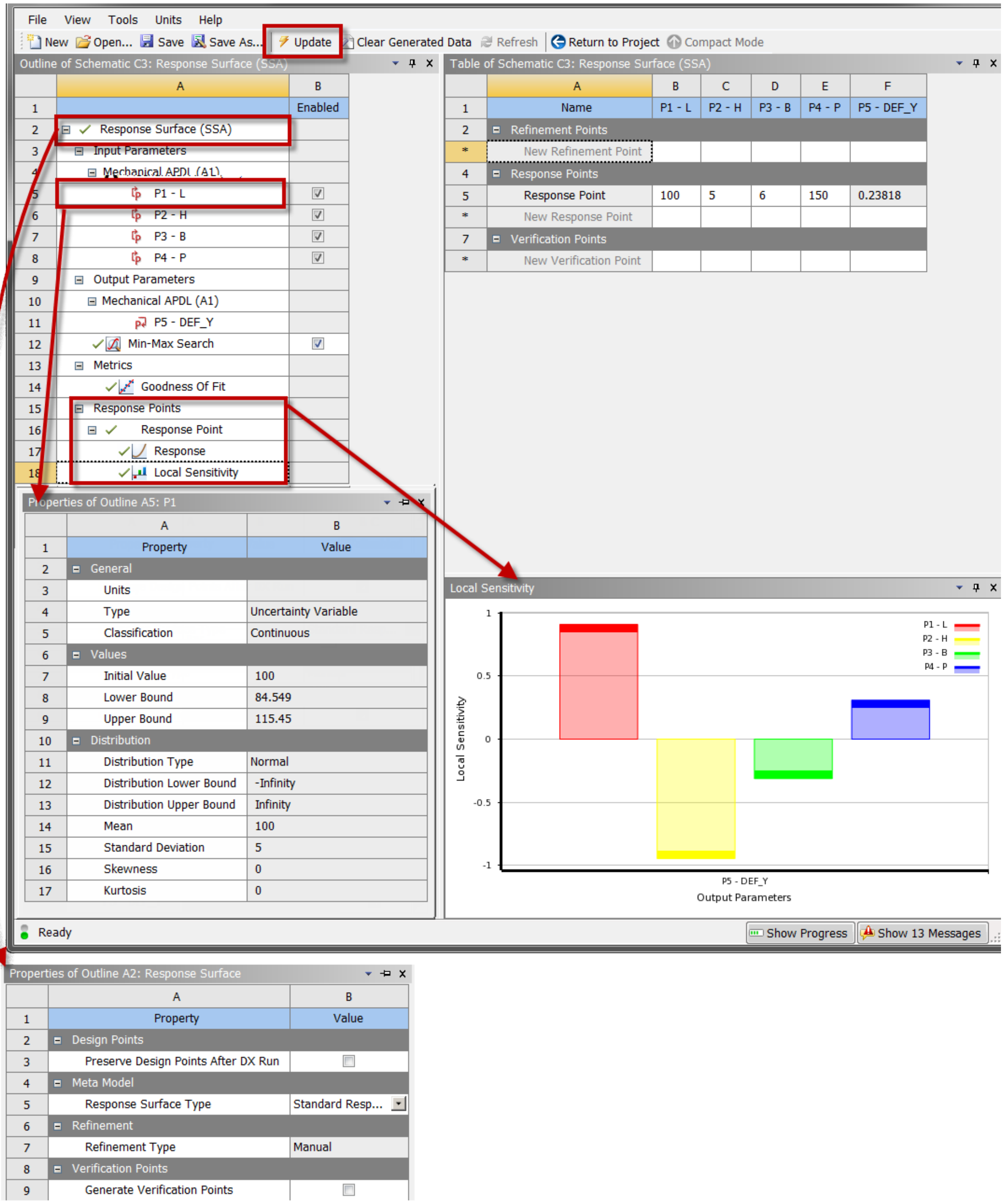
Parameters Correlation Workspace



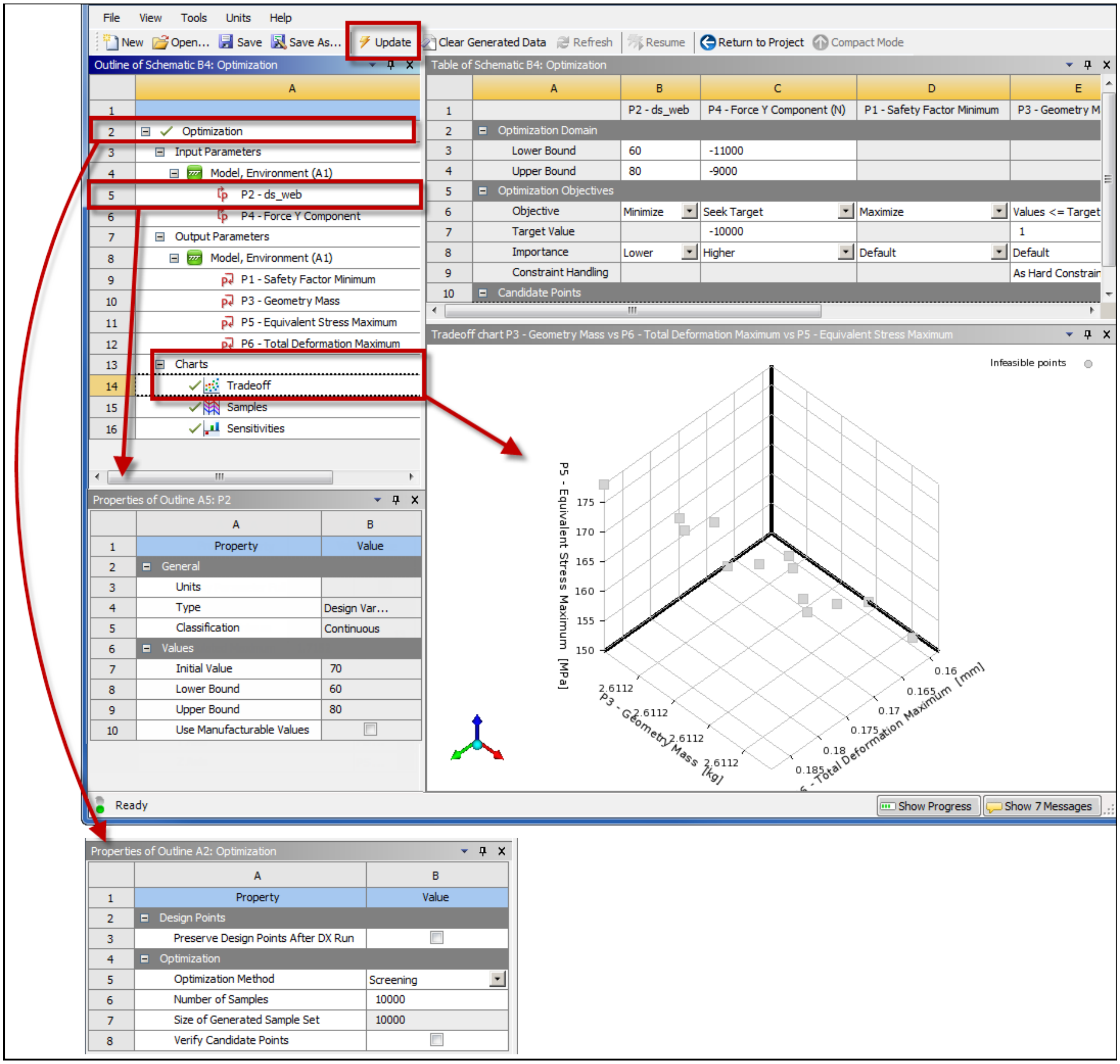
Design of Experiments Workspace



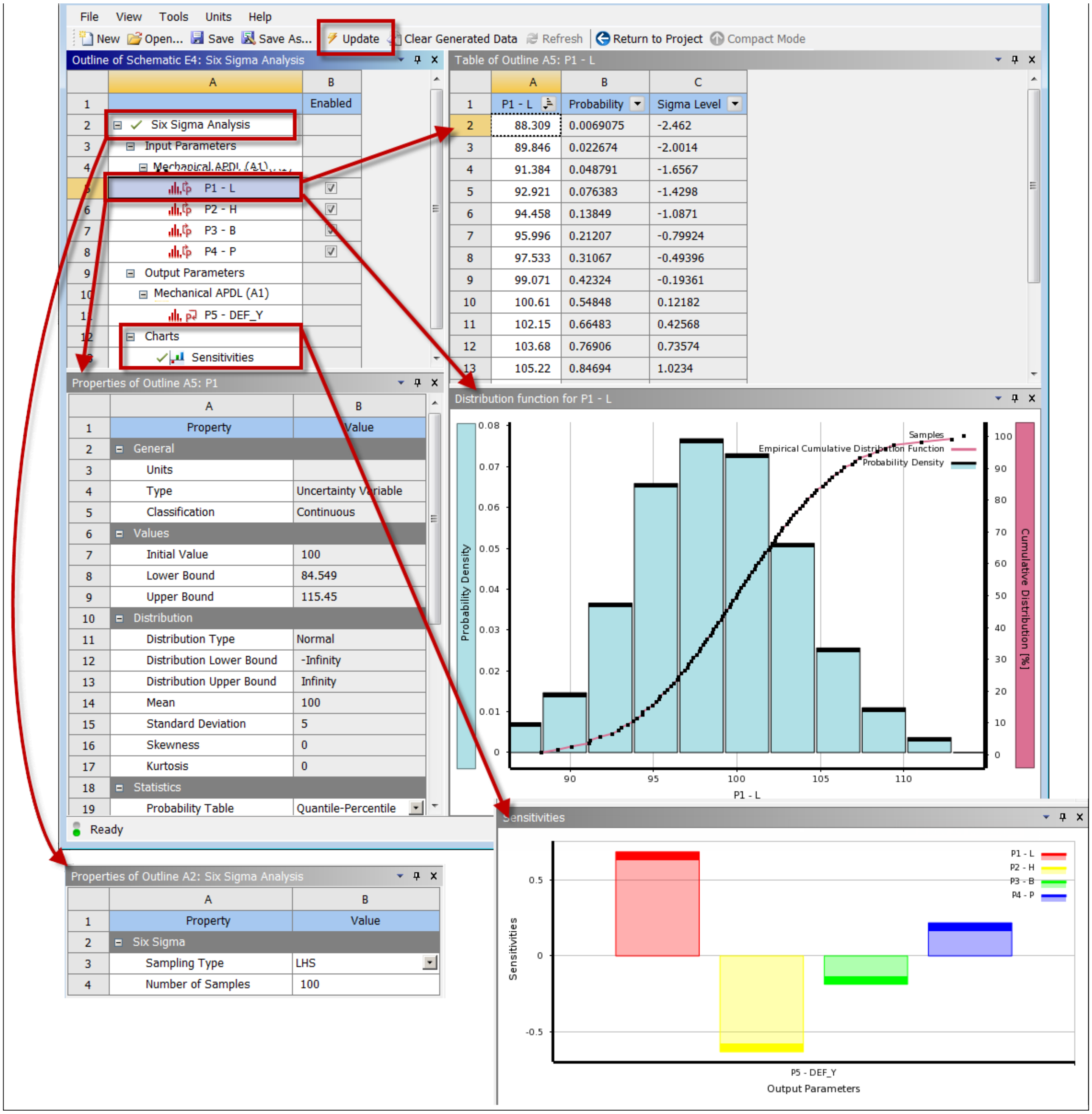
Response Surface Workspace



Optimization Workspace



Six Sigma Analysis Workspace



Downloading Input Files for Tutorials

For the optimization tutorials, download and extract the files that are used as inputs:

1. Click this link to download the ZIP file that contains the input files.
2. Extract these files to a directory that you can access.

Each tutorial will instruct you to navigate to and select one or more files in this directory.

What Have You Learned?

From this tutorial, you've learned the following:

- DesignXplorer uses response surfaces to help you to investigate your design space, letting you find the combinations of input parameters that provide the best performance, determine how parameters influence your design, and evaluate the robustness of your design.
- DesignXplorer offers a variety of features to help you to better understand your design, including:
  - “What if?” studies
  - Parameters correlations
  - DOEs
  - Response surfaces
  - Goal-driven optimizations
  - Six Sigma analyses
- The general workflow for using DesignXplorer consists of the following steps:
  - Define parameters.
  - Add a DesignXplorer system to the **Project Schematic**.
  - Work through the sequence indicated by the cells in the DesignXplorer system.
- Parameters can be added in most Ansys products and numerous external CAD packages via the user interface, Mechanical APDL input files, and user-defined expressions.
- The workspaces for the **Parameter Set** bar and DesignXplorer system cells all have similar layouts. Each workspace includes the **Toolbox** with chart options and the following panes:
  - **Outline**
  - **Properties**
  - **Table**
  - **Chart**
- Assuming that you have downloaded and extracted the input files (p. 27) for the subsequent tutorials, you are now ready to learn more by stepping through example projects.

Note:

As you step through these example projects, your results may not exactly match the results that are shown in this guide. However, your results should be very similar.

Performing a Parametric "What If" Study

This tutorial gives step-by-step instructions for performing a parametric "What If?" study of a hitch receiver. It investigates the behavior of stress, mass, and deformation in the receiver as geometry parameters are changed during vertical loading.

Note:

"What If?" studies use functionality inherent to Ansys Workbench. A DesignXplorer license is not required to complete this tutorial.

This tutorial covers the following topics:

Getting Started

Promoting Dimensions and Solution Results to Input and Output Parameters

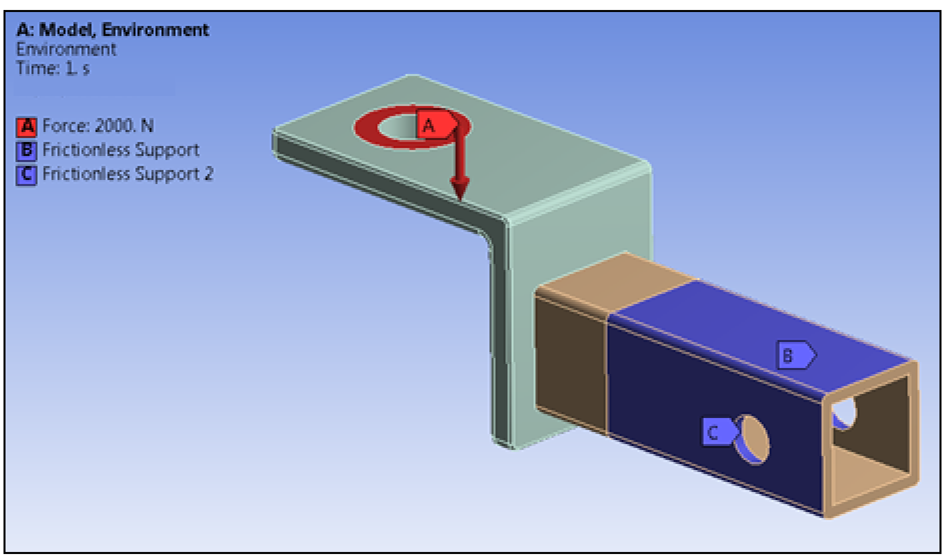
Working with Design Points in Workbench

Adding Parameter Charts

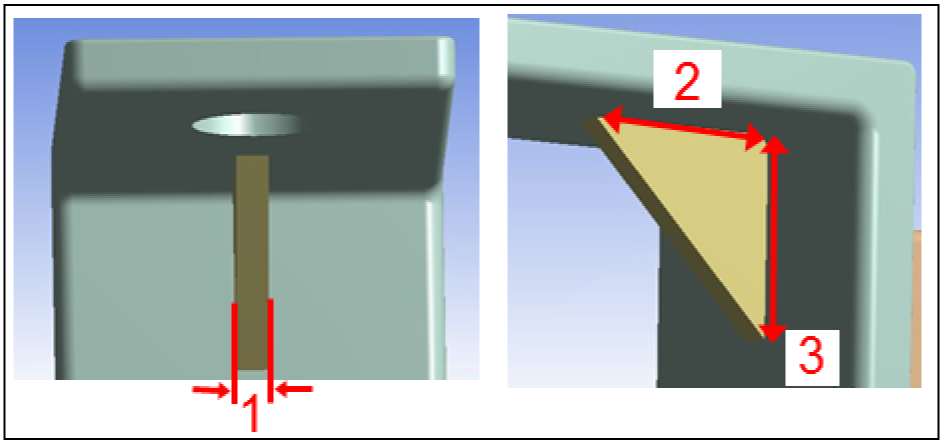
Viewing the Solution

Getting Started

The hitch receiver is composed of a multibody part (three bodies) created in Ansys DesignModeler. The following figure shows constraints and loads.



In this tutorial, you'll promote the dimensions labeled **1**, **2**, and **3** in the following figure to input parameters.



Opening the Archived Workbench Project

1. Start Workbench and select **File** → **Open**.
2. In the file browser, locate and open the archived Workbench project file `receiver.wbpz`, which is in the directory to which you extracted the input files (p. 27).

   A dialog box appears for saving this archived file to a standard Workbench project file.
3. Save the project as `reciever.wbpj` in either this same directory or another directory.

Promoting Dimensions and Solution Results to Input and Output Parameters

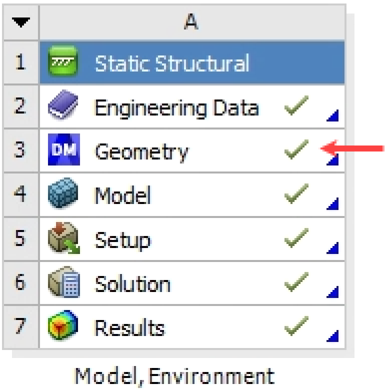
The following topics explain how to promote dimensions and solution results in Ansys DesignModeler and Mechanical to input parameters and output parameters:

Promoting Dimensions in DesignModeler to Input Parameters
Promoting a Property and Solution Results in Mechanical to Output Parameters

Promoting Dimensions in DesignModeler to Input Parameters

To promote dimensions in DesignModeler to input parameters, do the following:

1. On the **Project Schematic**, double-click the **Geometry** cell.



DesignModeler starts.

2. In the **Tree Outline**, select **Gusset**.

3. In the **Details View**, select the check box for **FD1, Depth (>0)** to promote it to an input parameter.

   A dialog box opens for specifying the parameter name.

4. Type **`thick_ds`** and click **OK**.

5. In the **Tree Outline** under **YZPlane**, select **Sketch4**.

6. In the **Details View** under **Dimensions**, select the check box for **H6** to promote it to an input parameter.

   A dialog box opens for specifying the parameter name.

7. Type **`horiz_ds`** and click **OK**.

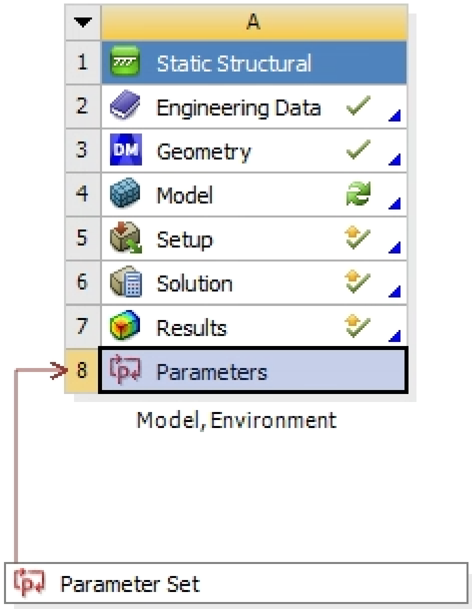
8. In the same **Details View** under **Dimensions**, select the check box for **V7** to promote it to an input parameter.

   A dialog box opens for specifying the parameter name.

9. Type **`vert_ds`** and click **OK**.

10. Exit DesignModeler.

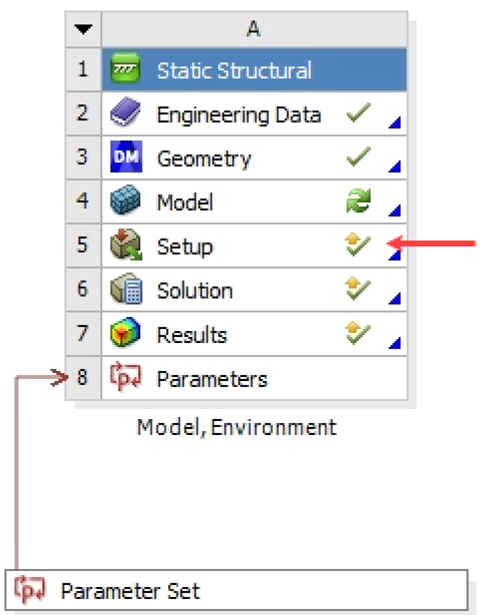
    The **Project Schematic** now includes the **Parameter Set** bar. An upward arrow leaves the **Parameter Set** bar and connects to the **Parameters** cell in the **Static Structural** system, indicating that input parameters have been defined.



Promoting a Property and Solution Results in Mechanical to Output Parameters

To promote a property and two solution results in Mechanical to output parameters, do the following:

1. On the **Project Schematic**, double-click the **Setup** cell.

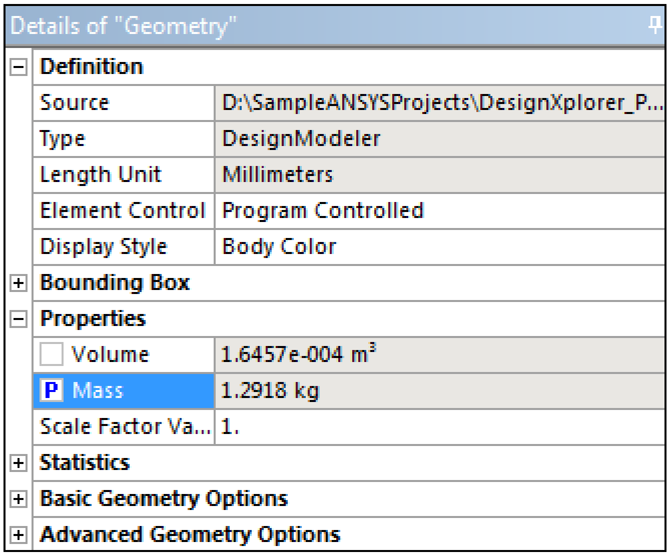


   A dialog box opens, indicating that upstream data must be re-read.

2. Click **Yes**. Mechanical starts.

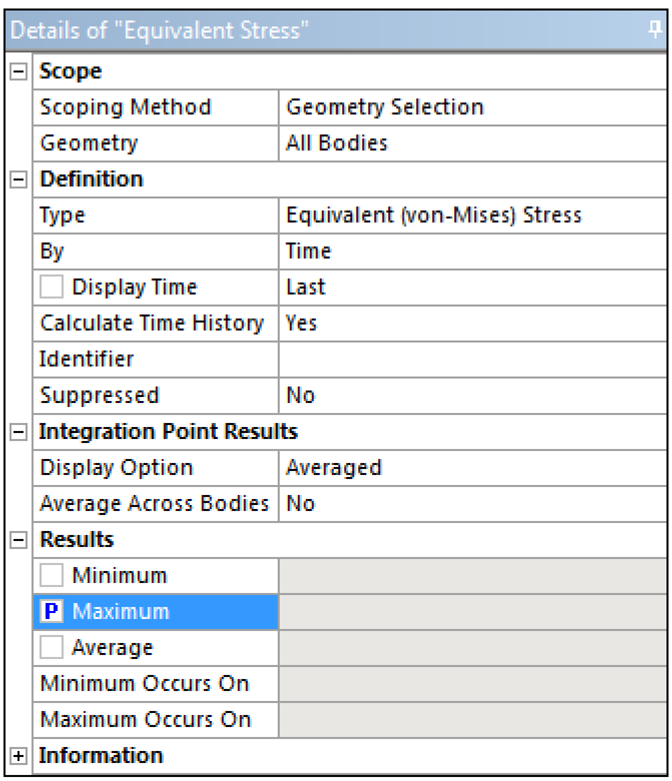
3. In the **Outline** view, select **Geometry**.

4. In the **Details** view under **Properties**, select the check box for **Mass** to promote the mass to an output parameter.



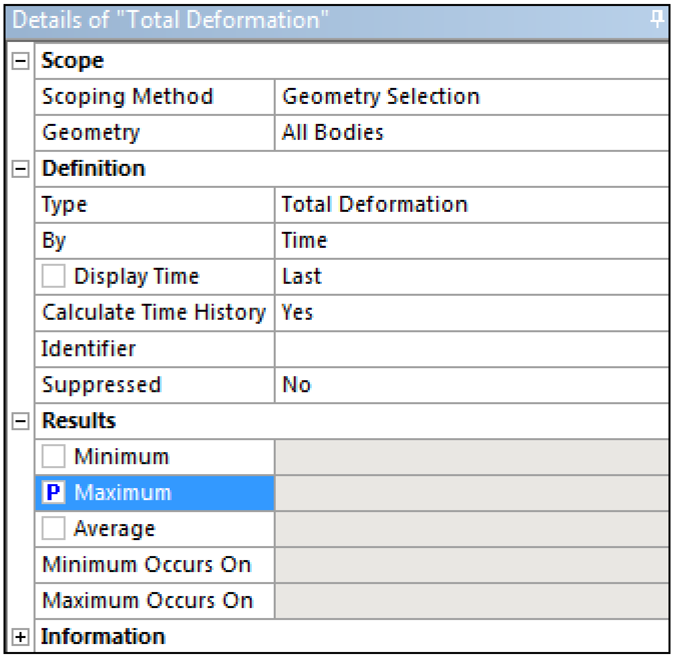
5. In the **Outline** view under **Solution**, select **Equivalent Stress**.

6. In the **Details** view under **Results**, select the check box for **Maximum** to promote the maximum equivalent stress to an input parameter.



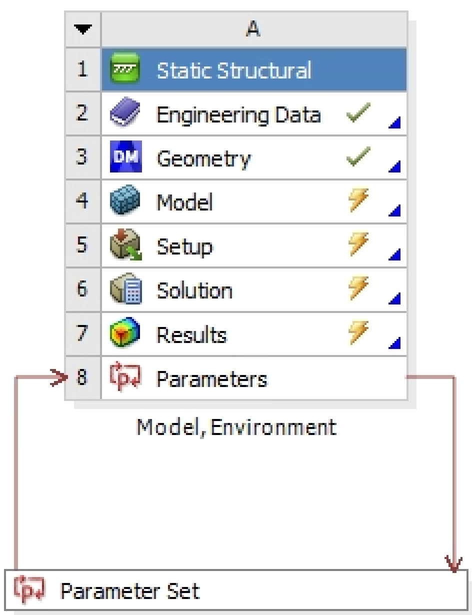
7. In the **Outline** view under **Solution**, select **Total Deformation**.

8. In the **Details** view under**Results**, select the check box for **Maximum** to promote the maximum total deformation to an output parameter.



9. Exit Mechanical.

   On the **Project Schematic**, a downward arrow now leaves the **Parameters** cell in the **Static Structural** system and connects to the **Parameter Set** bar, indicating that output parameters have been defined.



10. Update the project.

Working with Design Points in Workbench

After the project update completes, the **Parameter Set** bar contains a design point. You can look at the results for this design point and manually insert additional design points.

1. On the **Project Schematic**, double-click the **Parameter Set** bar to open it.

   The **Table** pane displays the design point that was generated.

Table of Design Points

	A	B	C	D	E	F	G	H	I	J
1	Name	P1 - thick_ds	P2 - horiz_ds	P3 - vert_ds	P4 - Geometry Mass	P5 - Equivalent Stress Maximum	P6 - Total Deformation Maximum	Retain	Retained Data	Note
2	Units	mm	mm	mm	kg	Pa	m			
3	DP 0 (Current)	2.5	30	30	1.2918	2.5492E+08	0.00033783	☑	✓	
*								☐		

2. Insert three additional design points as shown:

Table of Design Points

	A	B	C	D	E	F	G	H	I	J
1	Name	P1 - thick_ds	P2 - horiz_ds	P3 - vert_ds	P4 - Geometry Mass	P5 - Equivalent Stress Maximum	P6 - Total Deformation Maximum	☐ Retain	Retained Data	Note
2	Units	mm	mm	mm	kg	Pa	m			
3	DP 0 (Current)	2.5	30	30	1.2918	2.5492E+08	0.00033783	☑	✓	
4	DP 1	2.25	27	33	⚡	⚡	⚡	☐		
5	DP 2	2.75	33	30	⚡	⚡	⚡	☐		
6	DP 3	2.5	33	33	⚡	⚡	⚡	☐		
*								☐		

3. In the toolbar, click **Update All Design Points**.

   When the design point update is complete, the **Table** pane displays results for the added design points.

Table of Design Points

	A	B	C	D	E	F	G	H	I	J
1	Name	P1 - thick_ds	P2 - horiz_ds	P3 - vert_ds	P4 - Geometry Mass	P5 - Equivalent Stress Maximum	P6 - Total Deformation Maximum	☐ Retain	Retained Data	Note
2	Units	mm	mm	mm	kg	Pa	m			
3	DP 0 (Current)	2.5	30	30	1.2918	2.5492E+08	0.00033783	☑	✓	
4	DP 1	2.25	27	33	1.2896	2.9583E+08	0.00034853	☐		
5	DP 2	2.75	33	30	1.296	2.6628E+08	0.0003145	☐		
6	DP 3	2.5	33	33	1.2958	2.2858E+08	0.00030377	☐		
*								☐		

Adding Parameter Charts

In the **Parameters Set** bar, you can add charts. The follow topics explain how to add a Parameter chart and a Parameters Parallel chart:

Adding a Parameter Chart

Adding a Parameters Parallel Chart

Adding a Parameter Chart

First, you'll add a Parameter chart, which illustrates the relationship between a selected parameter and each of the design points.

Note:

If you do not see the **Chart** pane in the bottom right of the workspace, select **View** → **Chart**.

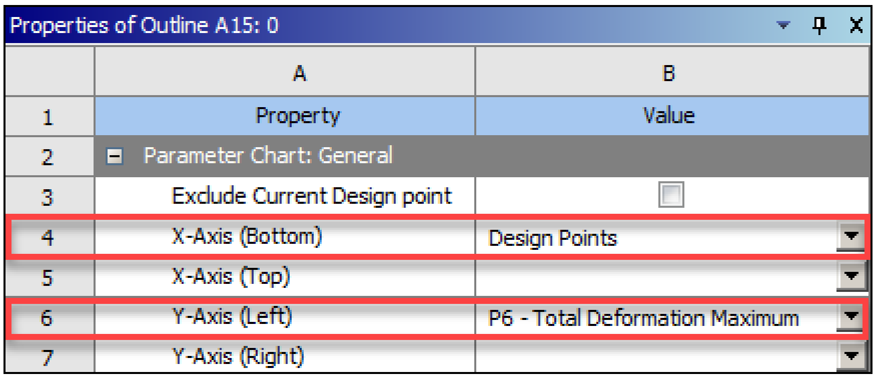
1. In the **Outline** pane under **Output Parameters**, select **P6 — Total Deformation Maximum**. Currently, lower in the **Outline** pane under **Charts**, no chart instances exist.

2. If the **Toolbox** is not open, select **View** → **Toolbox** to open it. The **Toolbox** displays the Parameter charts that can be inserted.

3. In the **Toolbox**, double-click **Design Points vs P6** to add this chart.

4. In the **Outline** pane under **Charts**, select **Parameter Chart 0**.

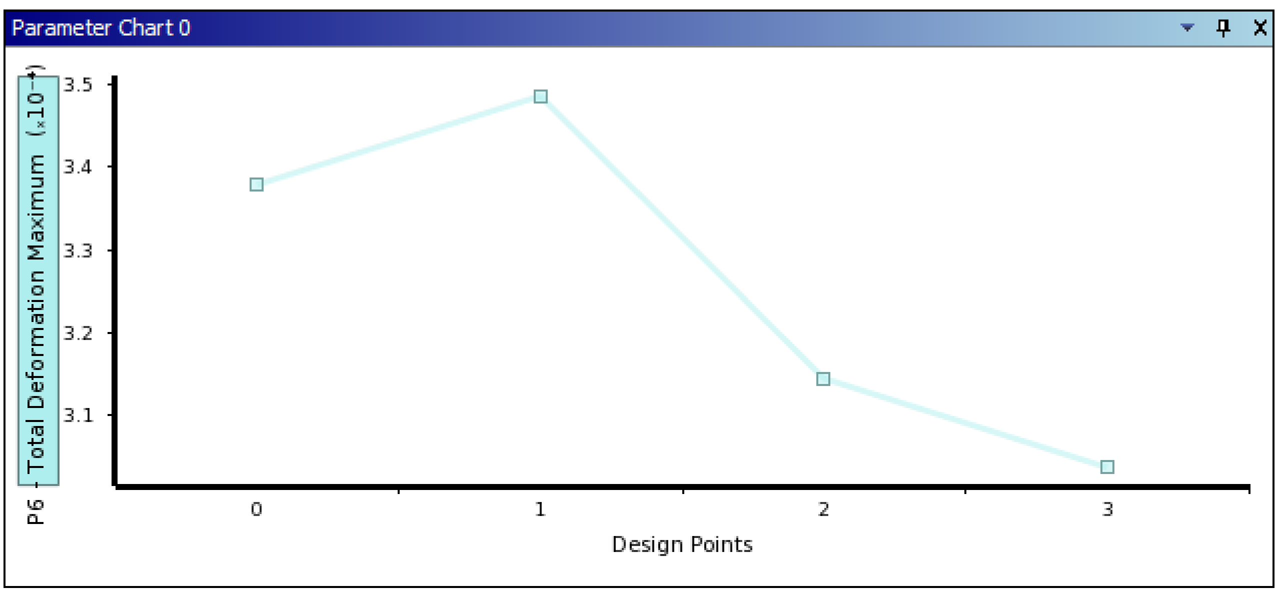
   In the **Properties** pane, **X-Axis (Bottom)** is set to **Design Points** and **Y-Axis (Left)** is set to **P6 — Total Deformation Maximum**:



Properties of Outline A15: 0

	A	B
1	Property	Value
2	Parameter Chart: General	
3	Exclude Current Design point	☐
4	X-Axis (Bottom)	Design Points
5	X-Axis (Top)	
6	Y-Axis (Left)	P6 - Total Deformation Maximum
7	Y-Axis (Right)	

   The **Graph** pane displays this chart. The design points are arranged on the X axis on the bottom of the chart. The **P6** values are arranged on the Y axis on the left side of the chart.



Adding a Parameters Parallel Chart

Next, you'll add a Parameters Parallel chart, which illustrates the relationship between all parameters and each of the design points.

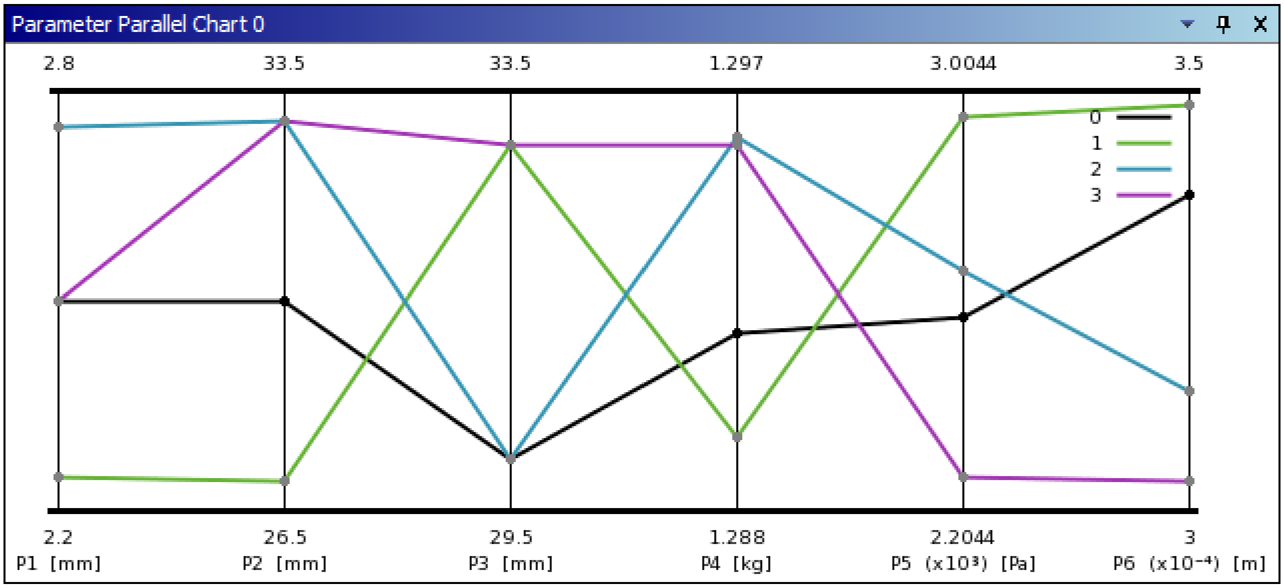
1. In the **Outline** pane, select any input or output parameter.

   Because the Parameters Parallel chart plots parallel Y axes for all input and out parameters, any parameter can be selected.

2. In the **Toolbox**, double-click **Parameters Parallel Chart (all)** to add this chart.

3. In the **Outline** pane under **Charts**, select **Parameter Parallel Chart 0**.

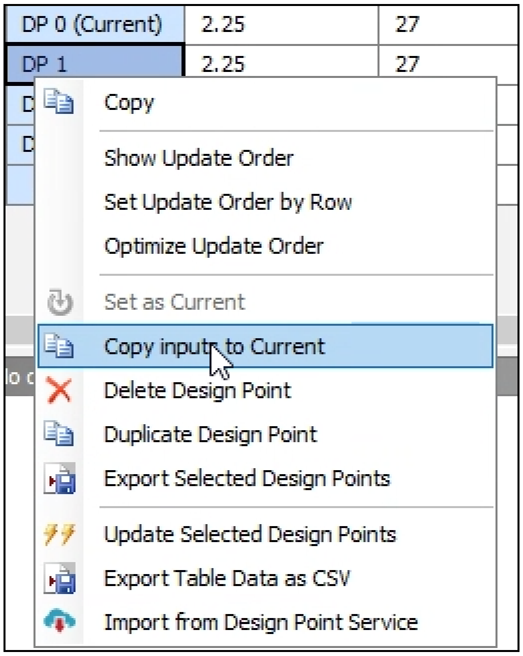
   In the **Graph** pane, the Parameter Parallel chart displays all input and output parameters along the X axis on the bottom of the chart. The top and bottom values indicate the range relative to each parameter. Each color-coded line on the plot represents a design point.



Viewing the Solution

A solution is generated for the design point designated as **(Current)** in the **Table** pane. If, based on your design exploration, you decide that another design point is preferable, you can define it as the current design point. In this topic, you'll copy the inputs for **DP 1** to the current design point.

1. In the **Table** pane, right-click **DP 1** and select **Copy inputs to Current**.



The values for **DP 1** are copied to **DP 0**, and **DP 0** becomes outdated.

Table of Design Points

	A	B	C	D	E	F	G	H	I	J
1	Name	P1 - thick_ds	P2 - horiz_ds	P3 - vert_ds	P4 - Geometry Mass	P5 - Equivalent Stress Maximum	P6 - Total Deformation Maximum	Retain	Retained Data	Note
2	Units	mm	mm	mm	kg	Pa	m			
3	DP 0 (Current)	2.25	27	33	1.2918	2.5492E+08	0.00033783	☑	✓	
4	DP 1	2.25	27	33	1.2896	2.9583E+08	0.00034853	☐		
5	DP 2	2.75	33	30	1.296	2.6628E+08	0.0003145	☐		
6	DP 3	2.5	33	33	1.2958	2.2858E+08	0.00030377	☐		
*								☐		

2. Right-click **DP 0** and select **Update Selected Design Points**.

   When the update is complete, in the **Table** pane, the results for **DP 0** now match those for **DP 1**.

Table of Design Points

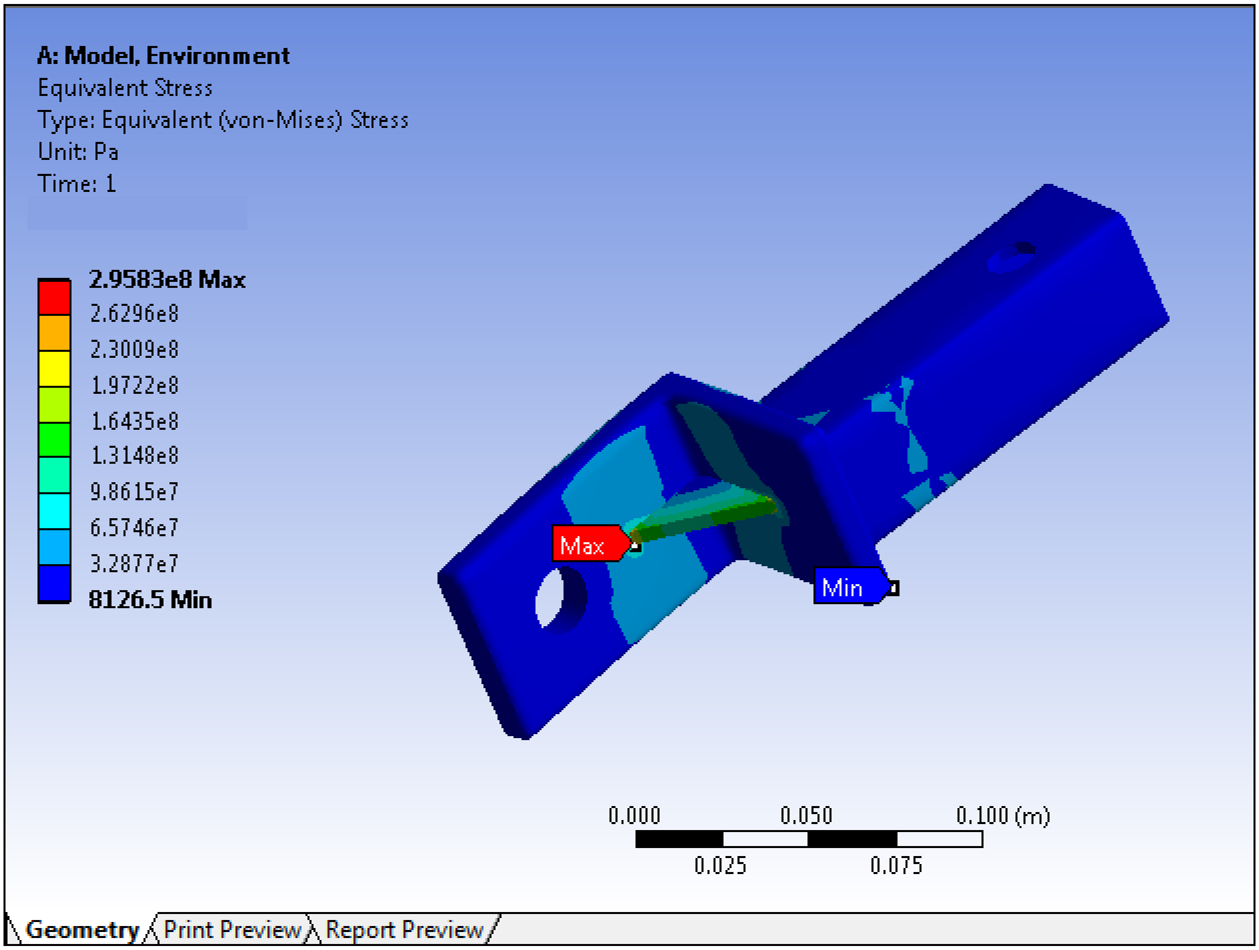
	A	B	C	D	E	F	G	H	I	J
1	Name	P1 - thick_ds	P2 - horiz_ds	P3 - vert_ds	P4 - Geometry Mass	P5 - Equivalent Stress Maximum	P6 - Total Deformation Maximum	Retain	Retained Data	Note
2	Units	mm	mm	mm	kg	Pa	m			
3	DP 0 (Current)	2.25	27	33	1.2896	2.9583E+08	0.00034853	☑	✓	
4	DP 1	2.25	27	33	1.2896	2.9583E+08	0.00034853	☐		
5	DP 2	2.75	33	30	1.296	2.6628E+08	0.0003145	☐		
6	DP 3	2.5	33	33	1.2958	2.2858E+08	0.00030377	☐		
*								☐		

3. Close the **Parameter Set** bar.

4. On the **Project Schematic**, double-click the **Model** cell. Mechanical starts.

5. In the **Outline** view under **Solution**, select **Equivalent Stress** to view the solution for the current design point.

   In the following figure, the hitch receiver is rotated so that you can clearly see the locations for the maximum and minimum equivalent stress.



6. Exit Mechanical.

7. Exit Workbench, saving project changes.

Performing a Parameters Correlation

This tutorial gives step-by-step instructions for performing a parameters correlation to calculate the deformation of a simple cantilever beam under applied force. It uses a DesignXplorer system that is linked to a Mechanical APDL file to demonstrate parameter parsing and selection and how to show results.

This tutorial covers the following topics:

What is a Parameters Correlation?

Getting Started

Defining Parameters

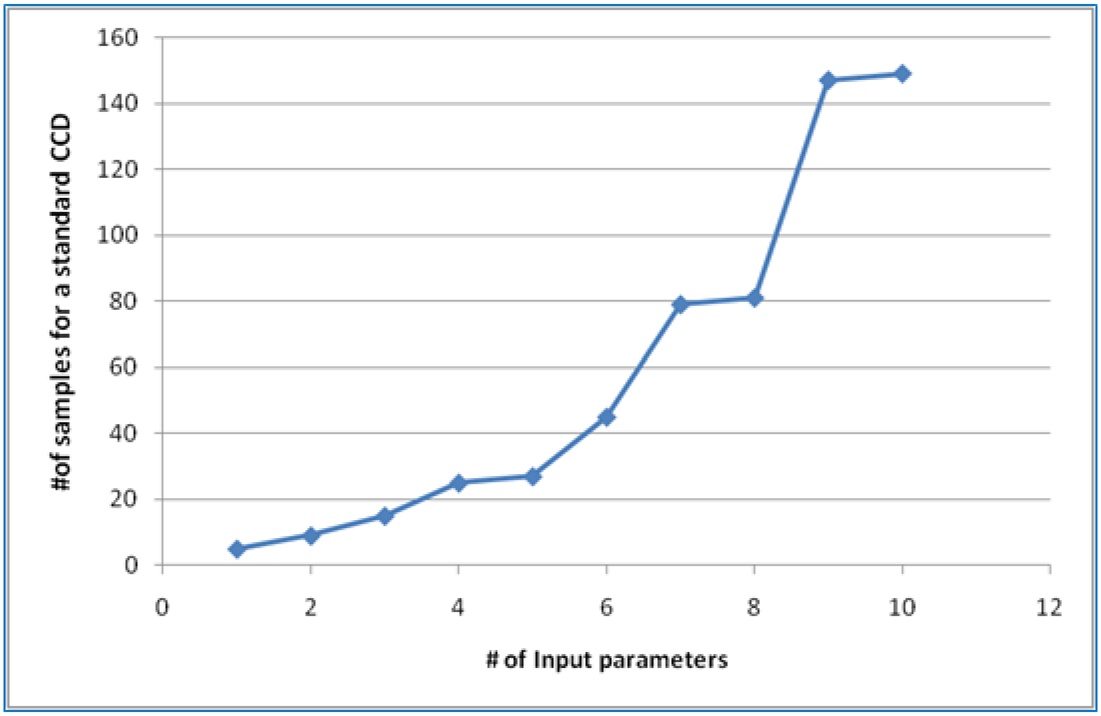
Generating Design Points

Performing the Parameters Correlation

Viewing the Correlation Matrix and Charts

What is a Parameters Correlation?

As you add more input parameters to a DOE, the number of design points increases dramatically, decreasing the efficiency of the analysis process. Because DesignXplorer works best with fewer than 20 parameters, you might want to exclude inputs that are not actively contributing toward your intended design. Removing less important parameters from a DOE reduces the generation of unnecessary sampling points.



Benefits

A parameters correlation allows you to:

- Determine which input parameters have the most (and the least) impact on your design.
- Identify the degree to which the relationship is linear or quadratic.

Visual Assessment

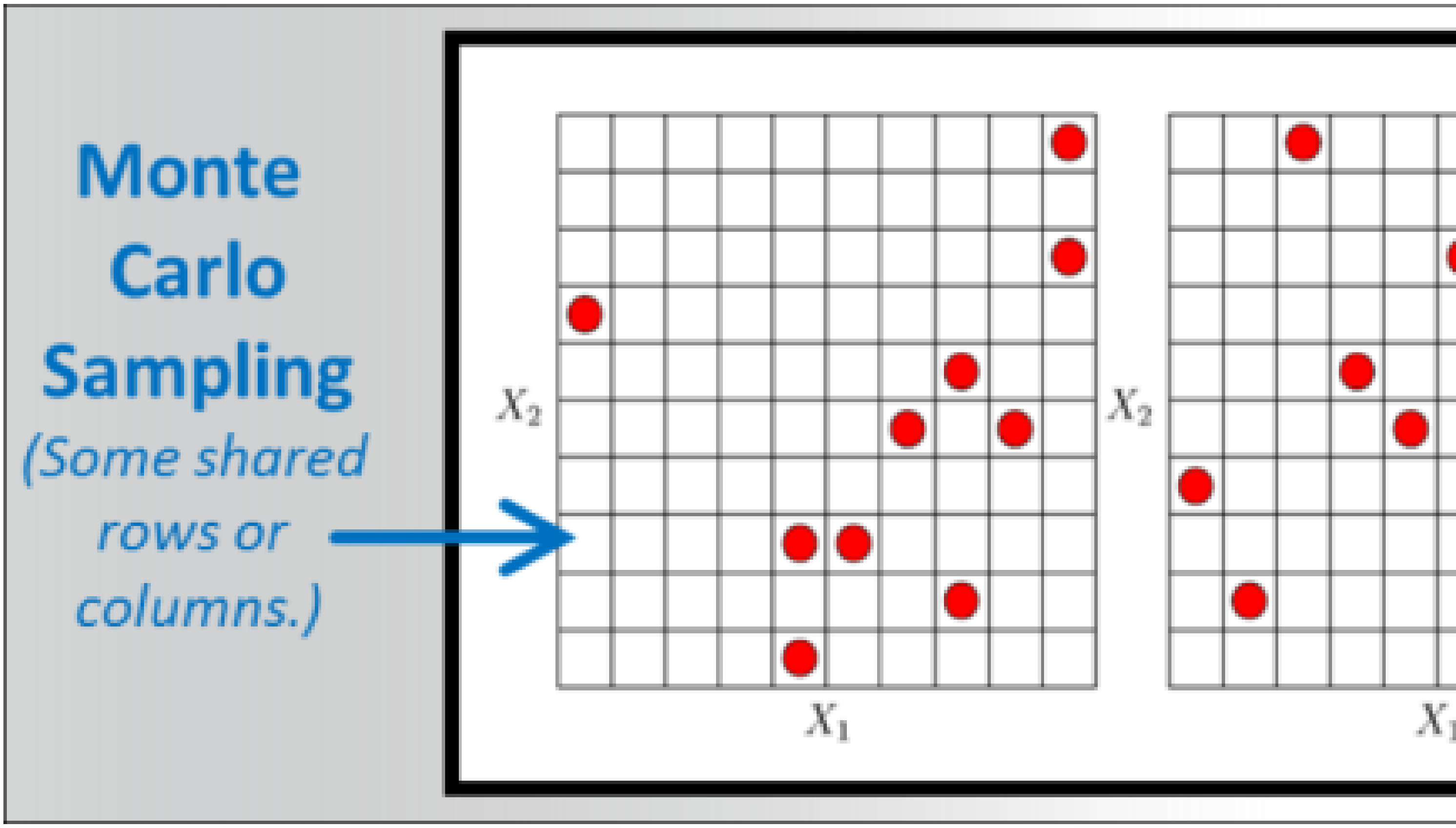
For a parameters correlation, DesignXplorer provides the following charts so that you can visually assess parametric impacts:

- Correlation matrix and chart
- Determination matrix and chart
- Correlation scatter chart
- Sensitivity chart

Sampling

A parameters correlation performs simulations on a random sampling of the design space to identify correlations between all parameters.

DesignXplorer uses the *Latin Hypercube Sampling (LHS)* method to generate samples. This means that the points are randomly generated but no two points share input parameters of the same value. The following figure shows how samples generated by LHS vary in placement from those generated by Monte Carlo sampling.



Correlation Methods

For a parameters correlation, DesignXplorer can use either the Pearson's linear correlation method or the Spearman's rank correlation method.

Pearson's Linear Correlation Method

- Uses actual data for correlation evaluation.
- Correlation coefficients are based on the sample values.

height (H)	Stress (S)
8.0357473	42.049729
7.9530498	41.707272
8.5795715	36.684773
8.2932448	37.968797
7.6854265	40.610932

- Used to correlate linear relationships.

Spearman's Rank Correlation Method

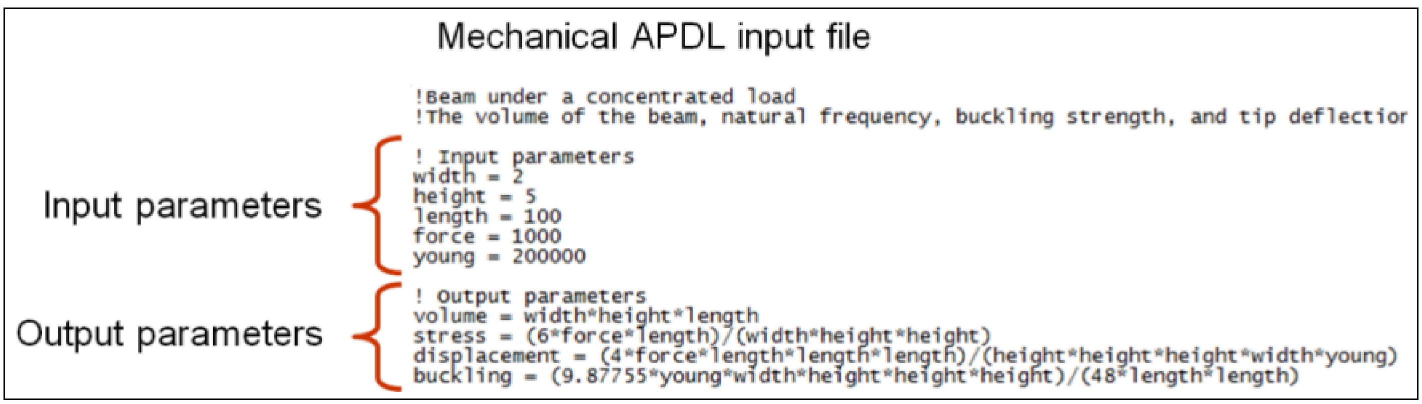
- Uses ranks of data.
- Correlation coefficients are based on the rank of samples.

height rank	Stress Rank
3	1
4	2
1	5
2	4
5	3

- Recognizes non-linear *monotonic* relationships, which are less restrictive than linear ones. In a monotonic relationship, one of the following occurs:
  - As the value of one variable increases, the value of the other variable also increases.
  - As the value of one variable increases, the value of the other variable decreases.
- Deemed the more accurate method.

Getting Started

This tutorial uses a textbook example of a simple model of a cantilever beam to calculate deformation under an applied force. You will add this model to the **Project Schematic** by importing a Mechanical APDL input file, which the following figure explains:



1. Start Workbench.
2. In the **Toolbox** under **Component Systems**, double-click **Mechanical APDL** to add this system to the **Project Schematic**.
3. On the **Project Schematic**, right-click the **Analysis** cell and select **Add Input File** → **Browse**.
4. In the file browser, locate and open the file `BeamEquations.inp`, which is in the directory to which you extracted the input files (p. 27).
5. Update the project.
6. Save the project as `PCBeam.wbpj` in either this same directory or another directory.

Defining Parameters

1. On the **Project Schematic**, double-click the **Analysis** cell to open it.
2. In the **Outline** pane, select the **Process "BeamEquations.inp"** cell.
3. In the **Properties** pane, identify input parameters and then output parameters as shown:

Properties: No data

	A	B	C	D
1	APDL Parameter	Initial Value	Input	Output
2	WIDTH	2	☑	☐
3	HEIGHT	5	☑	☐
4	LENGTH	100	☑	☐
5	FORCE	1000	☑	☐
6	YOUNG	2E+05	☑	☐
7	VOLUME		☐	☑
8	STRESS		☐	☑
9	DISPLACEMENT		☐	☑
10	BUCKLING		☐	☑

4. Close the **Analysis** cell.

Generating Design Points

The **Project Schematic** now includes the **Parameter Set** bar. Next, you'll generate a design point.

1. Double-click the **Parameter Set** bar to open it.

   The **Table** pane displays one design point for which results have not yet been calculated.

2. In the toolbar, click **Update All Design Points**.

   When the update is complete, the **Table** pane displays results for **DP 0**, which is the current design point.

Table of Design Points

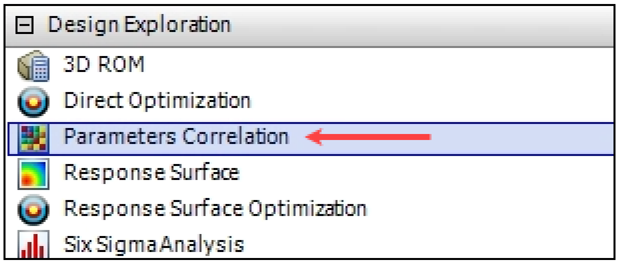
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Name	P1 - WIDTH	P2 - HEIGHT	P3 - LENGTH	P4 - FORCE	P5 - YOUNG	P6 - VOLUME	P7 - STRESS	P8 - DISPLACEMENT	P9 - BUCKLING	Retain	Retained Data	Note
2	DP 0 (Current)	2	5	100	1000	2E+05	1000	12000	80	1028.9	☑	✓	
*											☐		

3. Close the **Parameter Set** bar.

Performing the Parameters Correlation

You'll now add a **Parameters Correlation** system and then set properties and the lower and upper bounds for the input parameters.

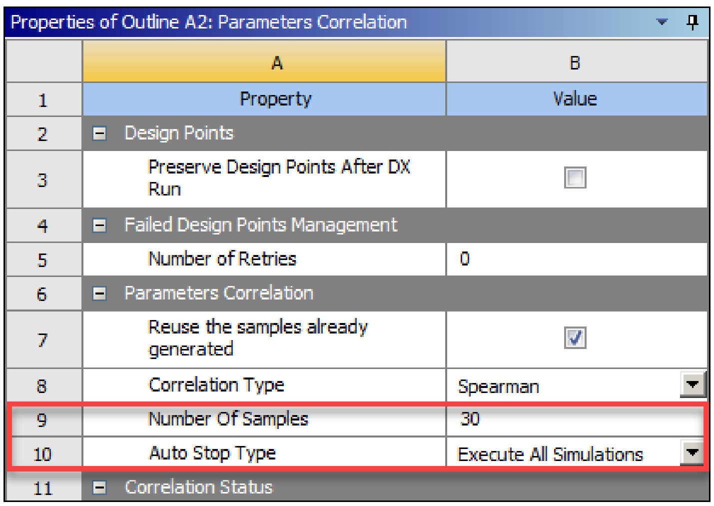
1. In the **Toolbox** under **Design Exploration**, double-click **Parameters Correlation** to add this system to the **Project Schematic**.



2. In the new **Parameters Correlation** system, double-click the **Parameters Correlation** cell to open it.

3. In the **Outline** pane, select **Parameters Correlation**.

4. In the **Properties** pane, set the properties as shown:

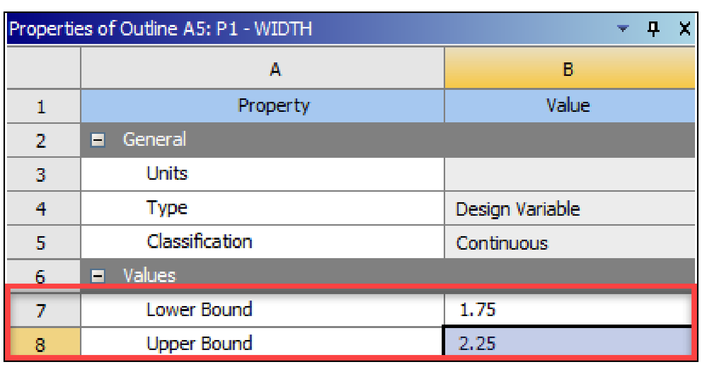


Properties of Outline A2: Parameters Correlation

	A	B
1	Property	Value
2	Design Points	
3	Preserve Design Points After DX Run	☐
4	Failed Design Points Management	
5	Number of Retries	0
6	Parameters Correlation	
7	Reuse the samples already generated	☑
8	Correlation Type	Spearman
9	Number Of Samples	30
10	Auto Stop Type	Execute All Simulations
11	Correlation Status	

5. In the **Outline** pane, select the input parameter **P1–WIDTH**.

6. In the **Properties** pane under **Values**, set the lower and upper bounds as shown:



Properties of Outline A5: P1 - WIDTH

	A	B
1	Property	Value
2	General	
3	Units	
4	Type	Design Variable
5	Classification	Continuous
6	Values	
7	Lower Bound	1.75
8	Upper Bound	2.25

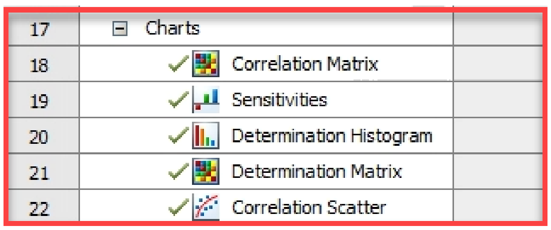
7. Following the process outlined in steps 5 through 6, set the lower and upper bounds for the other input parameters:

- For **P2—HEIGHT**, set **Lower Bound** to **4.5** and **Upper Bound** to **5.5**. (The bounds for this input parameter are already set.)
- For **P3—LENGTH**, set **Lower Bound** to **80** and **Upper Bound** to **120**.
- For **P4—FORCE**, set **Lower Bound** to **800** and **Upper Bound** to **1200**.
- For **P5—YOUNG**, set **Lower Bound** to **1.8E+05** and **Upper Bound** to **2E+05**. (The bounds for this input parameter are already set.)

8. Update the **Parameters Correlation** cell. The update may take a few minutes.

Viewing the Correlation Matrix and Charts

Once the update is complete, in the **Outline** pane under **Charts**, you can see that default charts have been generated:



When you select a chart in the **Outline** pane, the **Table** pane displays the chart data. The **Chart** pane displays the chart itself.

Tip:

To add another instance of a chart, select **View** → **Toolbox** to open the **Toolbox** and then double-click the type of chart to add. You will then see this other instance in the **Outline** pane under **Charts**.

Descriptions follow for these **Parameters Correlation** charts: Correlation Matrix, Sensitivities, Correlation Scatter, and Determination Histogram. For comprehensive information on all **Parameters Correlation** charts, see Working with Parameters Correlation Charts in the *DesignXplorer User's Guide*

Correlation Matrix

When **Correlation Matrix** is the chart selected in the **Outline** pane, the **Table** pane displays correlation coefficients. A correlation coefficient indicates if there is a relationship between two variables and indicates whether the relationship is positive or negative. The **Graph** pane displays the Correlation Matrix.

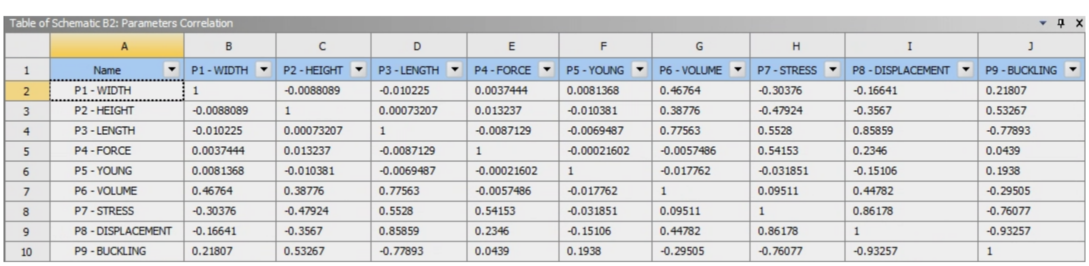


Table of Schematic B2: Parameters Correlation

	A	B	C	D	E	F	G	H	I	J
1	Name	P1 - WIDTH	P2 - HEIGHT	P3 - LENGTH	P4 - FORCE	P5 - YOUNG	P6 - VOLUME	P7 - STRESS	P8 - DISPLACEMENT	P9 - BUCKLING
2	P1 - WIDTH	1	-0.0088089	-0.010225	0.0037444	0.0081368	0.46764	-0.30376	-0.16641	0.21807
3	P2 - HEIGHT	-0.0088089	1	0.00073207	0.013237	-0.010381	0.38776	-0.47924	-0.3567	0.53267
4	P3 - LENGTH	-0.010225	0.00073207	1	-0.0087129	-0.0069487	0.77563	0.5528	0.85859	-0.77893
5	P4 - FORCE	0.0037444	0.013237	-0.0087129	1	-0.00021602	-0.0057486	0.54153	0.2346	0.0439
6	P5 - YOUNG	0.0081368	-0.010381	-0.0069487	-0.00021602	1	-0.017762	-0.031851	-0.15106	0.1938
7	P6 - VOLUME	0.46764	0.38776	0.77563	-0.0057486	-0.017762	1	0.09511	0.44782	-0.29505
8	P7 - STRESS	-0.30376	-0.47924	0.5528	0.54153	-0.031851	0.09511	1	0.86178	-0.76077
9	P8 - DISPLACEMENT	-0.16641	-0.3567	0.85859	0.2346	-0.15106	0.44782	0.86178	1	-0.93257
10	P9 - BUCKLING	0.21807	0.53267	-0.77893	0.0439	0.1938	-0.29505	-0.76077	-0.93257	1

In this Correlation Matrix, you can see that input parameter **P3–LENGTH** is a major input because it drives all the outputs, particularly **P8–DISPLACEMENT** and **P6–VOLUME**.

On the other hand, input parameter **P5–YOUNG** is not important because it has little impact on the outputs. In this case, you might choose to disable **P5–YOUNG** by deselecting its check box in the **Properties** pane. If you do this, a dialog box will open, indicating that this parameter change will clear all generated data in this system. If you click **Yes** and update the **Parameters Correlation** cell, all charts are updated accordingly. However, the results shown for this parameters correlation assumes that you click **No** so that results are based on the originally generated data.

Sensitivities Chart

When **Sensitivities** is the chart selected in the **Outline** pane, the **Graph** pane shows global sensitivities of the output parameters with respect to the input parameters. Positive sensitivity occurs when increasing the input increases the output. Negative sensitivity occurs when increasing the input decreases the the output.

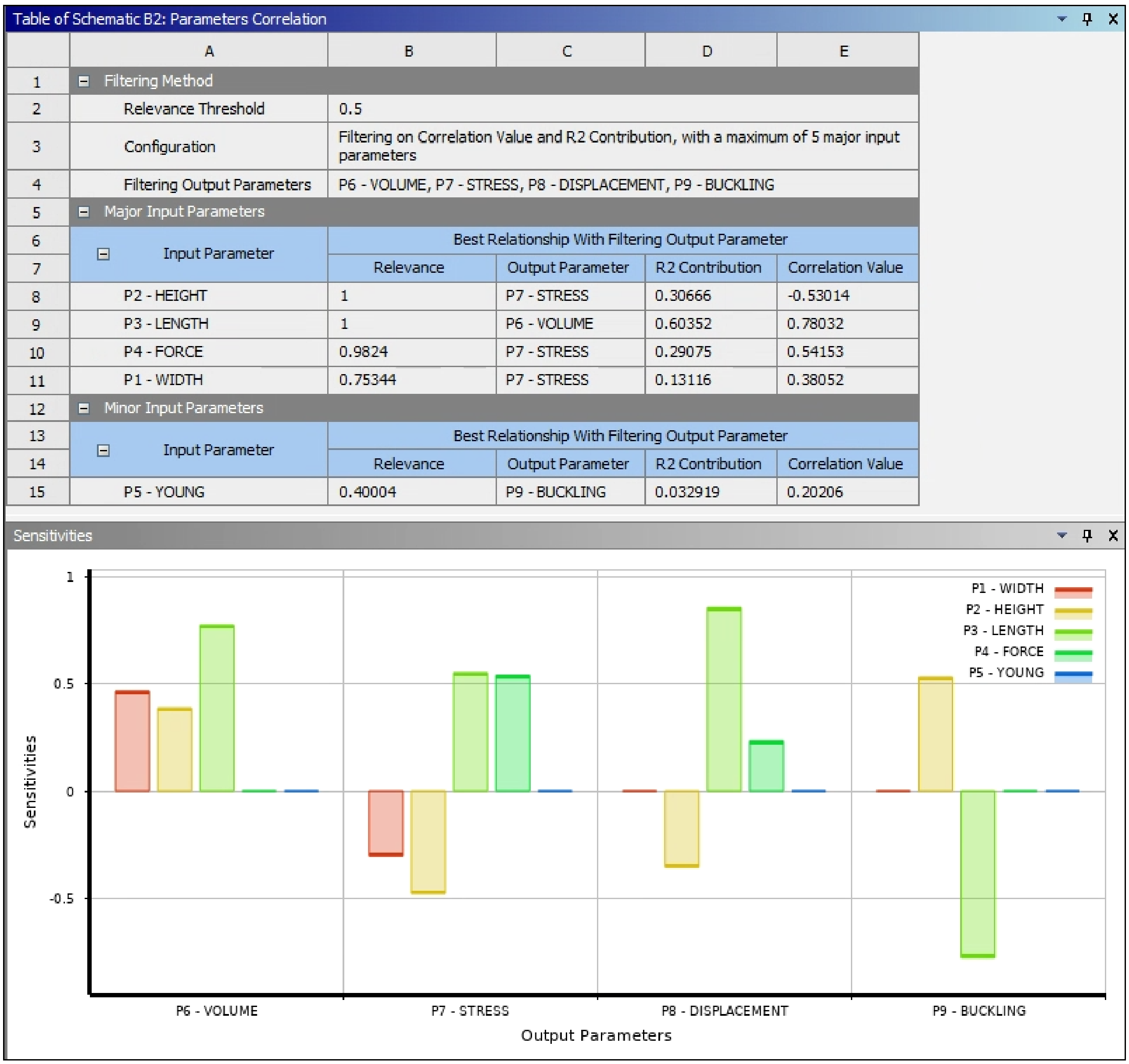
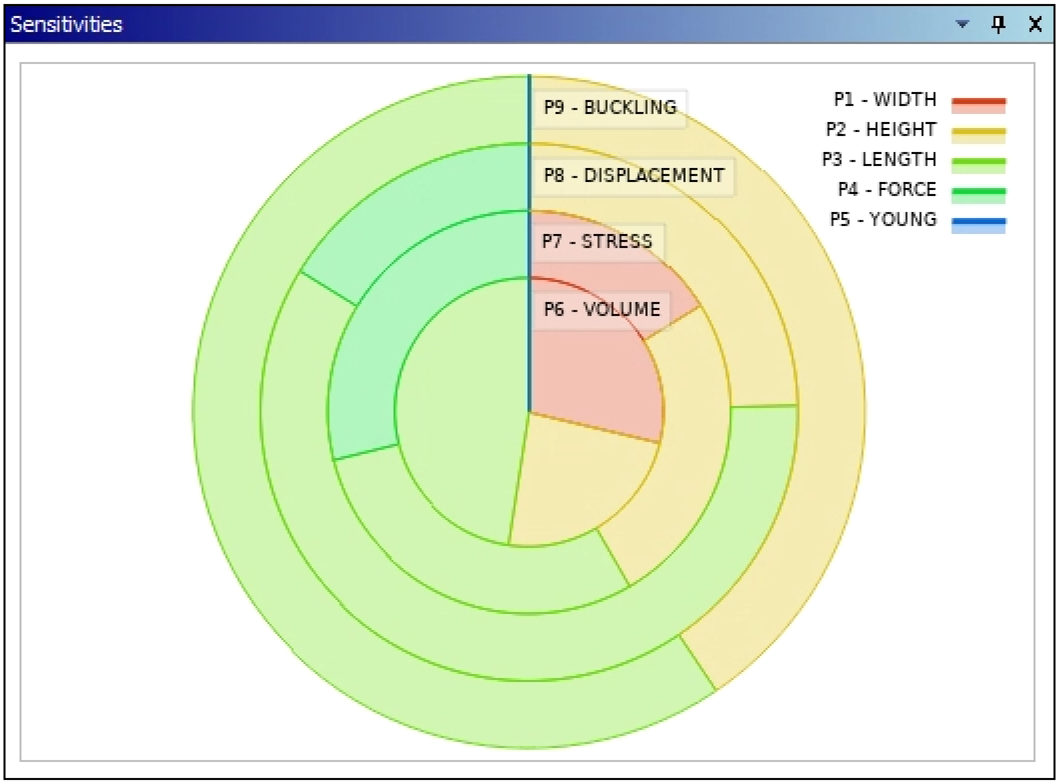


Table of Schematic B2: Parameters Correlation

	A	B	C	D	E
1	⊟ Filtering Method				
2	Relevance Threshold	0.5			
3	Configuration	Filtering on Correlation Value and R2 Contribution, with a maximum of 5 major input parameters			
4	Filtering Output Parameters	P6 - VOLUME, P7 - STRESS, P8 - DISPLACEMENT, P9 - BUCKLING			
5	⊟ Major Input Parameters				
6	⊟ Input Parameter	Best Relationship With Filtering Output Parameter			
7		Relevance	Output Parameter	R2 Contribution	Correlation Value
8	P2 - HEIGHT	1	P7 - STRESS	0.30666	-0.53014
9	P3 - LENGTH	1	P6 - VOLUME	0.60352	0.78032
10	P4 - FORCE	0.9824	P7 - STRESS	0.29075	0.54153
11	P1 - WIDTH	0.75344	P7 - STRESS	0.13116	0.38052
12	⊟ Minor Input Parameters				
13	⊟ Input Parameter	Best Relationship With Filtering Output Parameter			
14		Relevance	Output Parameter	R2 Contribution	Correlation Value
15	P5 - YOUNG	0.40004	P9 - BUCKLING	0.032919	0.20206

In the **Properties** pane for a Sensitivities chart, you can set **Mode** to either **Bar** (default) or **Pie**. The following figure shows the same Sensitivities chart in **Pie** mode.



Generally, the impact of an input parameter on an output parameter is driven by two things:

- Amount by which the output parameter varies across the variation range of an input parameter.
- Variation range of an input parameter. Typically, the wider the variation range is, the larger the impact of the input parameter.

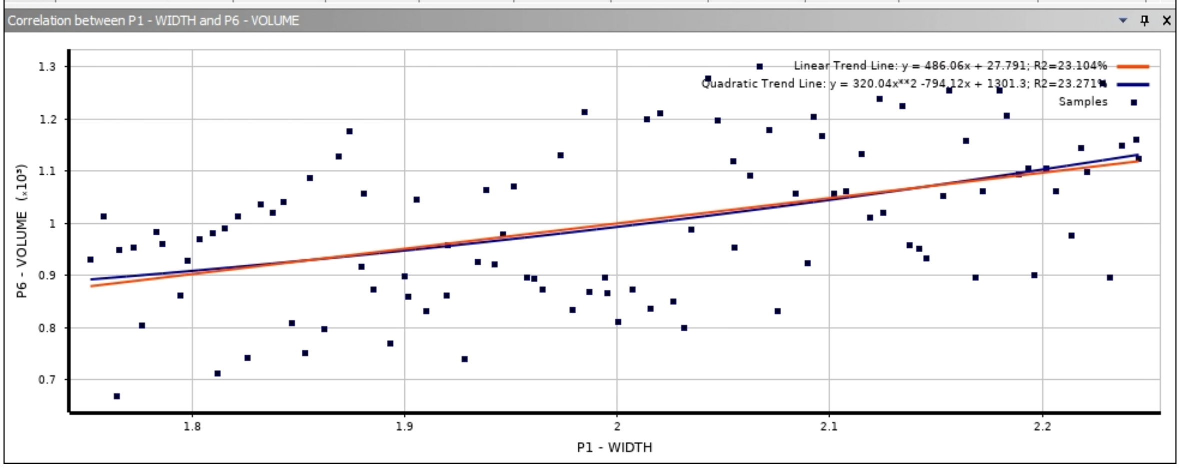
Statistical sensitivities are based on the Spearman-Rank Order correlation coefficients that simultaneously take both aspects into account.

Correlation Scatter Chart

When **Correlation Scatter** is the chart selected in the **Outline** pane, the **Table** pane displays the linear and quadratic coefficient of determination (R^2) values. The **Graph** pane displays either or both linear and quadratic trend lines.

Table of Schematic B2: Parameters Correlation

	A	B	C	D	E	F	G	H	I	J
1	Name	P1 - WIDTH	P2 - HEIGHT	P3 - LENGTH	P4 - FORCE	P5 - YOUNG	P6 - VOLUME	P7 - STRESS	P8 - DISPLACEMENT	P9 - BUCKLING
2	⊟ Correlation Matrix (Linear)									
3	P1 - WIDTH	1	-0.0088089	-0.010225	0.0037444	0.0081368	0.46764	-0.30376	-0.16641	0.21807
4	P2 - HEIGHT	-0.0088089	1	0.00073207	0.013237	-0.010381	0.38776	-0.47924	-0.3567	0.53267
5	P3 - LENGTH	-0.010225	0.00073207	1	-0.0087129	-0.0069487	0.77563	0.5528	0.85859	-0.77893
6	P4 - FORCE	0.0037444	0.013237	-0.0087129	1	-0.00021602	-0.0057486	0.54153	0.2346	0.0439
7	P5 - YOUNG	0.0081368	-0.010381	-0.0069487	-0.00021602	1	-0.017762	-0.031851	-0.15106	0.1938
8	P6 - VOLUME	0.46764	0.38776	0.77563	-0.0057486	-0.017762	1	0.09511	0.44782	-0.29505
9	P7 - STRESS	-0.30376	-0.47924	0.5528	0.54153	-0.031851	0.09511	1	0.86178	-0.76077
10	P8 - DISPLACEMENT	-0.16641	-0.3567	0.85859	0.2346	-0.15106	0.44782	0.86178	1	-0.93257
11	P9 - BUCKLING	0.21807	0.53267	-0.77893	0.0439	0.1938	-0.29505	-0.76077	-0.93257	1
12	⊟ Determination Matrix (Quadratic)									
13	P1 - WIDTH	1	0.0030728	0.010601	0.0057427	0.0034215	0.23271	0.1448	0.064529	0.066967
14	P2 - HEIGHT	0.0049689	1	0.0064632	0.0011573	0.0050364	0.14797	0.28106	0.15902	0.32874
15	P3 - LENGTH	0.00054381	0.0017389	1	0.0016544	0.00080713	0.60931	0.30105	0.69415	0.55613
16	P4 - FORCE	0.0096895	0.0017813	0.036023	1	0.0076607	0.005474	0.29951	0.11372	0.031747



For this particular Correlation Scatter chart, in the **Properties** pane under **Axes**, **X Axis** is set to **P1–WIDTH** and **Y Axis** is set to **P6–VOLUME**.

Properties of Outline : Correlation Scatter

	A	B	C
1	Property	Value	Enabled
2	⊟ Chart		
3	Display Parameter Full Name	☑	
4	⊟ Axes		
5	X Axis	P1 - WIDTH	
6	Y Axis	P6 - VOLUME	
7	⊟ Trend Lines		
8	Linear	y = 486.06x + 27.791; R2=23 .104%	☑
9	Quadratic	y = 320.04x**2 -794.12x + 1301 .3; R2=23.271%	☑

Under **Trend Lines**, you can see the **Linear** and **Quadratic** values for $\mathbf{R^2}$. Because both the linear and quadratic tread lines are enabled, the chart displays both lines. The closer the samples lie to the curve, the closer the $\mathbf{R^2}$ value is to the optimum value of **1**.

Determination Histogram

When **Determination Histogram** is the chart selected in the **Outline** pane, the **Graph** pane shows you what input parameters drive a selected output parameter.

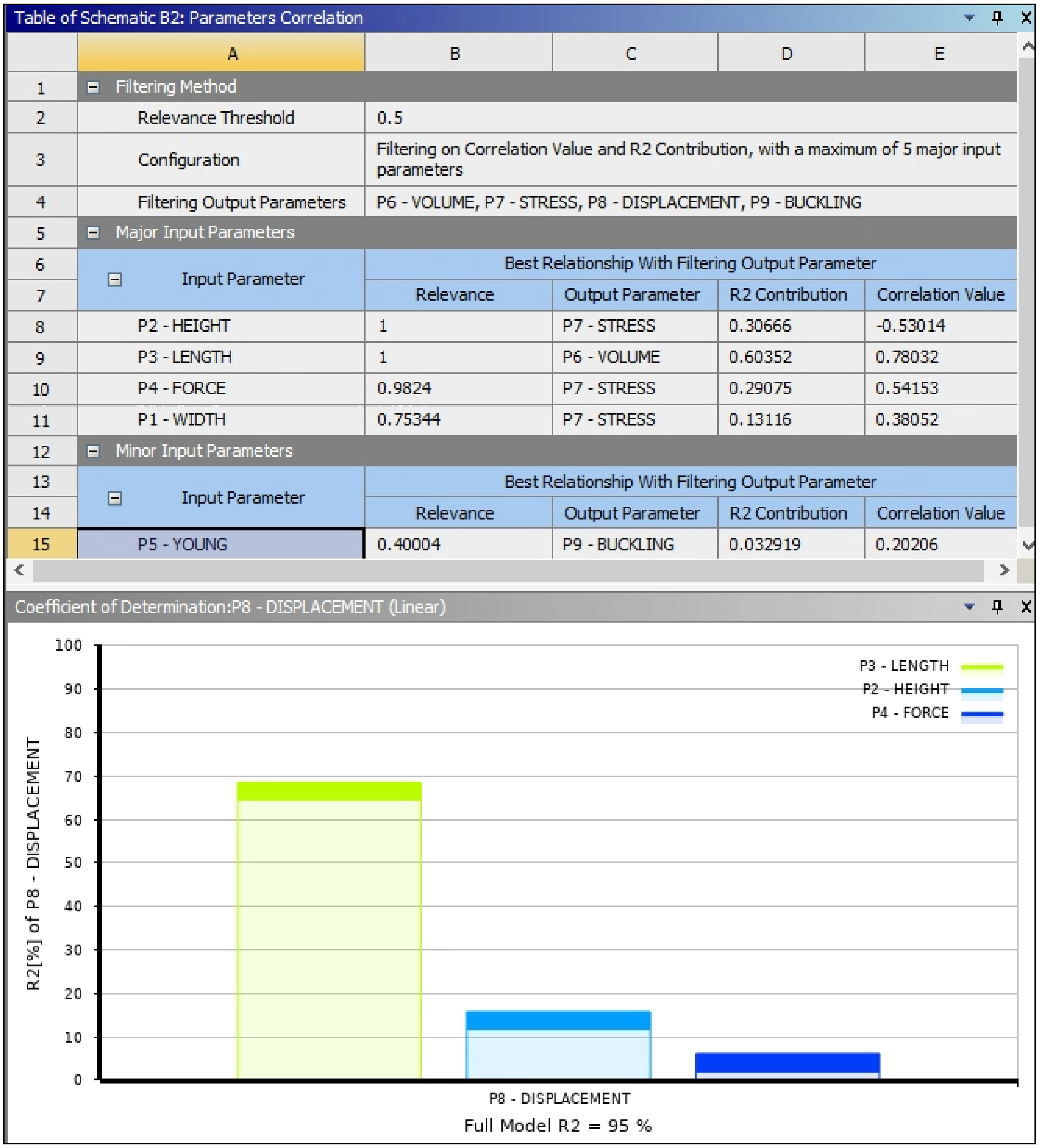
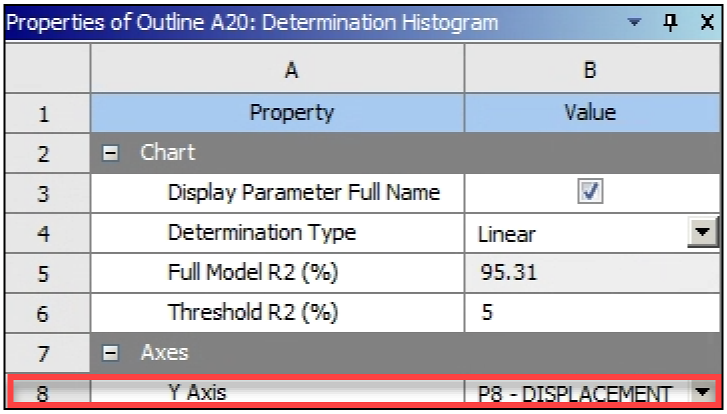


Table of Schematic B2: Parameters Correlation

	A	B	C	D	E
1	Filtering Method				
2	Relevance Threshold	0.5			
3	Configuration	Filtering on Correlation Value and R2 Contribution, with a maximum of 5 major input parameters			
4	Filtering Output Parameters	P6 - VOLUME, P7 - STRESS, P8 - DISPLACEMENT, P9 - BUCKLING			
5	Major Input Parameters				
6	Input Parameter	Best Relationship With Filtering Output Parameter			
7		Relevance	Output Parameter	R2 Contribution	Correlation Value
8	P2 - HEIGHT	1	P7 - STRESS	0.30666	-0.53014
9	P3 - LENGTH	1	P6 - VOLUME	0.60352	0.78032
10	P4 - FORCE	0.9824	P7 - STRESS	0.29075	0.54153
11	P1 - WIDTH	0.75344	P7 - STRESS	0.13116	0.38052
12	Minor Input Parameters				
13	Input Parameter	Best Relationship With Filtering Output Parameter			
14		Relevance	Output Parameter	R2 Contribution	Correlation Value
15	P5 - YOUNG	0.40004	P9 - BUCKLING	0.032919	0.20206

For this particular Determination Histogram chart, in the **Properties** pane under **Axes**, **Y Axis** is set to **P8–DISPLACMENT**.

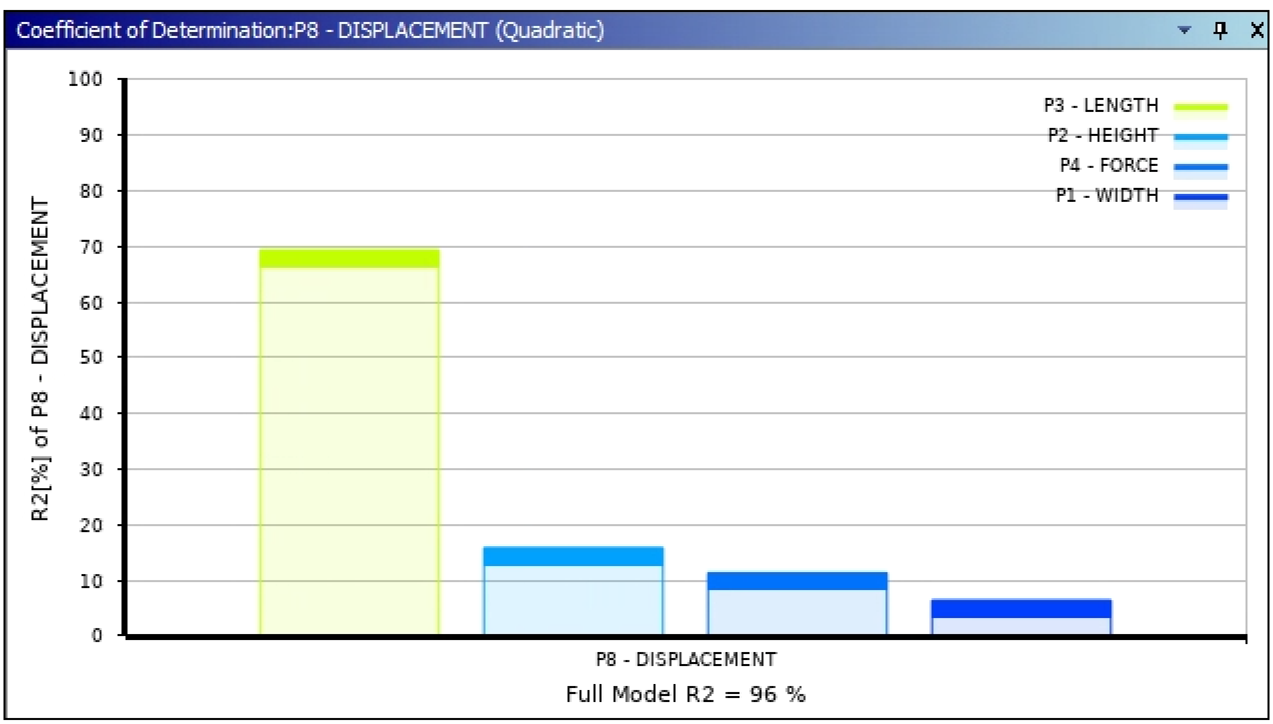


In the Determination Histogram chart, you can see that input parameters **P3–LENGTH**, **P2–HEIGHT**, and **P4 – FORCE** all affect output parameter **P8–DISPLACEMENT**. You can also see that of these three input parameters, **P3–LENGTH** has the greatest impact by far.

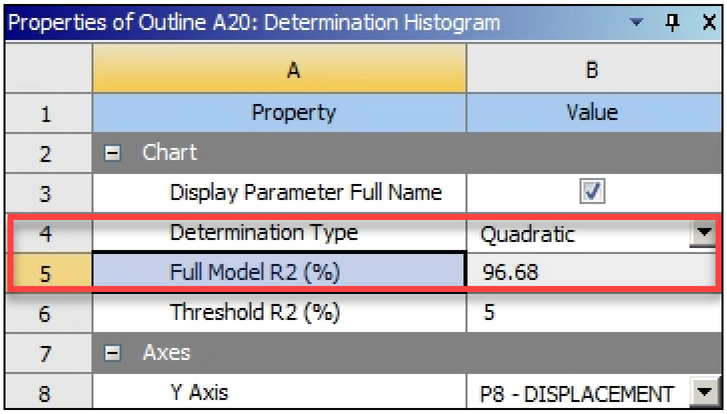
In either the **Chart** pane or **Properties** pane, you can check the **Full Model R2 (%)** value to see how well output variations are explained by input variations. The closer this value is to 100, the more certain it is that output variations result from the input parameters. The lower the value, the more likely that other factors such as noise, mesh error, or an insufficient number of points might be causing the output variations.

- In the **Chart** pane, the value for linear determination is **95%**.
- In the **Properties** pane, the value for linear determination is more precise at **95.31%**.

To view the Determination Histogram chart for the quadratic determination, in the **Properties** pane, set **Determination Type** to **Quadratic**. In the following figure, you can see that with a quadratic determination type, input parameter **P3– LENGTH** still has the highest impact on **P8–DISPLACEMENT**. The **Full Model R2 (%)** value for the quadratic determination is **96%**.



In the **Properties** pane, the more precise **Full Model R2 (%)** value for the quadratic determination is **96.68%**.



Properties of Outline A20: Determination Histogram

	A	B
1	Property	Value
2	⊟ Chart	
3	Display Parameter Full Name	☑
4	Determination Type	Quadratic
5	Full Model R2 (%)	96.68
6	Threshold R2 (%)	5
7	⊟ Axes	
8	Y Axis	P8 - DISPLACEMENT

Note:

In some cases, the relationship between parameters can be more complex and so cannot be explained completely with a linear or quadratic correlation. If you pursue the study with a response surface or goal-driven optimization, it will be difficult to build a Full 2nd Order Polynomials response surface. In this case, try using another response surface type, such as Kriging or Non-Parametric Regression.

You can filter your input parameters to keep only the most important by enabling or disabling their check boxes in the **Outline** pane. As indicated earlier, when you do this, a dialog box will open, indicating that parameter changes will clear all generated data in the system. If you click **Yes** and update the **Parameters Correlation** cell, all charts are updated accordingly.

When you are finished looking at charts, close the **Parameters Correlation** cell. Then, exit Workbench, saving project changes.

Using a Design of Experiments

This tutorial introduces a Design of Experiments (DOE) and shows how to use the **Design of Experiments** cell in a DesignXplorer system to investigate the impact of input parameters on the mass, equivalent stress, and total deformation of a geometry model:

What is a Design of Experiments?

DOE Types

Getting Started

Working with the DOE

Working with the Response Surface

Solving for the Desired Design Point

What is a Design of Experiments?

Overview

A Design of Experiments (DOE) is a technique used to determine the location of sampling points in such a way that the space of random input parameters is most efficiently explored and required information is obtained with a minimum number of sampling points. A **Design of Experiments** cell is available in these DesignXplorer systems: **Response Surface**, **Response Surface Optimization**, **ROM Builder**, and **Six Sigma Analysis**.

Note:

A **ROM Builder** system is used to produce a ROM. For more information, see "Using ROMs" in the *DesignXplorer User's Guide*.

Benefits

A DOE allows you to:

- Define input parameter limits.
- Select the DOE type best suited to your project and purposes.
- Specify DOE properties.
- Preview the DOE.
- Generate the DOE and view associated charts.
- Generate the response surface and view associated charts.

- Solve for the desired design point and view results.

Design Points and the DOE Matrix

Once you've defined input parameters, you can either preview or update the DOE to generate the *DOE Matrix*. The DOE Matrix is constructed of design points that can be submitted to the parent analysis system for solution.

- When you preview the DOE, the DOE Matrix is generated for viewing, but the design points are not solved. No output parameter values are available.
- When you update the DOE, the DOE Matrix is generated, and the design points are solved. Output parameter values are available.

Once the design points in the DOE Matrix are solved, you can update the **Response Surface** cell of the DesignXplorer system. Based on the design point information contained in the DOE Matrix, response surfaces are generated for each of the output parameters.

DOE Types

When setting up your DOE, you can select any of the DOE types available in DesignXplorer:

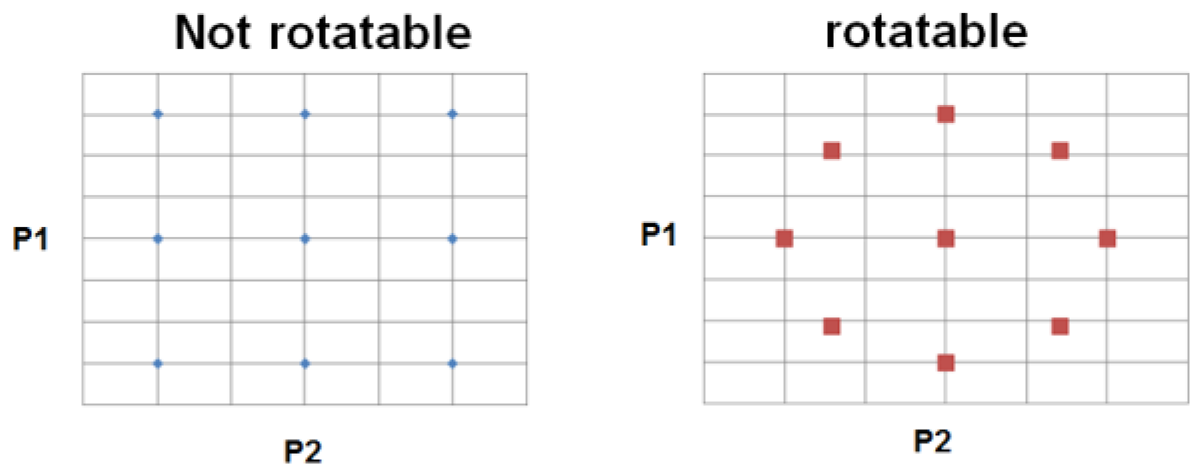
- Central Composite Design
- Optimal Space-Filling Design
- Box-Behnken Design
- Custom
- Custom + Sampling
- Sparse Grid Initialization
- Latin Hypercube Sampling Design

DOE Type Terminology

To understand the differences between DOE types, you need to know the following terms:

Rotatable

The degree to which the experimental design matrix is biased in any direction. A rotatable design can be rotated around its center point without changing the prediction variance of a given point. The predicted response is dependent only on its distance from the center point of the design.



Orthogonality

The degree to which the main effect and interaction estimates of interest are dependent on each other. For example, two vectors are orthogonal if the sum of the products of their corresponding elements is zero. A design is orthogonal if the effects of any variable have a sum of zero across the effects of the other variables. With an non-orthogonal design, some variables are interdependent. Because you cannot estimate the interactions between those variables, you cannot remove the unwanted effects that may result from them. With an orthogonal design, however, you can estimate the impact of each variable independently of the impacts of the other variables, which means that you can isolate any unwanted effects and remove them from your design.

	P1	P2
Run 1	1	1
Run 2	1	1
Run 3	-1	-1
Run 4	-1	-1

Not orthogonal because you can only estimate the combined effect of P1 and P2

	P1	P2
Run 1	1	1
Run 2	1	-1
Run 3	-1	1
Run 4	-1	-1

Orthogonal because you can estimate the independent effect of P1 and P2

Leverage

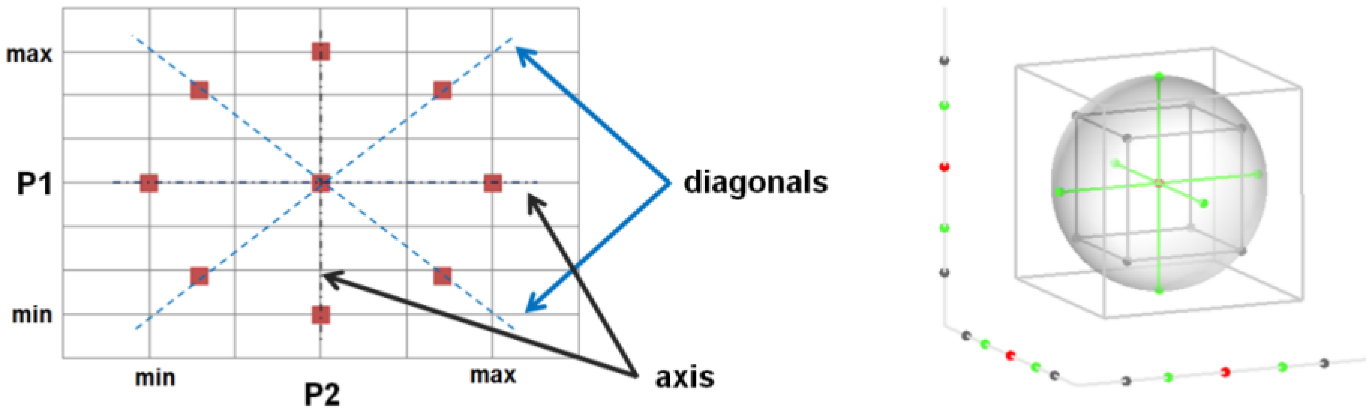
The opportunity of sample points to have an abnormal influence on the outcome.

Central Composite Design (CCD)

A Central Composite Design (CCD) is a five-level fractional factorial design that is suitable for calibrating the quadratic response model. A CCD consists of:

- 1 center point

- 2*N axis points located at the –a and +a positions on each axis of the selected input parameter
- 2^(N-f) factorial points located at the -1 and +1 positions along the diagonals of the input parameter space



There are five design types for CCD that you can use to improve the response surface fit for the DOE:

Face-Centered

Includes three levels and is not rotatable.

Rotatable

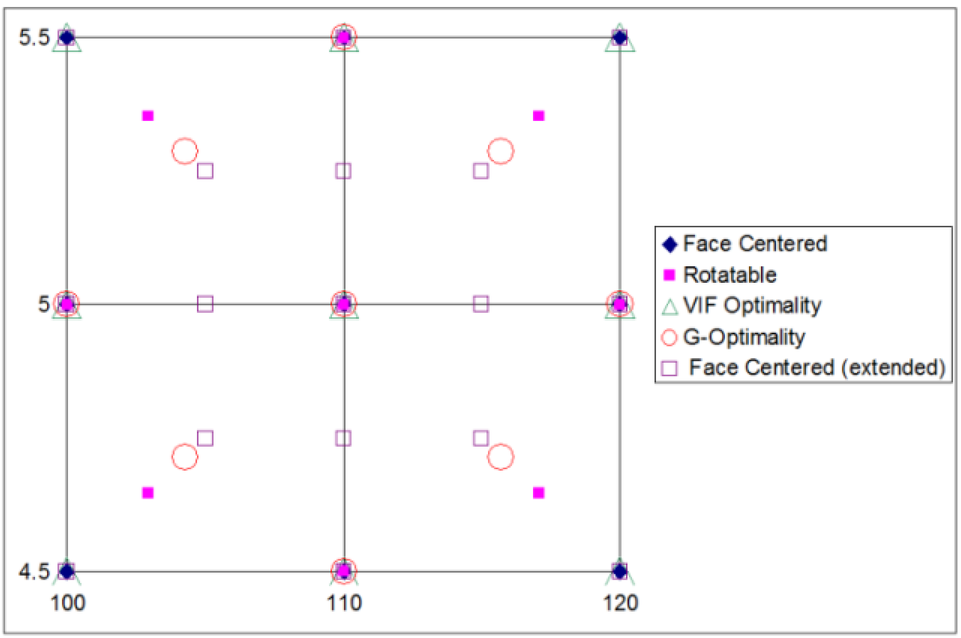
Includes five levels and is rotatable. The drawback to this CCD type is that it does not get sampling points at all extremes. Rotatable is preferred, however, because prediction variance is the same for any two locations that are the same distance from the design center.

Variance Inflation Factor (VIF) Optimality

Maximizes orthogonalilty.

G-Optimality

Minimizes leverage.



Auto-Defined

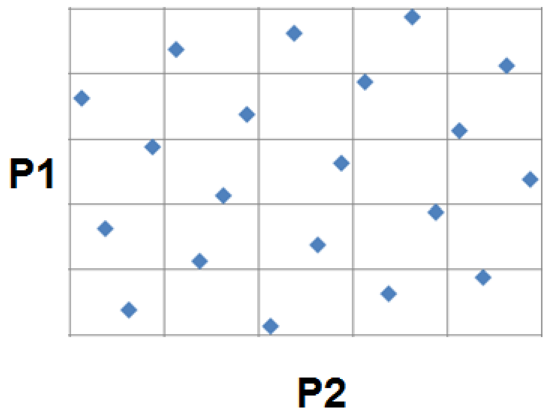
This option is the default. It automatically switches between **G-Optimality** for five input parameters or **VIF Optimality** for any other number of input parameters.

Optimal Space-Filling Design (OSF)

To offset the noise associated with physical experimentation, classical DOE types such as CCD focus on parameter settings near the perimeter of the design region. Because computer simulation is not quite as subject to noise, though, an Optimal Space-Filling (OSF) design is able to distribute the design parameters equally throughout the design space with the objective of gaining the maximum insight into the design with the fewest number of points.

One advantage of an OSF design is that you can specify the number of points to be generated. This is especially useful when limited computation time is available. Another advantage is its ability to fill the design space, which makes it appropriate when a more complex meta-modeling technique such as Kriging, Non-Parametric Regression, or Neural Network is used.

Possible disadvantages of an OSF design are that extremes (such as the corners of the design space) are not necessarily covered, that the selection of too few design points can result in a low quality of response prediction, and that some randomness is included in the choice of a starting point.

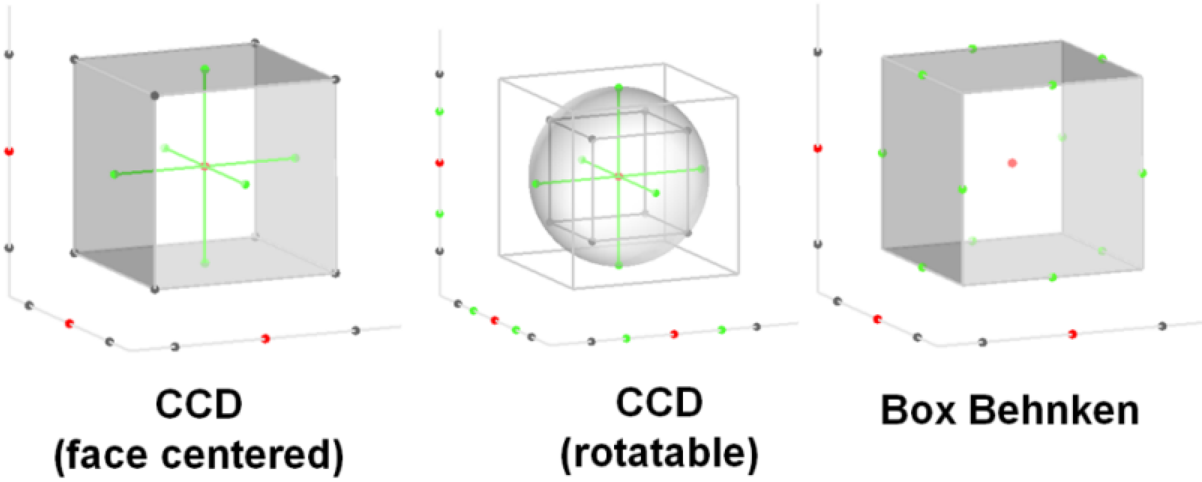


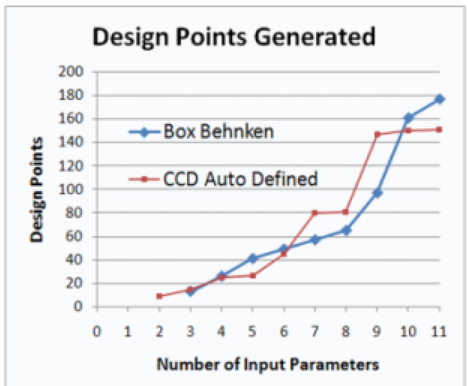
Box-Behnken Design

A Box-Behnken design is a three-level quadratic design that does not contain fractional factorial design. The sample combinations are treated in such a way that they are located at midpoints of edges formed by any two factors. The design is rotatable (or in cases, nearly rotatable).

One advantage of a Box-Behnken design is that it requires fewer design points than a full factorial CCD and generally requires fewer design points than a fractional factorial CCD. Additionally, a Box-Behnken design avoids extremes (such as the corners of the design space), allowing you to work around extreme factor combinations.

Possible disadvantages of a Box-Behnken design are that prediction at the corners of the design space is poor and that there are only three levels per parameter.

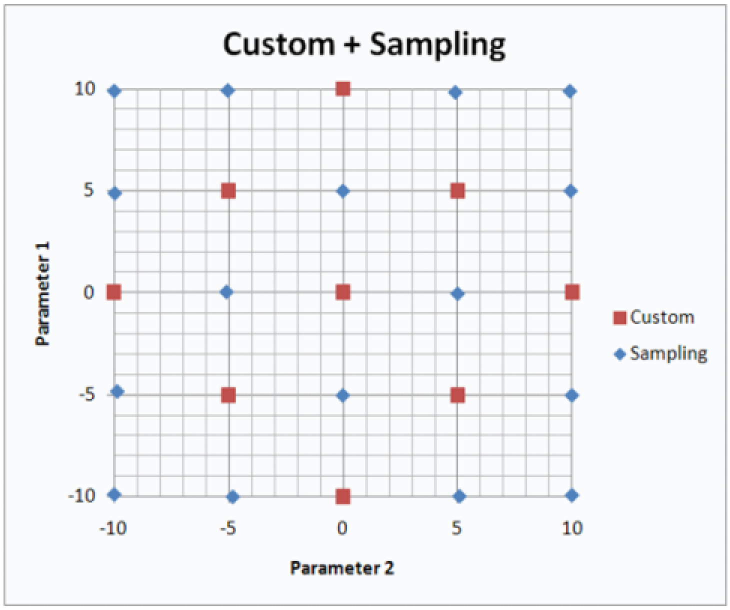




Custom and Custom + Sampling

A Custom design and a Custom + Sampling design allow you to create your own DOE types, offering a variety of ways to customize your DOE.

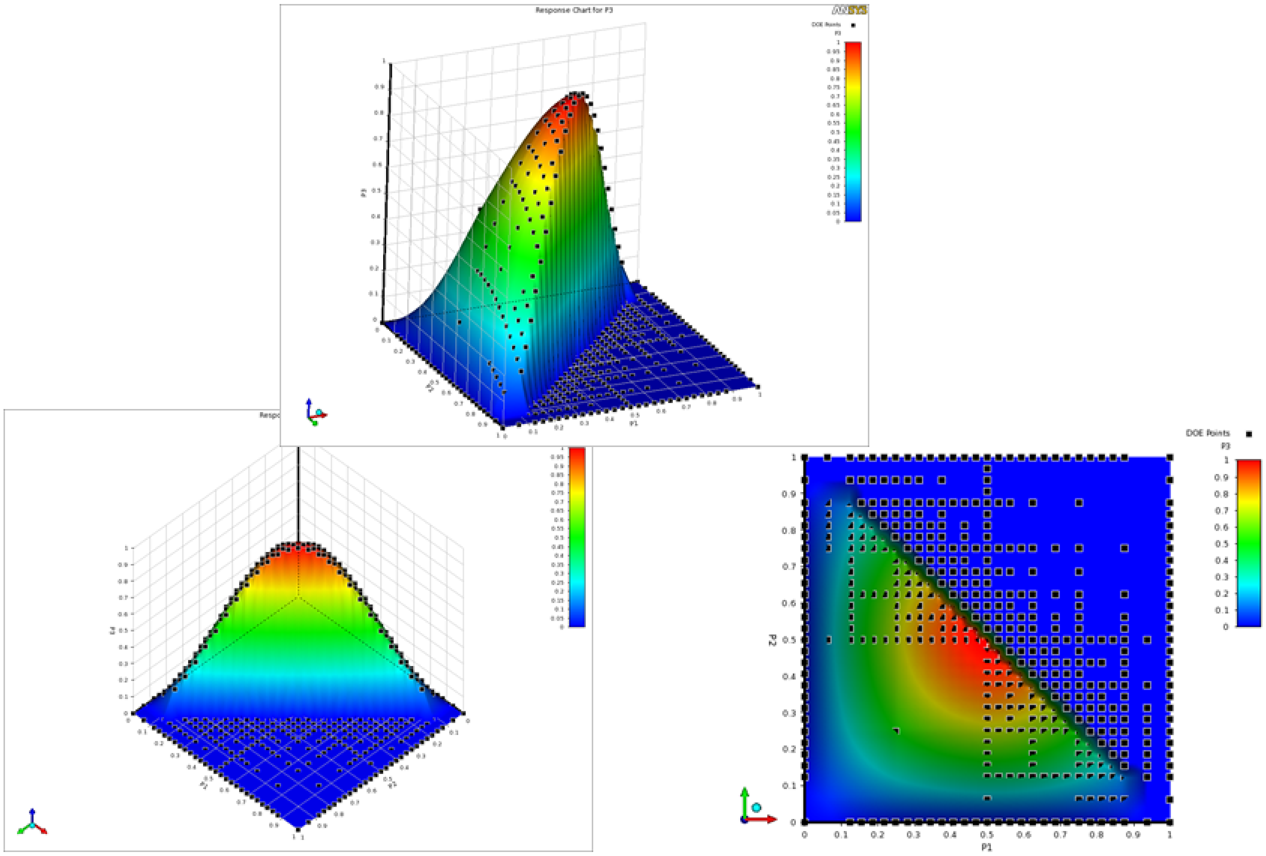
- A Custom design enables you to create a table of input parameters instead of using the default DOE, import design point files from an external CSV (Comma-Separated Values) file, and define design points to enrich an existing DOE.
- A Custom + Sampling design offers the same options as a Custom design. In addition, it allows you to add sampling to automatically fill up the design space more efficiently.



Sparse Grid Initialization

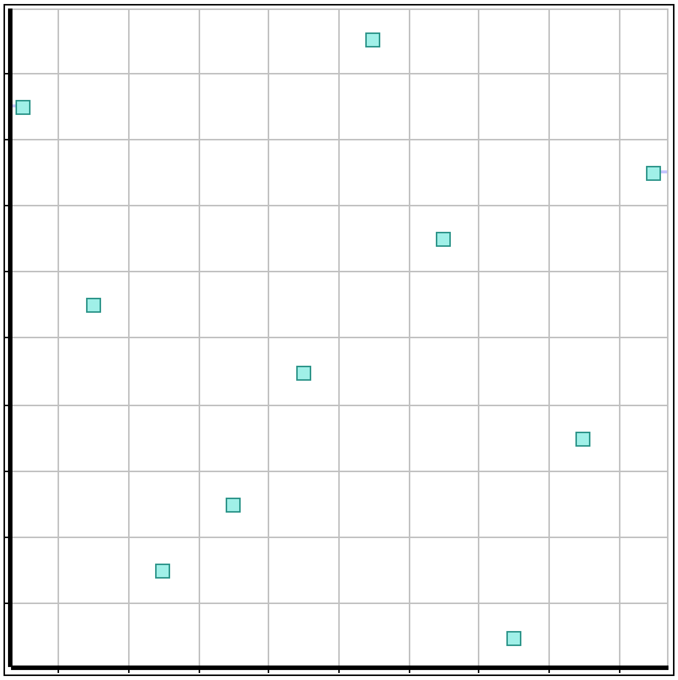
A Sparse Grid Initialization must be used when creating a Sparse Grid response surface. A Sparse Grid Initialization is an adaptive meta-model driven by the accuracy that you request. It increases the accuracy of the response surface by automatically refining the matrix of design points in locations where the relative error of the output parameter is higher. A Sparse Grid Initialization consists of one center point and 2*n axis points. A Sparse Grid Initialization allows as many linear basis functions as there are points of discretization.

One advantage of a Sparse Grid Initialization is that it refines only in the directions necessary, which means fewer design points are needed for the same quality response surface. Another is that Sparse Grid Initialization is effective at handling discontinuities.



Latin Hypercube Sampling Design

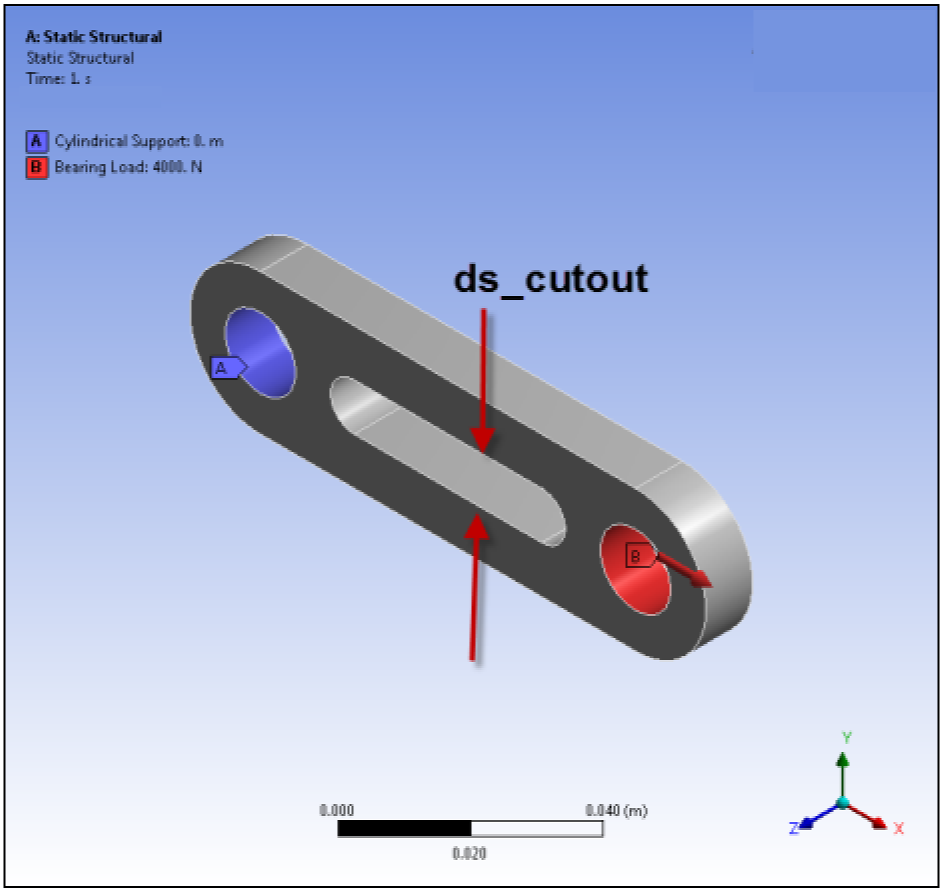
A Latin Hypercube Sampling Design uses an advanced form of the Monte Carlo sampling method to avoid clustering samples. In Latin Hypercube Sampling (LHS), points are randomly generated in a square grid across the design space, but no two points share the same value. This means that no point shares a row or a column of the grid with any other point.



One disadvantage of LHS is that extremes, such as the corners of the design space, are not necessarily covered. Additionally, the selection of too few design points can result in a lower quality of response prediction. When **Sample Types** is set to **CCD Samples** (default), a maximum of 20 input parameters is supported. This selection generates the same number of samples that a CCD DOE would generate for the same number of inputs, resulting in an LHS design that has the same cost as the corresponding CCD design.

Getting Started

For this tutorial, a model for a link with three cutouts is created in Ansys DesignModeler. Boundary conditions are applied.



Input Parameters

- ds_cutout
- Bearing Load X Component

Output Parameters

- Solid Mass
- Equivalent Stress Maximum
- Total Deformation Maximum
- Equivalent Stress 2 Maximum

Opening the Archived Workbench Project

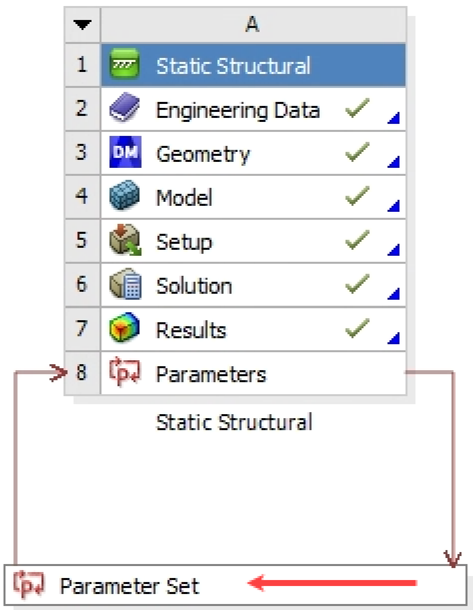
1. Start Workbench.
2. Select **File** → **Open**.
3. In the file browser, locate and open the archived Workbench project file `link1.wbpz`, which is in the directory to which you extracted the input files (p. 27).

   A dialog box appears for saving this archived file to a standard Workbench project file.

4. Save the project as `link1.wbpj` in either this same directory or another directory.

Verifying Input and Output Parameters

1. On the **Project Schematic**, double-click the **Parameter Set** bar to open it.



2. In the **Outline** pane, confirm that your input and output parameters are defined as shown:

Outline of All Parameters

	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	⊟ Input Parameters			
3	⊟ Static Structural (A1)			
4	P1	ds_cutout	5	
5	P3	Bearing Load X Component	4000	N
*	New input parameter	New name	New expression	
7	⊟ Output Parameters			
8	⊟ Static Structural (A1)			
9	P2	Solid Mass	0.17858	kg
10	P4	Equivalent Stress Maximum	109.44	MPa
11	P5	Total Deformation Maximum	0.013564	mm
12	P6	Equivalent Stress 2 Maximum	36.88	MPa
*	New output parameter		New expression	
14	Charts			

3. Close the **Parameter Set** bar.

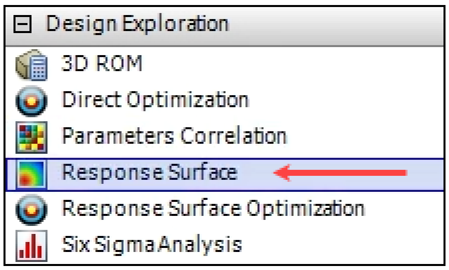
Subsequent topics provide step-by-step instructions for setting up and generating the DOE, generating a response surface and its associated charts, and viewing results.

Working with the DOE

You'll now add a **Response Surface** system, configure and update the **Design of Experiments** cell, and view associated charts.

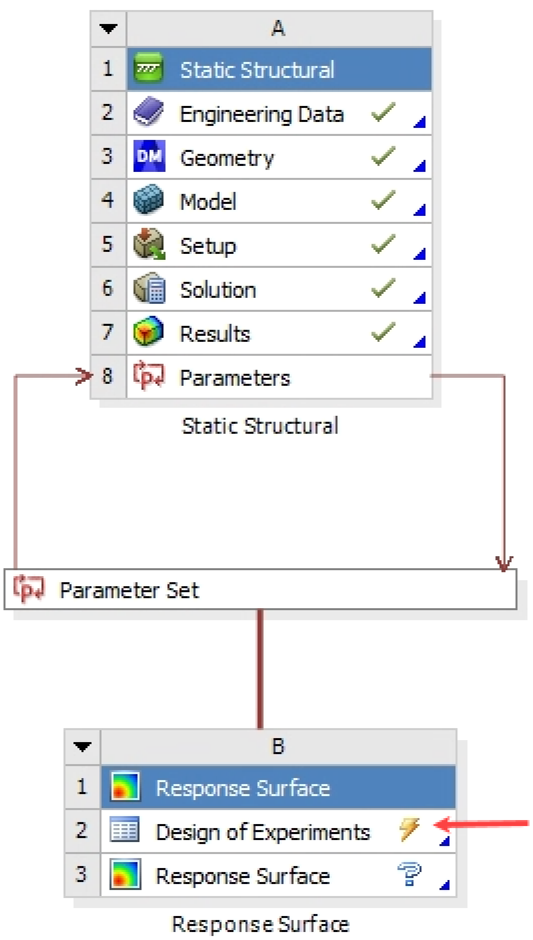
Add the Response Surface System

In the **Toolbox** under **Design Exploration**, double-click **Response Surface** to add this system to the **Project Schematic**.



Configure and Update the Design of Experiments Cell

1. In the **Response Surface** system, double-click the **Design of Experiments** cell to open it.



You can see your input and output parameters in the **Outline** pane:

Outline of Schematic B2: Design of Experiments

	A	B
1		Enabled
2	Design of Experiments	
3	Input Parameters	
4	Static Structural (A1)	
5	P1 - ds_cutout	☑
6	P3 - Bearing Load X Component	☑
7	Output Parameters	
8	Static Structural (A1)	
9	P2 - Solid Mass	
10	P4 - Equivalent Stress Maximum	
11	P5 - Total Deformation Maximum	
12	P6 - Equivalent Stress 2 Maximum	
13	Charts	

2. In the **Outline** pane, select the input parameter **P1 - ds_cutout**.

   In the **Properties** pane, confirm that the upper and lower bounds are set as shown:

Properties of Outline A5: P1 - ds_cutout

	A	B
1	Property	Value
2	General	
3	Units	
4	Type	Design Variable
5	Classification	Continuous
6	Values	
7	Lower Bound	4.5
8	Upper Bound	5.5
9	Allowed Values	Any

3. In the **Outline** pane, select the input parameter **P3 - Bearing Load X Component**.

   In the **Properties** pane, confirm that the upper and lower bounds are set as shown:

Properties of Outline A6: P3 - Bearing Load X Component

	A	B
1	Property	Value
2	⊟ General	
3	Units	N
4	Type	Design Variable
5	Classification	Continuous
6	⊟ Values	
7	Lower Bound	3600
8	Upper Bound	4400
9	Allowed Values	Any

4. In the **Outline** pane, select **Design of Experiments**.

5. In the **Properties** pane, confirm that **Design of Experiments Type** is set to **Central Composite Design**.

Properties of Outline A2: Design of Experiments

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Design of Experiments	
7	Design of Experiments Type	Central Composite Design
8	Design Type	Auto Defined
9	⊟ Design Point Report	
10	Report Image	None

6. In the toolbar, click **Preview** to see what design points are generated as part of the DOE. The **Table** pane displays nine design points, none of which are solved.

Table of Outline A2: Design Points of Design of Experiments

	A	B	C	D	E	F	G
1	Name	P1 - ds_cutout	P3 - Bearing Load X Component (N)	P2 - Solid Mass (kg)	P4 - Equivalent Stress Maximum (MPa)	P5 - Total Deformation Maximum (mm)	P6 - Equivalent Stress 2 Maximum (MPa)
2	1 DP 0	5	4000				
3	2	4.5	4000				
4	3	5.5	4000				
5	4	5	3600				
6	5	5	4400				
7	6	4.5	3600				
8	7	5.5	3600				
9	8	4.5	4400				
10	9	5.5	4400				

7. Update the **Design of Experiments** cell.

   When the update is complete, the **Table** pane displays results for these nine design points.

Table of Outline A2: Design Points of Design of Experiments

	A	B	C	D	E	F	G
1	Name	P1 - ds_cutout	P3 - Bearing Load X Component (N)	P2 - Solid Mass (kg)	P4 - Equivalent Stress Maximum (MPa)	P5 - Total Deformation Maximum (mm)	P6 - Equivalent Stress 2 Maximum (MPa)
2	1 DP 0	5	4000	0.17858	109.44	0.013564	36.88
3	2	4.5	4000	0.18289	112.98	0.01336	35.01
4	3	5.5	4000	0.17414	113.52	0.013953	37.569
5	4	5	3600	0.17858	98.492	0.012208	33.192
6	5	5	4400	0.17858	120.38	0.014921	40.568
7	6	4.5	3600	0.18289	101.68	0.012024	31.509
8	7	5.5	3600	0.17414	102.17	0.012557	33.812
9	8	4.5	4400	0.18289	124.28	0.014696	38.511
10	9	5.5	4400	0.17414	124.87	0.015348	41.326

View DOE Charts

The update also generates default charts, which you can see in the **Outline** pane under **Charts**:

Outline of Schematic B2: Design of Experiments

	A	B
1		Enabled
2	Design of Experiments	
3	Input Parameters	
4	Static Structural (A1)	
5	P1 - ds_cutout	☑
6	P3 - Bearing Load X Component	☑
7	Output Parameters	
8	Static Structural (A1)	
9	P2 - Solid Mass	
10	P4 - Equivalent Stress Maximum	
11	P5 - Total Deformation Maximum	
12	P6 - Equivalent Stress 2 Maximum	
13	Charts	
14	Parameters Parallel	
15	Design Points vs Parameter	

When you select a chart in the **Outline** pane, the **Table** pane displays the chart data. The **Chart** pane displays the chart itself.

This topic describes only the Design Points vs Parameter chart.

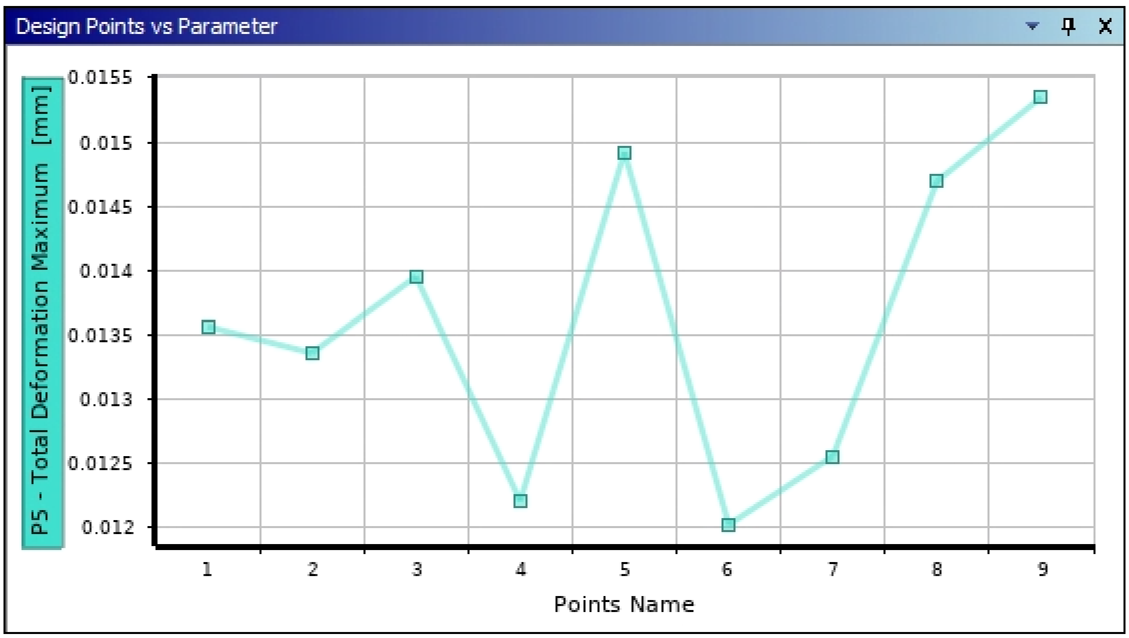
1. In the **Outline** pane under **Charts**, select **Design Points vs Parameter**.

2. In the **Properties** pane under **Axes**, set properties as shown:

Properties of Outline A15: Design Points vs Parameter

	A	B
1	Property	Value
2	⊟ Chart	
3	Display Parameter Full Name	☑
4	⊟ Axes	
5	X-Axis (Bottom)	Design Points
6	X-Axis (Top)	
7	Y-Axis (Left)	P5 - Total Deformation Maximum
8	Y-Axis (Right)	

The **Chart** pane displays the Design Points vs Parameters chart. The nine solved design points are shown across the bottom of the chart. The output parameter values for **Total Deformation Maximum** are shown on the left side of the chart.



3. If you want, in the **Properties** pane under **Axes**, change the selections to p additional charts.

4. When finished, close the **Design of Experiments** cell.

Working with the Response Surface

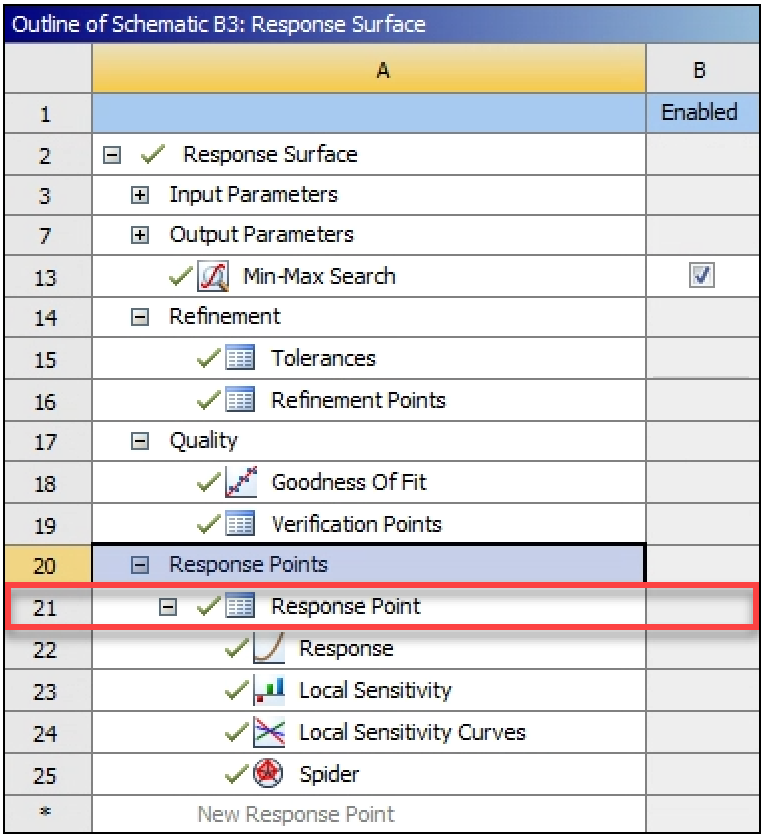
You'll now configure and update the **Response Surface** cell and view associated charts.

Configure and Update the Response Surface Cell

1. In the **Response Surface** system, double-click the **Response Surface** cell to open it.

2. Update the **Response Surface** cell.

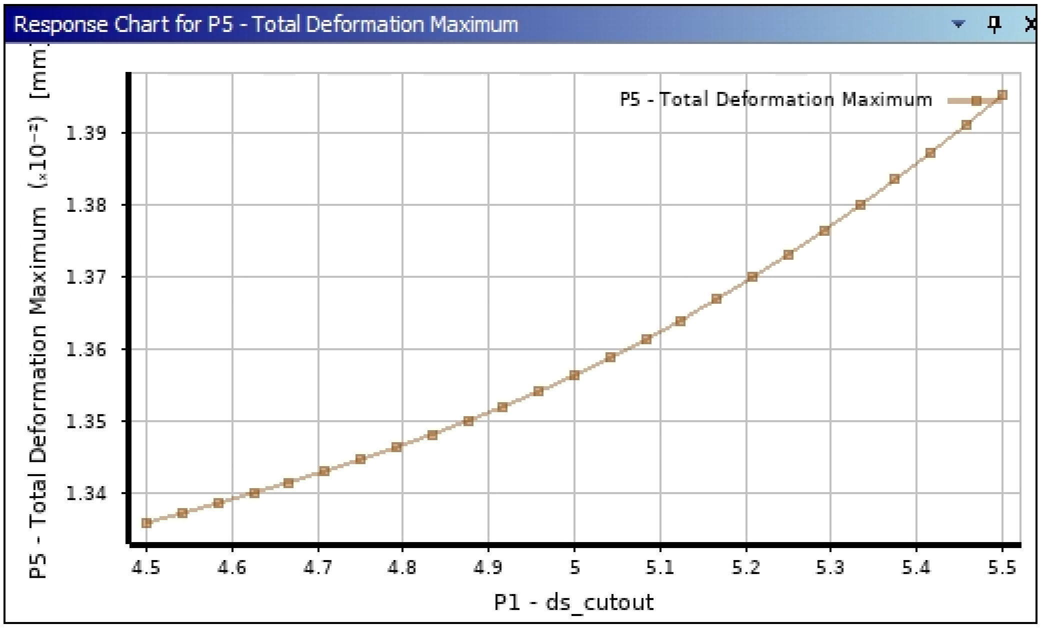
3. In the **Outline** pane under **Response Points**, you can see that a single response point is generated, along with charts for this response point.



When you select a chart for the response point, the **Table** pane displays the chart data. The **Chart** pane displays the chart itself.

View Response Charts

When **Response** is the chart selected, the **Graph** pane displays the Response chart. A Response chart allows you to see how changes to an input parameter affect a selected output parameter.

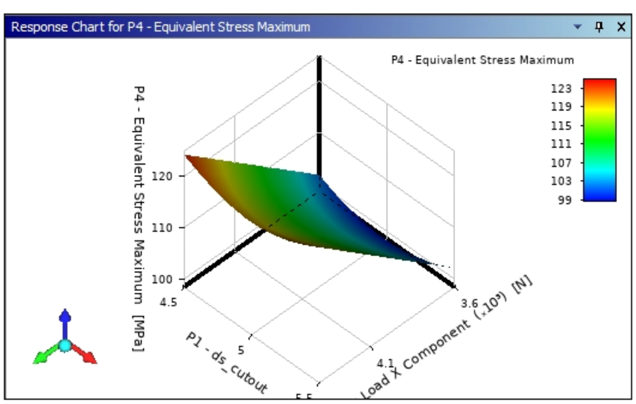


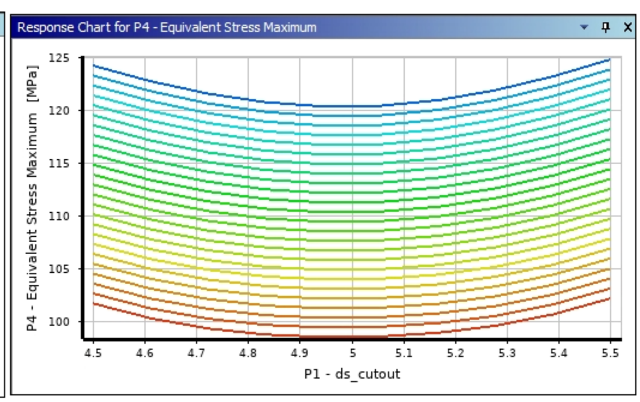
In the **Properties** pane for a Response chart, **Mode** provides three viewing options: **2D**, **3D**, and **2D Slices**. For this particular Response chart, the properties are set as shown:

Properties of Outline A22: Response

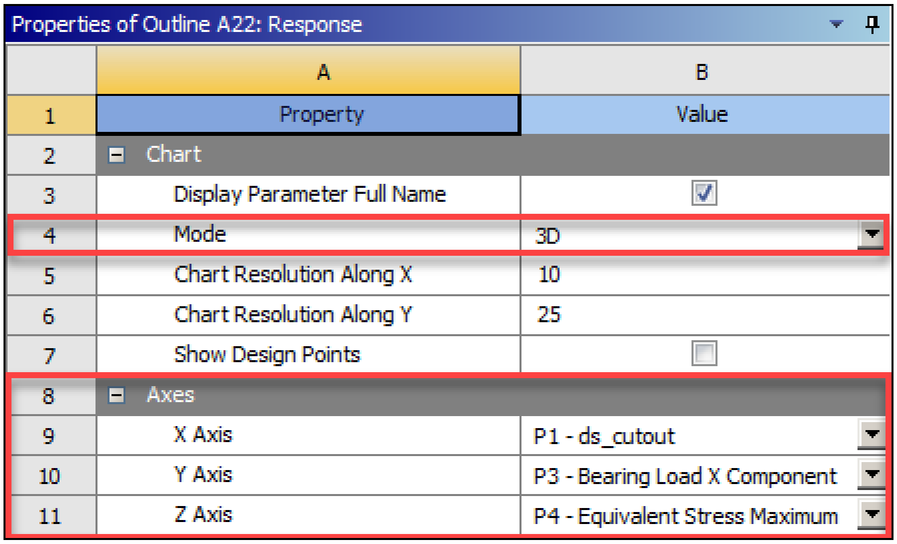
	A	B
1	Property	Value
2	⊟ Chart	
3	Display Parameter Full Name	☑
4	Mode	2D
5	Chart Resolution Along X	25
6	Show Design Points	☐
7	⊟ Axes	
8	X Axis	P1 - ds_cutout
9	Y Axis	P5 - Total Deformation Maximum
10	⊟ Input Parameters	

The following images show a 3D Response chart and a 2D Slices Response chart.





To generate the 3D Response chart, the properties are set as shown:



Properties of Outline A22: Response

	A	B
1	Property	Value
2	⊟ Chart	
3	Display Parameter Full Name	☑
4	Mode	3D
5	Chart Resolution Along X	10
6	Chart Resolution Along Y	25
7	Show Design Points	☐
8	⊟ Axes	
9	X Axis	P1 - ds_cutout
10	Y Axis	P3 - Bearing Load X Component
11	Z Axis	P4 - Equivalent Stress Maximum

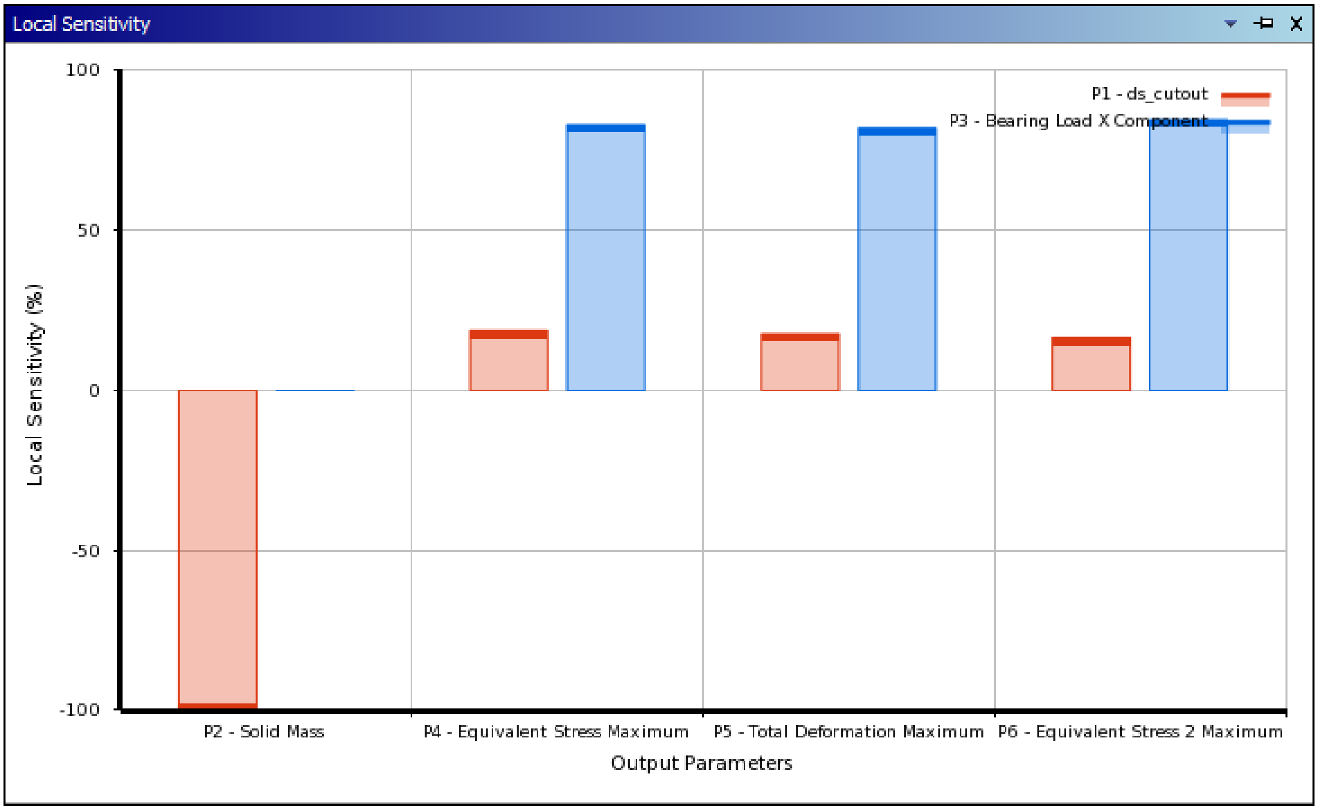
To generate the 2D Slice Response chart, **Mode** is changed to **2D Slices**. All other properties remain unchanged.

View Additional Response Point Charts

To further explore your design space, you can view these additional charts for a response point: Local Sensitivity, Local Sensitivity Curves, and Spider.

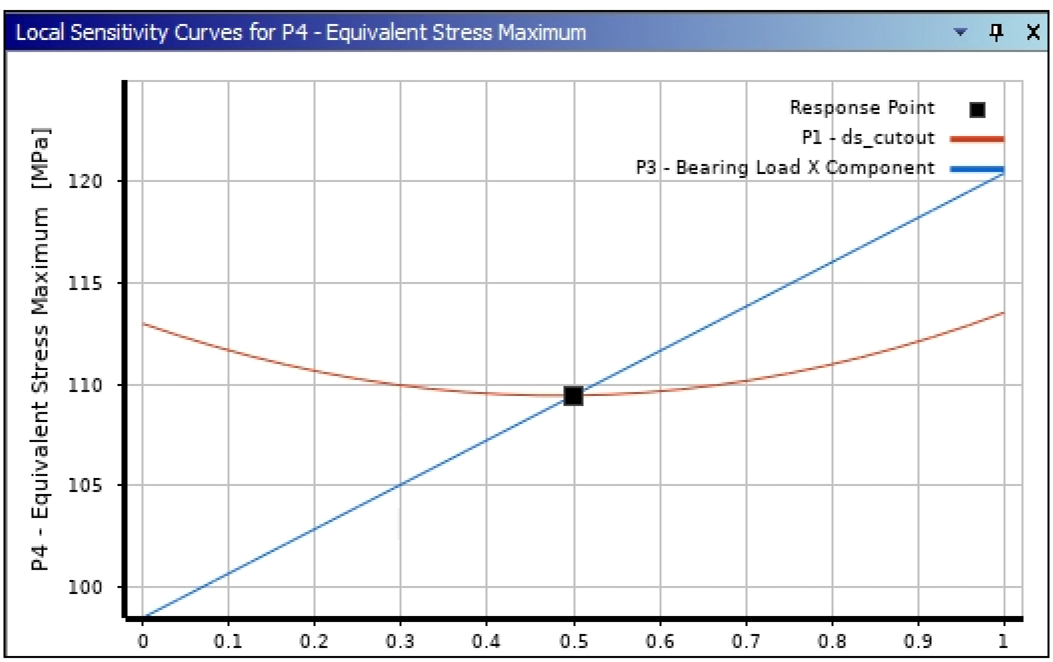
Local Sensitivity Chart

The Local Sensitivity chart allows you to see the impact of continuous input parameters (both with and without manufacturable values) on output parameters. This chart calculates the change of the output based on the change of each input independently, allowing you to see the weight of each input.



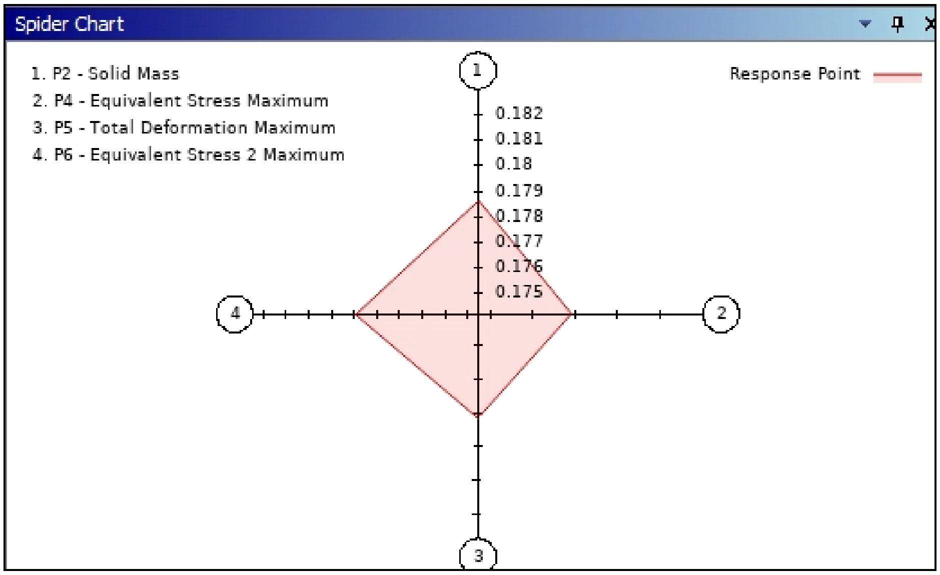
Local Sensitivity Curves Chart

The Local Sensitivity Curves chart helps you to focus your analysis by allowing you to view independent parameter variations within the standard Local Sensitivity chart. This multi-curve chart provides a means of viewing the impact of each input parameter on a specific output parameter.



Spider Chart

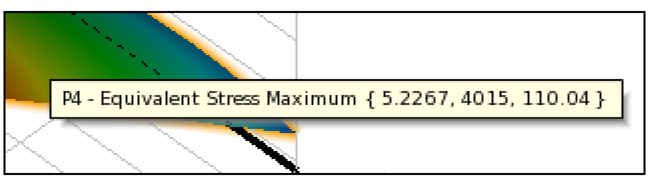
The Spider chart allows you to visualize the impact that changing the input parameter or parameters has on all of the output parameters simultaneously.



Solving for the Desired Design Point

In the **Response Surface** cell, you can now specify a design point, solve for it, and view the solution.

1. In the **Outline** pane under the single response point, select **Response** to view the Response chart. This topic assumes that you are viewing the 3D response chart shown in the previous topic. As you move the mouse cursor over the chart, the tooltip displays data for the current point.



2. Right-click a desired point on the Response chart and select **Explore Response Surface at Point**.

   This new response point is added to the **Table** pane as **Response Point 1**.

3. In the **Table** pane, right-click this new response point and select **Insert as Design Point**.

4. Close the **Response Surface** cell.

5. Double-click the **Parameters Set** bar to open it.

   The **Table** pane displays the newly added design point, **DP 1**. The input parameter values that you see will differ from those shown as they depend on the response point that you inserted as a design point.

Table of Design Points

	A	B	C	D	E
1	Name	P1 - ds_cutout	P3 - Bearing Load X Component	P2 - Solid Mass	P4 - Equivalent Stress Maximum
2	Units		N	kg	MPa
3	DP 0 (Current)	5	4000	0.17858	109.44
4	DP 1	5.2267	4015		
*					

6. Right-click **DP 1** and select **Copy inputs to Current**.

   The values for the new design point are copied to the current design point, which is designated by **(Current)**. The current design point is the one that will be solved and for which the solution will be displayed.

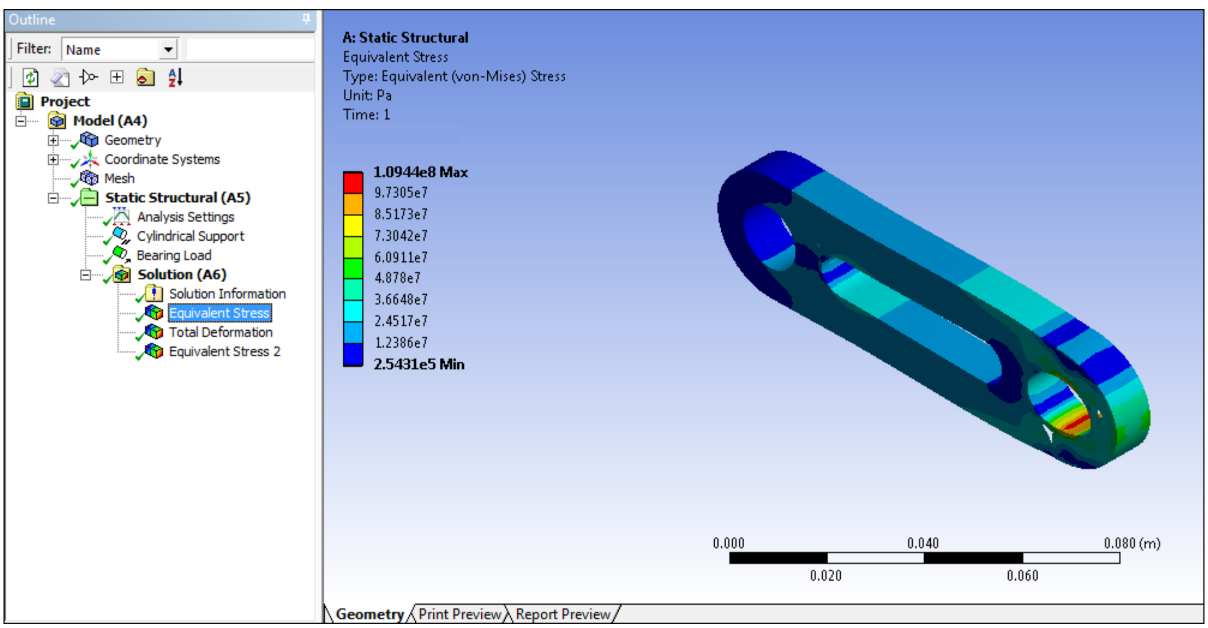
7. Close the **Parameter Set** bar.

8. Update the project.

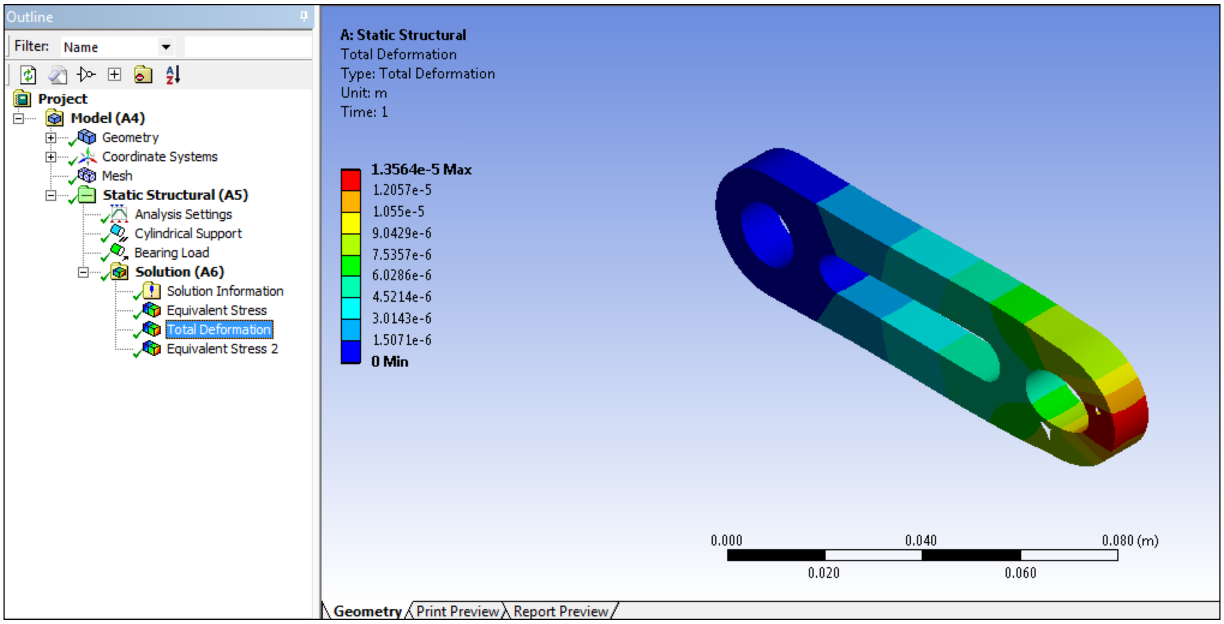
9. When the update is complete, in the **Static Structural** system, double-click the **Solution** cell to open it. Mechanical starts.

   Again, the results that you see will differ slightly from those shown.

10. In the **Outline** view under **Solution**, select **Equivalent Stress** to view this result.



11. In the **Outline** view under **Solution**, select **Total Deformation** to view this result.



12. Exit Mechanical.

13. Exit Workbench, saving project changes.

Performing a Goal-Driven Optimization

This tutorial provides an introduction to goal-driven optimizations in DesignXplorer. The end goal is to minimize the mass of a support while exceeding a safety factor, with safety as the most important consideration. This tutorial gives step-by-step instructions for importing an Ansys Mechanical input file, generating design points and a response surface, defining optimization goals and objectives, and viewing results for selected points.

This tutorial covers the following goal-driven optimization topics:

What is Goal-Driven Optimization?

Getting Started

Promoting Properties and Results in Mechanical to Parameters

Performing the Optimization

Viewing Goal-Driven Optimization Charts

Viewing Candidate Points

Viewing the Solution

Using ACT to Expose MATLAB Optimizers in DesignXplorer

What is Goal-Driven Optimization?

Overpane

Goal-driven optimization is a set of constrained, multi-objective optimization techniques in which the "best" possible designs are obtained from a sample set given goals that are set for parameters.

Benefits

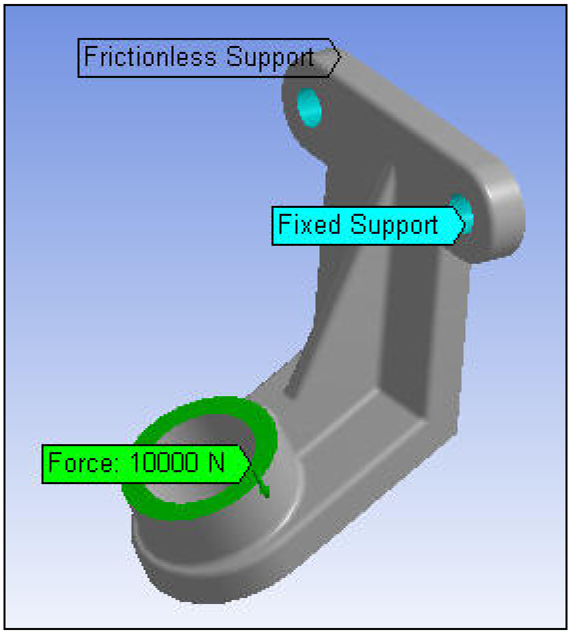
A goal-driven optimization allows you to determine the effect on input parameters with certain objectives applied on output parameters. To do this, you specify a series of design goals or objectives to use to generate an optimized design. You can define the optimization domain, specify values for input parameters, and weight goals in terms of their importance. Based on your specifications, DesignXplorer then generates a set of sample designs from which you can select the most promising candidate designs.

Response Surface Optimization

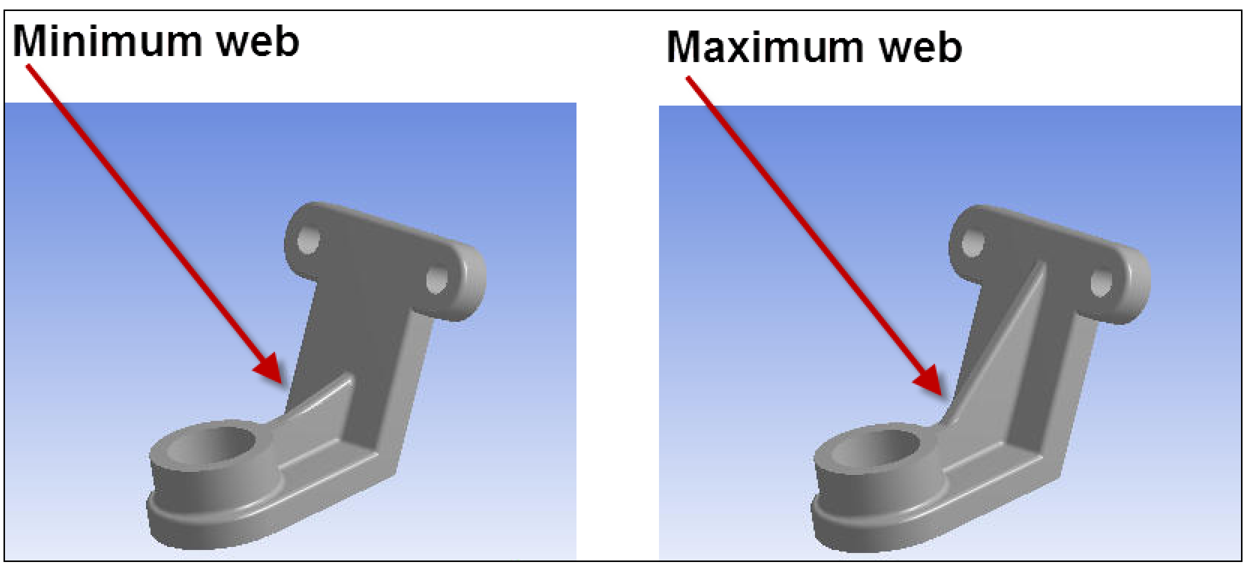
This tutorial uses a **Response Surface Optimization** system that obtains its information from its own **Response Surface** cell. Consequently, it is dependent on the quality of the response surface. The available optimization methods (Screening, MOGA, NLPQL, and MISQP) use response surface evaluations rather than real solves. Response surfaces are built from the DOE and quickly provide approximated values of output parameters throughout the design space.

Getting Started

The model for a support is created in DesignModeler. Constraints and loads are shown.



In this tutorial, you'll explore the *between mass*, which is the size of the web on the support, and the defined safety factor.



Open the Archived Workbench Project

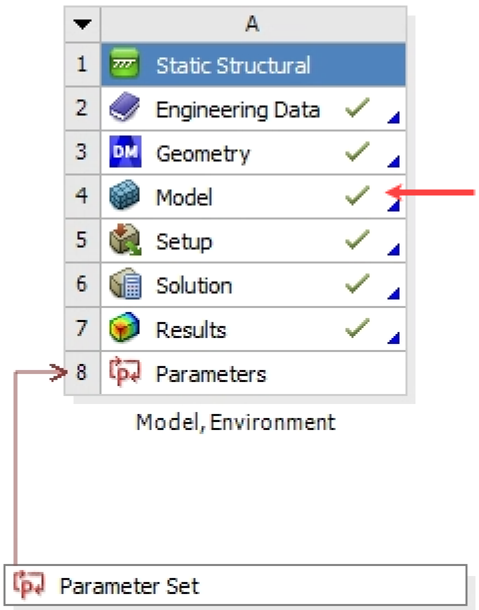
1. Start Workbench.
2. Select **File** → **Open**.
3. In the file browser, locate and open the archived Workbench project file `support.wbpz`, which is in the directory to which you extracted the input files (p. 27).

A dialog box appears for saving this archived file to a standard Workbench project file.

4. Save the project as `support.wbpj` in either this same directory or another directory.

Promoting Properties and Results in Mechanical to Parameters

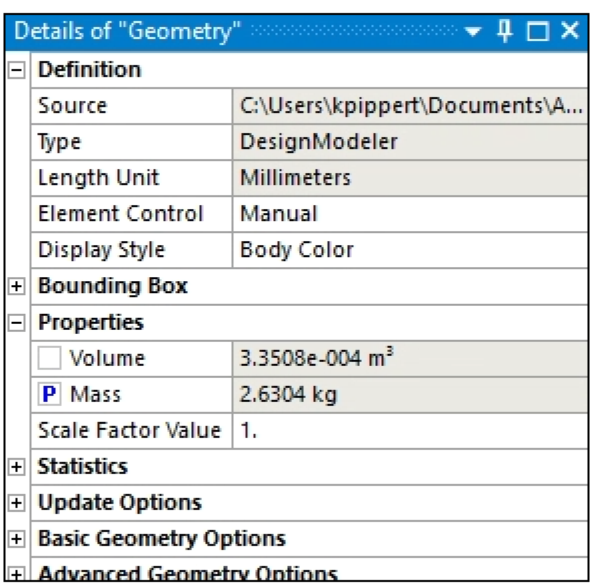
1. On the **Project Schematic**, double-click the **Model** cell.



Mechanical starts. You'll now promote dimensions in Mechanical to input parameters.

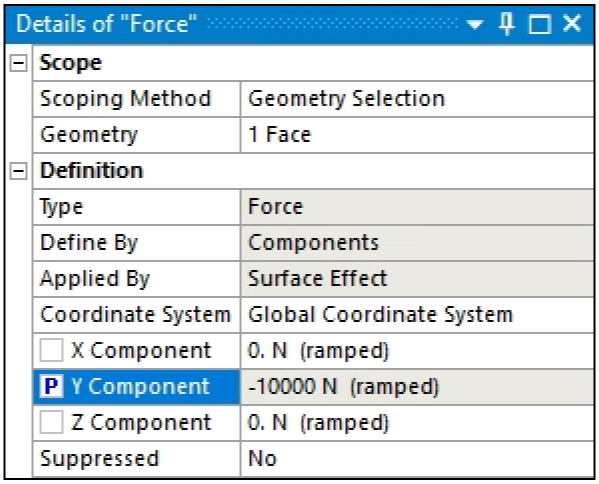
2. In **Outline** view, select **Geometry**.

3. In the **Details** view under **Properties**, select the check box for **Mass** to promote it to an output parameter.



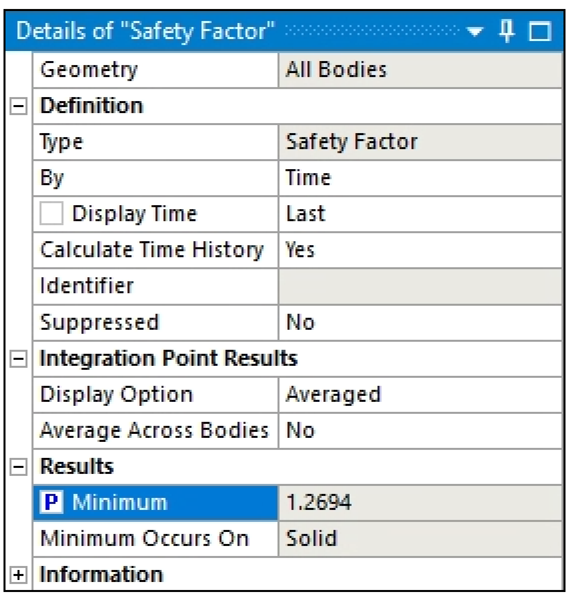
4. In **Outline** view under **Environment**, select **Force**.

5. In the **Details** view under **Definition**, select the check box for **Y Component** to promote it to an input parameter.



6. In the **Outline** view, under **Solution** > **Stress Tool**, select **Safety Factor**.

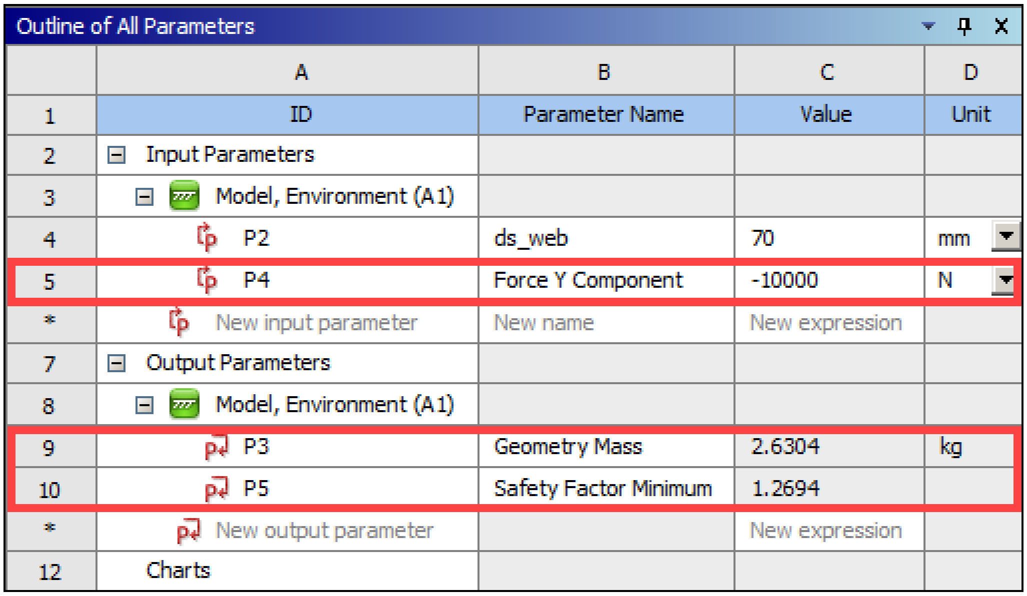
7. In the **Details** view under **Results**, select the check box for **Minimum** to promote it to an output parameter.



8. Exit Mechanical.

9. On the **Project Schematic**, double-click the **Parameter Set** bar to open it.

   In the **Outline** pane, you can see that the Mechanical values that you promoted to parameters display as input and output parameters.



Outline of All Parameters

	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	Input Parameters			
3	Model, Environment (A1)			
4	P2	ds_web	70	mm
5	P4	Force Y Component	-10000	N
*	New input parameter	New name	New expression	
7	Output Parameters			
8	Model, Environment (A1)			
9	P3	Geometry Mass	2.6304	kg
10	P5	Safety Factor Minimum	1.2694	
*	New output parameter		New expression	
12	Charts			

10. Close the **Parameter Set** bar.

Performing the Optimization

The following topics explain how to perform the optimization:

Adding the Goal-Driven Optimization System

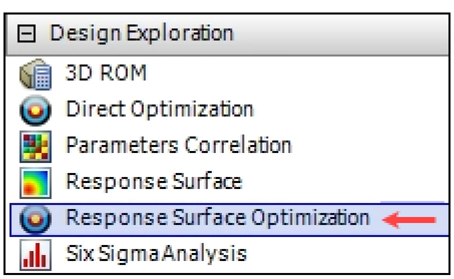
Configuring and Updating the Design of Experiments Cell

Configuring and Updating the Response Surface Cell

Configuring and Updating the Optimization Cell

Adding the Goal-Driven Optimization System

Your first step is to add the goal-driven optimization system. In the **Toolbox** under **Design Exploration**, double-click **Response Surface Optimization**.

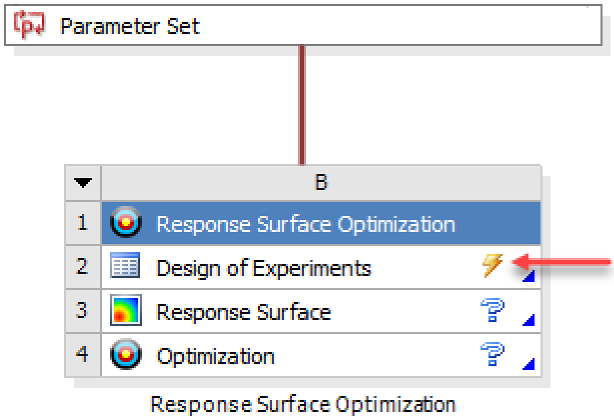


This system is added below the **Parameter Set** bar. You now need to work through each cell.

Configuring and Updating the Design of Experiments Cell

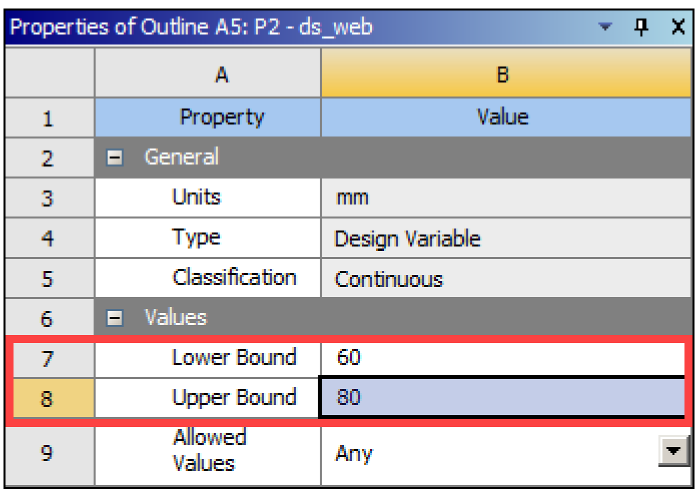
Your first step is to configure and update the **Design of Experiments** cell.

1. In the **Response Surface Optimization** system, double-click the **Design of Experiments** cell to open it.



2. In the **Outline** pane, select input parameter **P2 - ds_web**.

3. In the **Properties** pane, set the lower and upper bounds as shown:



Properties of Outline A5: P2 - ds_web

	A	B
1	Property	Value
2	General	
3	Units	mm
4	Type	Design Variable
5	Classification	Continuous
6	Values	
7	Lower Bound	60
8	Upper Bound	80
9	Allowed Values	Any

4. In the **Outline** pane, select input parameter **P4 - Force Y Component**.

5. In the **Properties** pane, confirm that the lower and upper bounds are set as shown:

Properties of Outline A6: P4 - Force Y Component

	A	B
1	Property	Value
2	⊟ General	
3	Units	N
4	Type	Design Variable
5	Classification	Continuous
6	⊟ Values	
7	Lower Bound	-11000
8	Upper Bound	-9000
9	Allowed Values	Any

6. In the **Outline** pane, select **Design of Experiments**.

7. In the **Properties** pane under **Design of Experiments**, set the DOE type and design type as shown:

Properties of Outline A2: Design of Experiments

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Design of Experiments	
7	Design of Experiments Type	Central Composite Design
8	Design Type	Face-Centered
9	Template Type	Standard
10	⊟ Design Point Report	
11	Report Image	None

8. Update the **Design of Experiments** cell. This will take a few minutes.

   When the update is complete, the **Table** pane displays a total of 9 design points, each of which is solved.

Table of Outline A2: Design Points of Design of Experiments

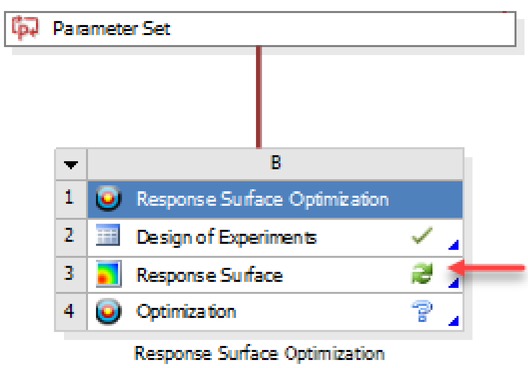
	A	B	C	D	E
1	Name	P2 - ds_web (mm)	P4 - Force Y Component (N)	P3 - Geometry Mass (kg)	P5 - Safety Factor Minimum
2	1 DP 0	70	-10000	2.6304	1.2694
3	2	60	-10000	2.6112	1.2963
4	3	80	-10000	2.6483	1.1038
5	4	70	-11000	2.6304	1.154
6	5	70	-9000	2.6304	1.4104
7	6	60	-11000	2.6112	1.1785
8	7	80	-11000	2.6483	1.0034
9	8	60	-9000	2.6112	1.4404
10	9	80	-9000	2.6483	1.2264

9. Close the **Design of Experiments** cell.

Configuring and Updating the Response Surface Cell

Your next step is to configure and update the **Response Surface** cell.

1. In the **Response Surface Optimization** system, double-click the **Response Surface** cell to open it.



2. In the **Outline** pane, select **Response Surface**.

3. In the **Properties** pane, ensure that **Response Surface Type** is set to the default value, **Genetic Aggregation**.

4. In the **Outline** pane, ensure that the check box for **Min-Max Search** is selected.

5. Update the **Response Surface** cell.

   The Min-Max Search performs a series of NLPQL optimizations to find the minimum and maximums.

6. In the **Outline** pane, select **Min-Max Search**.

   The **Table** pane displays the Min-Max Search results.

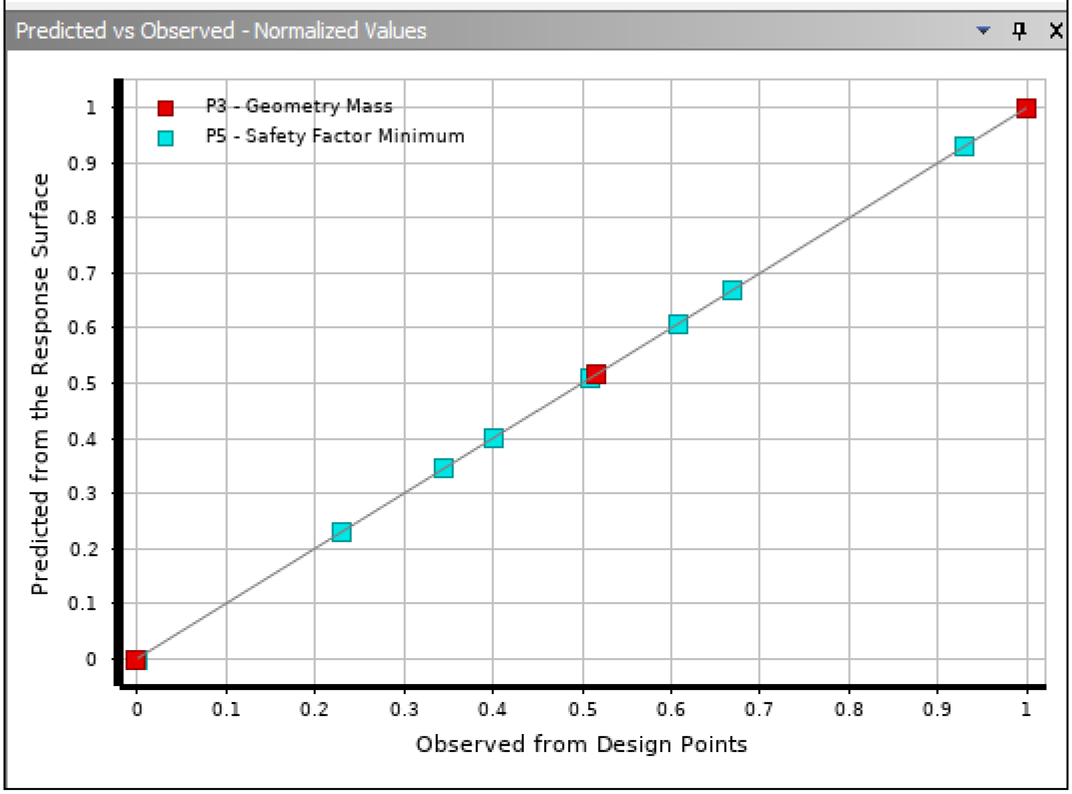
Table of Outline A11: Min-Max Search					
	A	B	C	D	
1	Name	P2 - ds_web (mm)	P4 - Force Y Component (N)	P3 - Geometry Mass (kg)	P5 - S
2	Output Parameter Minimums				
3	P3 - Geometry Mass	60	-10990	**2.6112**	1.179
4	P5 - Safety Factor Minimum	80	-11000	2.6483	**1.00**
5	Output Parameter Maximums				
6	P3 - Geometry Mass	80	-9443.1	**2.6483**	1.170
7	P5 - Safety Factor Minimum	62.254	-9000	2.6159	**1.44**

7. In the **Outline** pane under **Quality**, select **Goodness Of Fit**.

   In the **Table** pane, you can see goodness-of-fit metrics for all output parameters. In the **Graph** pane, you can see the Goodness of Fit chart. The response surface fits both output parameters fairly well.

Table of Schematic B3: Response Surface

	A	B	C
1		P3 - Geometry Mass	P5 - Safety Factor Minimum
2	Coefficient of Determination (Best Value = 1)		
3	Learning Points	1	0.99999
4	Cross-Validation on Learning Points	1	0.99989
5	Root Mean Square Error (Best Value = 0)		
6	Learning Points	0	0.00045684
7	Cross-Validation on Learning Points	0	0.0013686
8	Relative Maximum Absolute Error (Best Value = 0%)		
9	Learning Points	0	0.58379
10	Cross-Validation on Learning Points	0	1.4399
11	Relative Average Absolute Error (Best Value = 0%)		
12	Learning Points	0	0.2589
13	Cross-Validation on Learning Points	0	0.86443

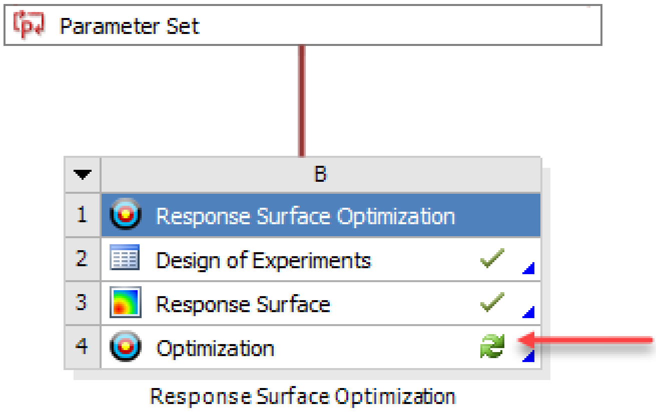


8. Close the **Response Surface** cell.

Configuring and Updating the Optimization Cell

Your next step is to configure and update the **Optimization** cell.

1. In the **Response Surface Optimization** system, double-click the **Optimization** cell to open it.



2. In the **Outline** pane, select **Optimization**.

3. In the **Properties** pane, set optimization properties as shown:

Properties of Outline : Optimization

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☑
4	Retain Data for Each Preserved Design Point	☑
5	⊟ Failed Design Points Management	
6	Number of Retries	0
7	⊟ Optimization	
8	Method Selection	Manual
9	Method Name	Screening
10	Estimated Number of Evaluations	2000
11	Verify Candidate Points	☐
12	Number of Samples	2000
13	Maximum Number of Candidates	3

4. In the **Outline** pane, select **Objectives and Constraints**.

5. In the **Table** pane, for parameters **P3 - Geometry Mass** and **P5 - Safety Factor Minimum**, specify objectives and constraints as shown:

Table of Schematic B4: Optimization

	A	B	C	D	E	F	G	
1	Name	Parameter	Objective			Constraint		
2			Type	Target	Tolerance	Type	Lower Bound	Uppe
3	Minimize P3	P3 - Geometry Mass	Minimize	2.6		No Constraint		
4	Seek P5 = 1.1	P5 - Safety Factor Minimum	Seek Target	1.1	0.1	No Constraint		
*		Select a Parameter						

6. Now, for the two parameters with objectives, ensure that the relative importance of the objectives are set as indicated:

    a. In the **Outline** pane under **Objectives and Constraints**, select **Minimize P3**. In the **Properties** pane, ensure that **Target** is set to **`2.6`**.

    b. In the **Properties** pane under **Decision Support Process**, ensure that **Objective Importance** is set to **Default**.

    c. In the **Outline** pane under **Objectives and Constraints**, select **Seek P5 = 1.1**. In the **Properties** pane, ensure that **Target** is set to **`1.1`** and **Tolerance** is set to **`0.1`**.

    d. In the **Properties** pane under **Decision Support Process**, set **Objective Importance** to **Higher**.

7. Update the **Optimization** cell.

Viewing Goal-Driven Optimization Charts

Once the update is complete, the following optimization charts are available: Candidate Points, Tradeoff, Samples, Sensitivities, and History.

- To view all but the History chart, in the **Outline** pane under **Results**, select the chart to display it in the **Graph** pane.

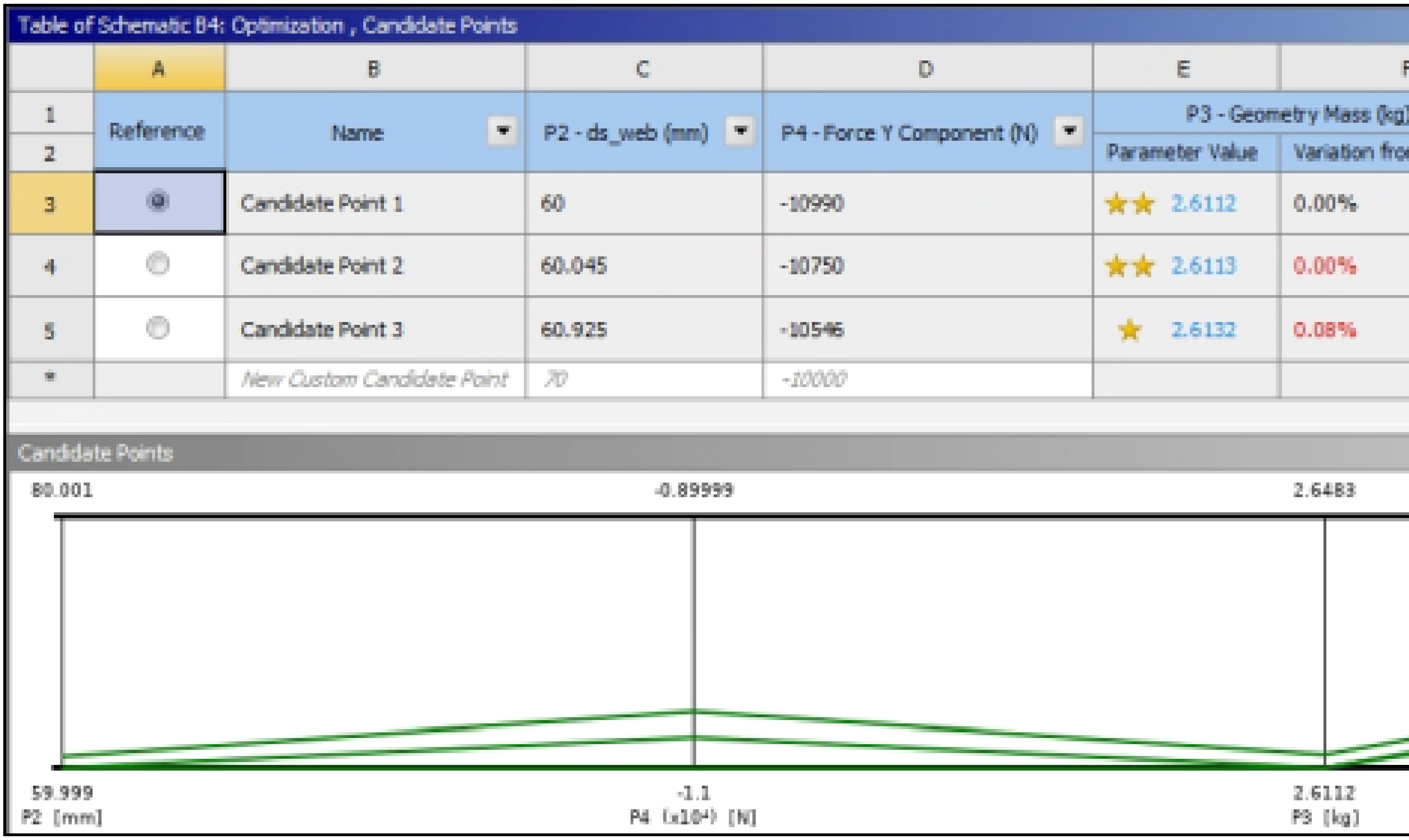
- To view the History chart for a parameter, in the **Outline** pane under either **Objectives and Constraints** or **Domain**, select the parameter to display the chart in the **Graph** pane.

For comprehensive information about optimization charts, see Goal-Driven Optimization Charts and Results in the *Ansys DesignXplorer User's Guide*.

Viewing Candidate Points

To view the candidate points found by the optimization, in the **Outline** pane under **Results**, select **Candidate Points**.

The **Table** pane displays candidate point data. In the following figure, **Candidate Point 1** is selected. The **Graph** pane plots the data for this candidate point. Placing the mouse cursor over the line for a candidate point highlights this line.



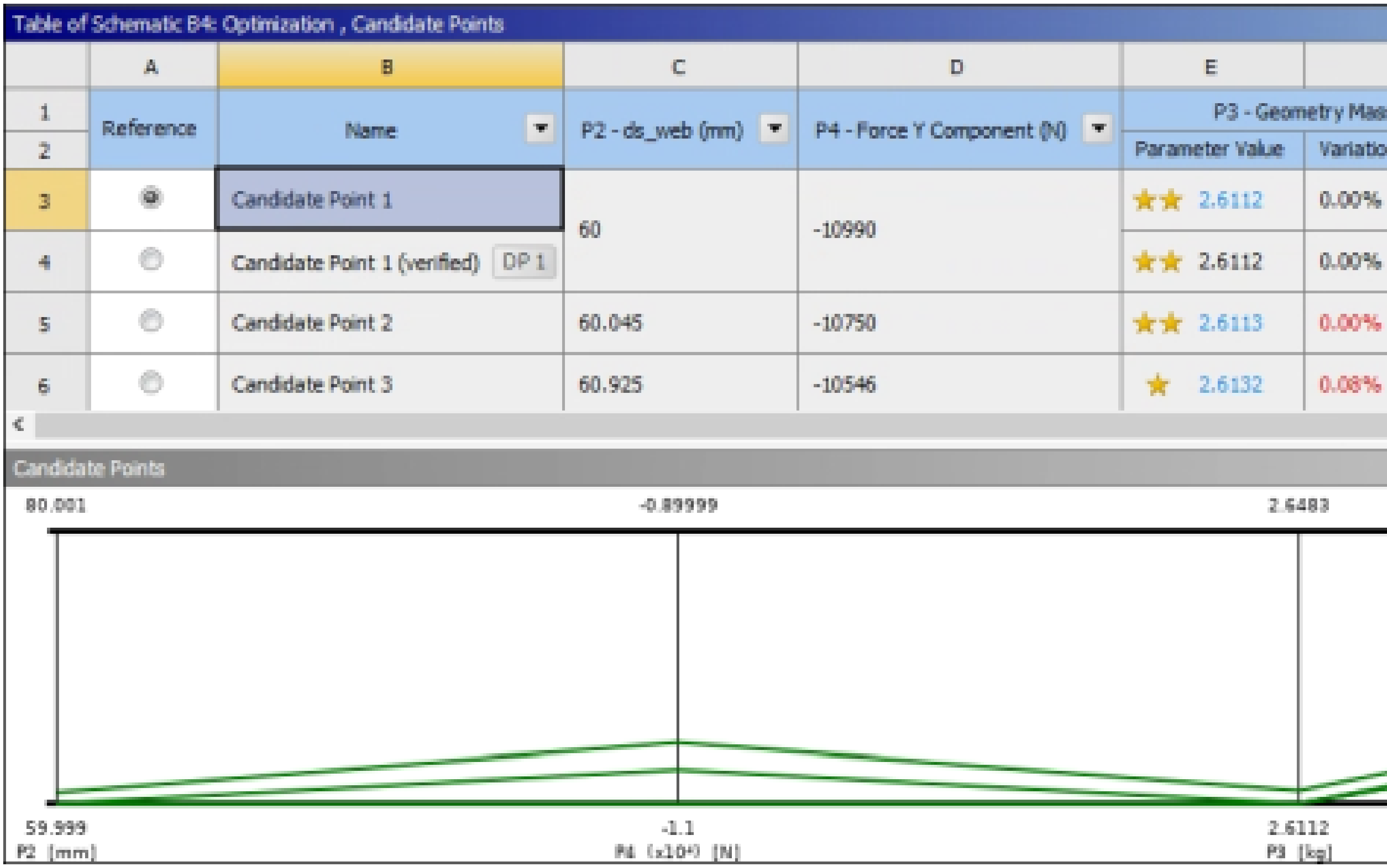
In the **Table** pane, you can see the input and output parameter values for the candidate points. The output parameter values calculated from simulations (design point updates) are displayed in black text. The output parameter values calculated from a response surface are displayed in the user-specified custom color, which is blue in the example. The number of gold stars or red crosses shown next to each goal-driven parameter indicate how well the parameter meets the stated goal, with three red crosses being the worst and three gold stars being the best.

For each parameter with an objective or constraint defined, the percentage of variation for all parameters with regard to an initial reference point is calculated. You can set any candidate point as the initial reference point by selecting the radio button in the **Reference** column.

The **Parameter Value** column displays the parameter value and stars indicating the quality of the candidate. In the **Variation from Reference** column, green text indicates variation in the expected direction. Red text indicates variation that is not in the expected direction. When there is no obvious direction (as for a constraint), the percentage value displays in black text.

Verifying the Most Viable Candidate Point

To verify that **Candidate Point 1** is the most viable update, right-click it and select **Verify by Design Point Update**. When the update is completed, the **Table** pane displays **Candidate Point 1 (verified)**, with solved values, next to the approximated values for **Candidate Point 1** so that you can easily compare them:



Viewing the Solution

A solution is generated for the design point designated as **(Current)** in the **Table** pane. If, based on your design exploration, you decide that another design point is preferable, you can define it as the current design point.

The following topics explain how to copy the inputs for **DP 1** to the current design point and then view results:

Viewing Results for the Most Viable Candidate Point
Viewing Results for the Response Point
Viewing Results in Mechanical

Viewing Results for the Most Viable Candidate Point

You'll now insert the most viable candidate point as a design point. Then, you'll copy its input values to the current design point.

1. In the **Outline** pane under **Results**, select **Candidate Points**.
2. Right-click **Candidate Point 1** and select **Insert as Design Point**.

   In the **Table** pane, **Candidate Point 1 (verified)** now has the label **DP 2** to its right.

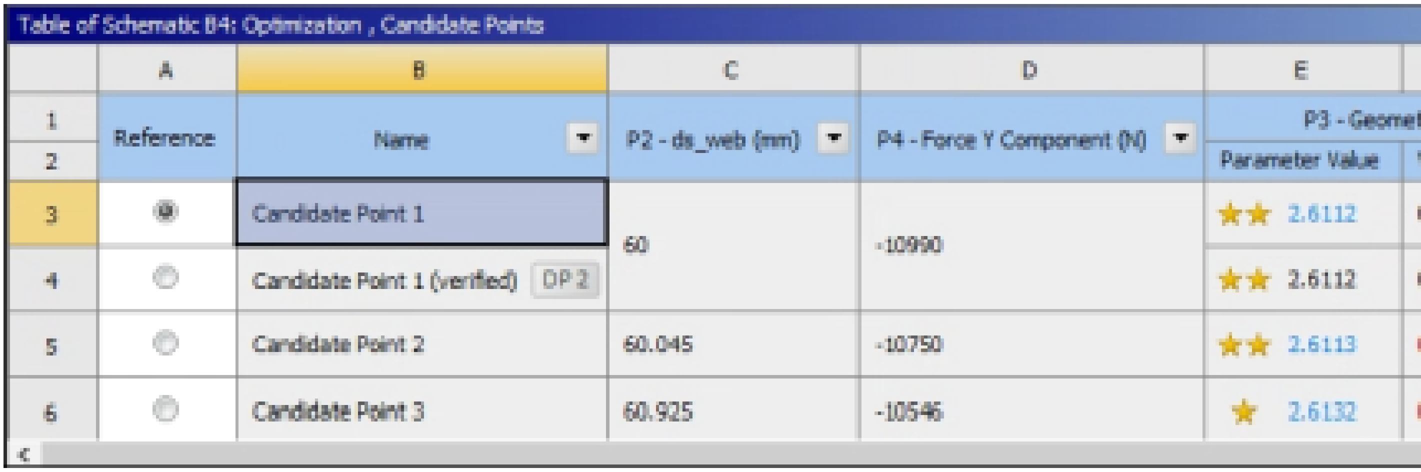


Table of Schematic B4: Optimization , Candidate Points

	A	B	C	D	E
1	Reference	Name	P2 - ds_web (mm)	P4 - Force Y Component (N)	P3 - Geomet
2					Parameter Value
3	◉	Candidate Point 1	60	-10990	★★ 2.6112
4	○	Candidate Point 1 (verified) DP 2			★★ 2.6112
5	○	Candidate Point 2	60.045	-10750	★★ 2.6113
6	○	Candidate Point 3	60.925	-10546	★ 2.6132

3. Close the **Optimization** cell.

4. On the **Project Schematic**, double-click the **Parameter Set** bar to open it.

5. In the **Table** pane, right-click **DP 1** and select **Copy inputs to Current**.

   The current design point now has the same input parameter values as your best candidate, **DP 1**. The **Note** column indicates that this design point is created from **Candidate Point 1** in the **Optimization** cell.

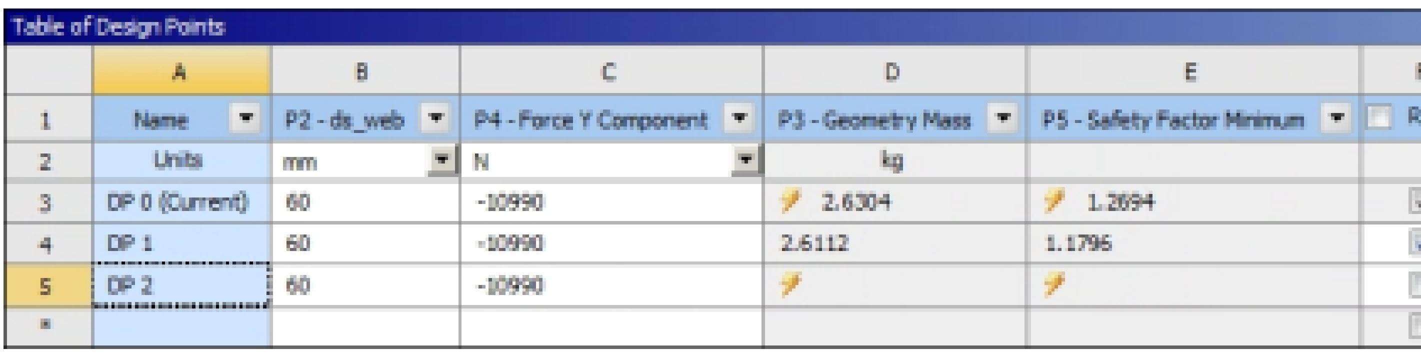


Table of Design Points

	A	B	C	D	E
1	Name	P2 - ds_web	P4 - Force Y Component	P3 - Geometry Mass	P5 - Safety Factor Minimum
2	Units	mm	N	kg	
3	DP 0 (Current)	60	-10990	2.6304	1.2694
4	DP 1	60	-10990	2.6112	1.1796
5	DP 2	60	-10990		
*					

6. Right-click the current design point and select **Update Selected Design Points**.

   Once the design point is updated, input values for the current design point are now the same as for **DP 1**. Updating only the current design point saves time and resources.

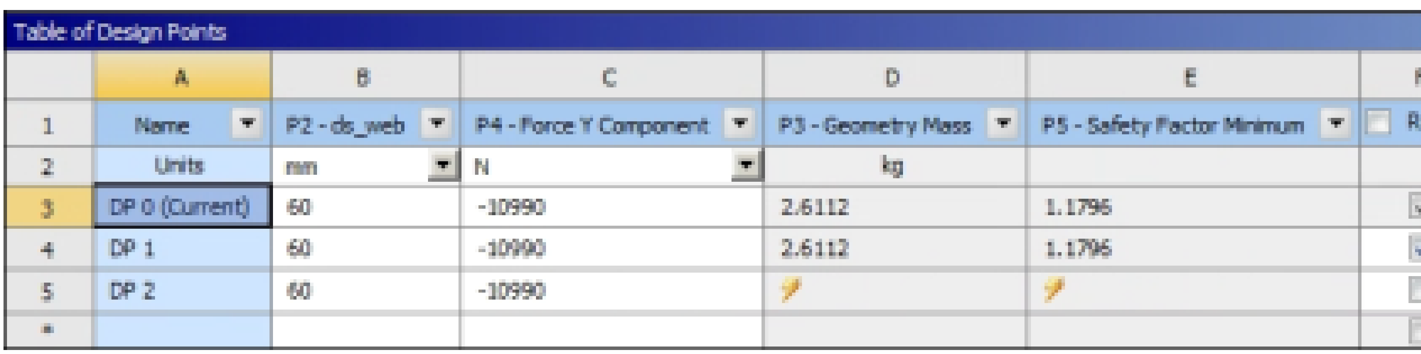


Table of Design Points

	A	B	C	D	E
1	Name	P2 - ds_web	P4 - Force Y Component	P3 - Geometry Mass	P5 - Safety Factor Minimum
2	Units	mm	N	kg	
3	DP 0 (Current)	60	-10990	2.6112	1.1796
4	DP 1	60	-10990	2.6112	1.1796
5	DP 2	60	-10990		
*					

7. For this tutorial, also update **DP 2** to check that the input values are the same:

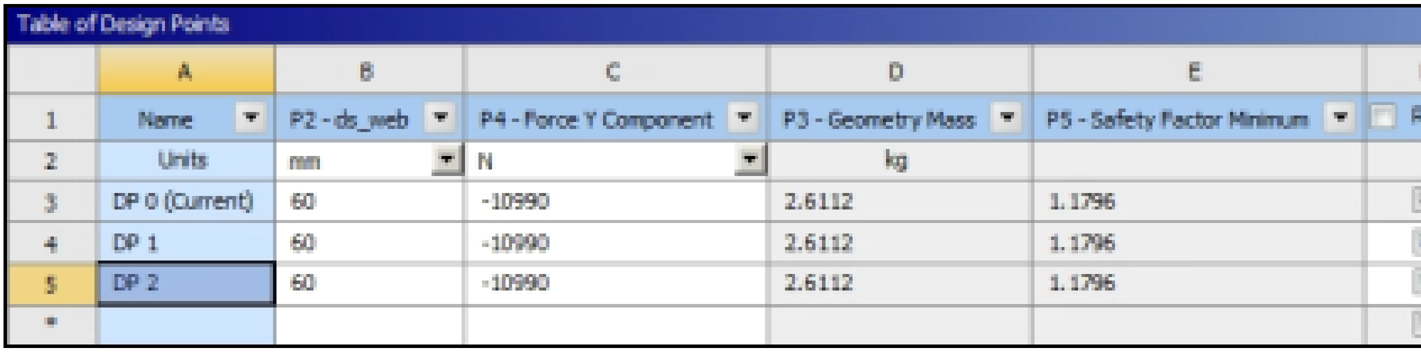
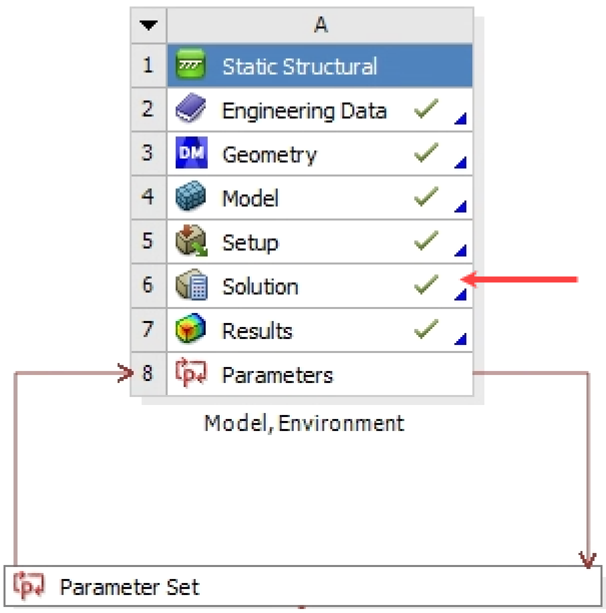


Table of Design Points

	A	B	C	D	E
1	Name	P2 - ds_web	P4 - Force Y Component	P3 - Geometry Mass	P5 - Safety Factor Minimum
2	Units	mm	N	kg	
3	DP 0 (Current)	60	-10990	2.6112	1.1796
4	DP 1	60	-10990	2.6112	1.1796
5	DP 2	60	-10990	2.6112	1.1796
*					

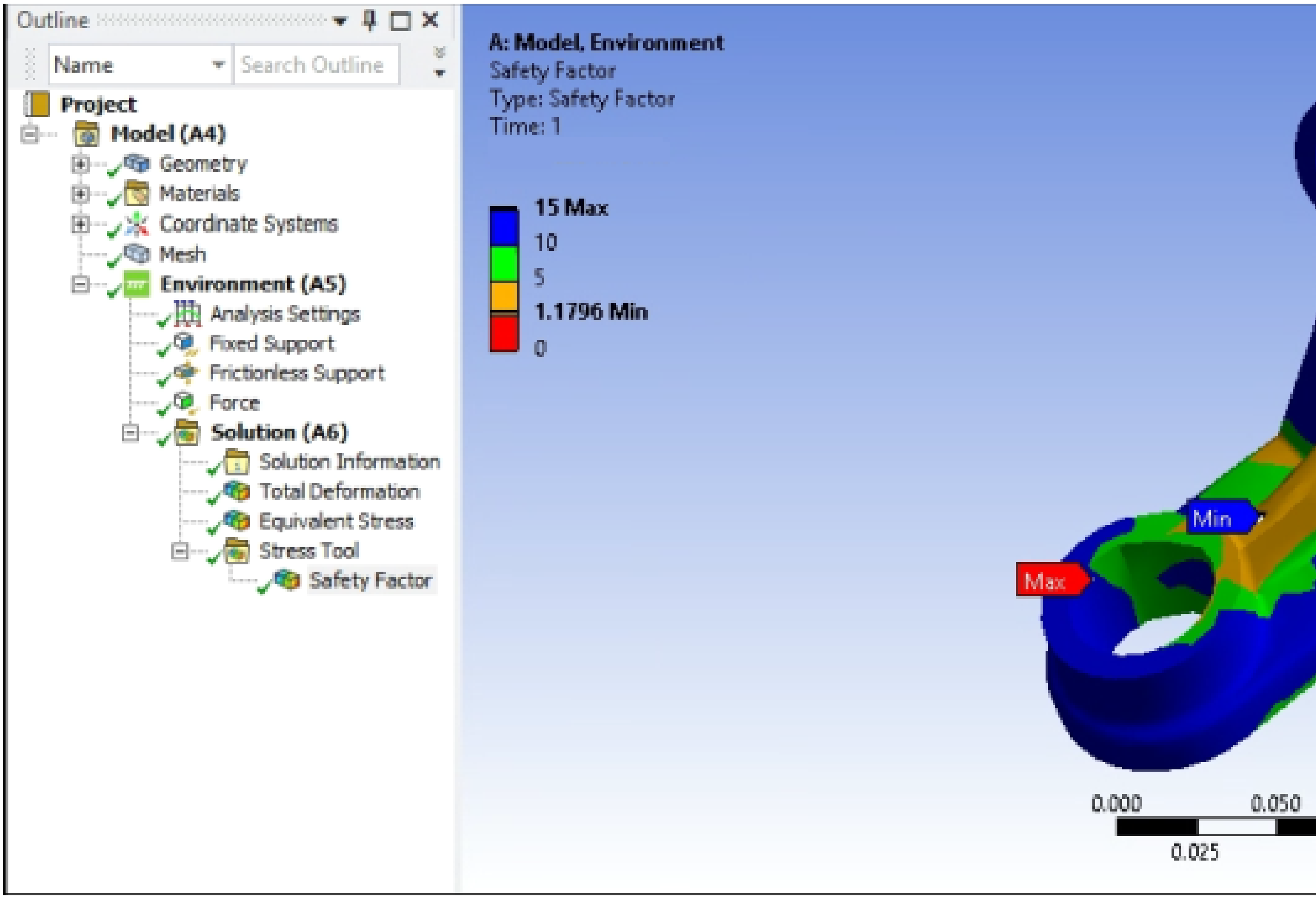
8. Close the **Parameter Set** bar.

9. In the **Static Structural** system, double-click the **Solution** cell to open it.



Mechanical starts.

10. In the **Outline** view, under **Solution** > **Stress Tool**, select **Safety Factor** to view the results.

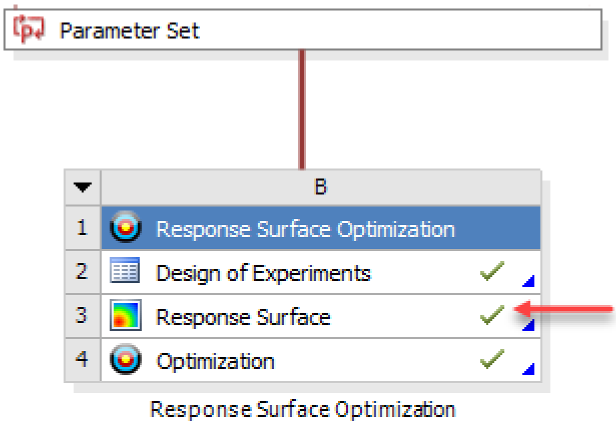


11. Exit Mechanical.

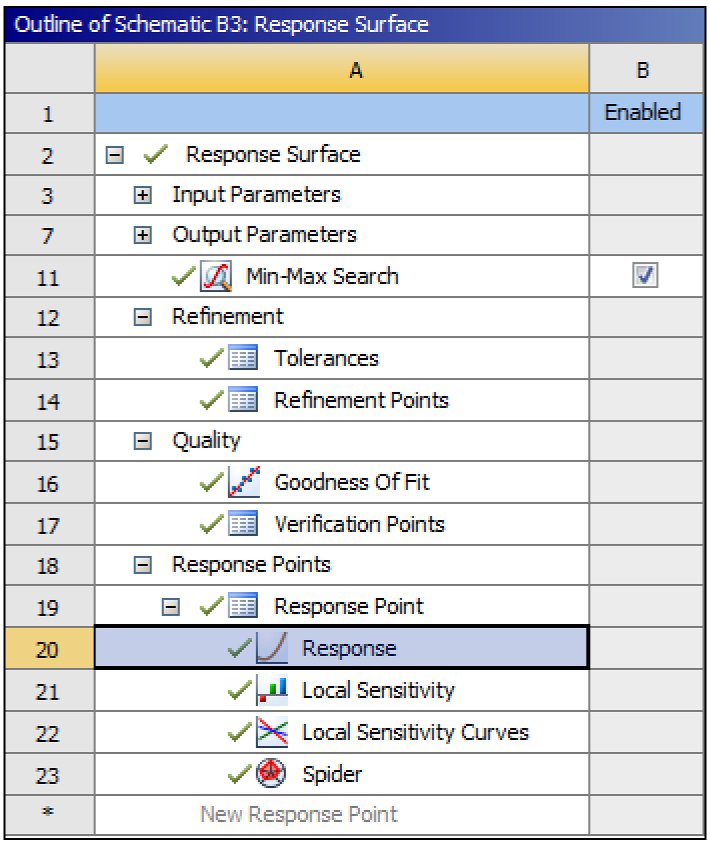
Viewing Results for the Response Point

Your next step is to view results and copy values for a selected point in the response surface to the response point in the **Table** pane. You'll then insert this response point as a design point, update it, and copy its values to the current design point. Lastly, you'll update the project.

1. In the **Response Surface Optimization** system, double-click the **Response Surface** cell to open it.



2. In the **Outline** pane under **Response Points** > **Response Point**, select **Response**, which displays the Response chart in the **Graph** pane.

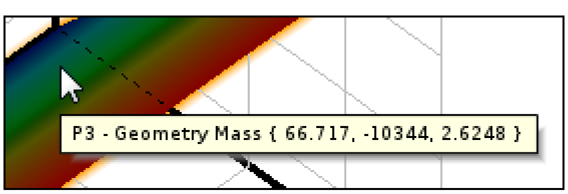


Outline of Schematic B3: Response Surface

	A	B
1		Enabled
2	Response Surface	
3	Input Parameters	
7	Output Parameters	
11	Min-Max Search	☑
12	Refinement	
13	Tolerances	
14	Refinement Points	
15	Quality	
16	Goodness Of Fit	
17	Verification Points	
18	Response Points	
19	Response Point	
20	Response	
21	Local Sensitivity	
22	Local Sensitivity Curves	
23	Spider	
*	New Response Point	

3. In the **Properties** pane for this chart, set **Mode** to **3D**.

4. In the **Graph** pane, move the mouse cursor over the chart to view data for various points.



5. Right-click a point and select **Copy Values to Response Point**.

   The values for this point are copied to the response point in the **Table** pane.

6. In the **Table** pane, right-click the response point and select **Insert as Design Point**.

7. Close the **Response Surface** cell.

8. On the **Project Schematic**, double-click the **Parameter Set** bar to open it.

   The **Table** pane now includes a design point named **DP 3**. The **Note** column indicates that this design point is created from the response surface.

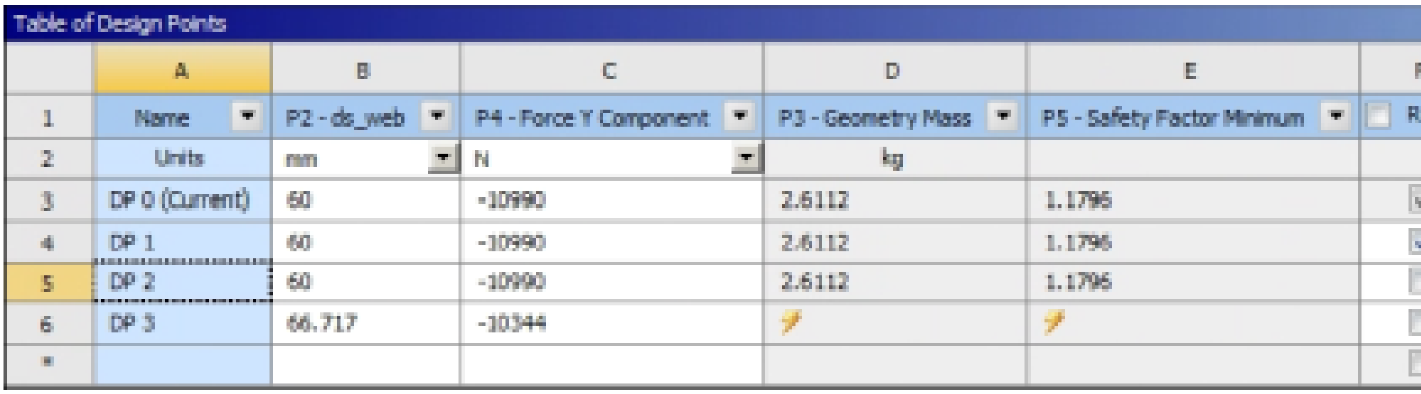


Table of Design Points

	A	B	C	D	E
1	Name	P2 - ds_web	P4 - Force Y Component	P3 - Geometry Mass	P5 - Safety Factor Minimum
2	Units	mm	N	kg	
3	DP 0 (Current)	60	-10990	2.6112	1.1796
4	DP 1	60	-10990	2.6112	1.1796
5	DP 2	60	-10990	2.6112	1.1796
6	DP 3	66.717	-10344		
*					

9. In the **Table** pane:

   a. For **DP 3**, select the **Retain** check box.

   b. Right-click **DP 3** and select **Update Selected Design Points**.

   c. Right-click **DP 3** and select **Set as Current**.

   Note:

   It is also possible to select **Export Selected Design Points** to view these results in a separate Workbench project.

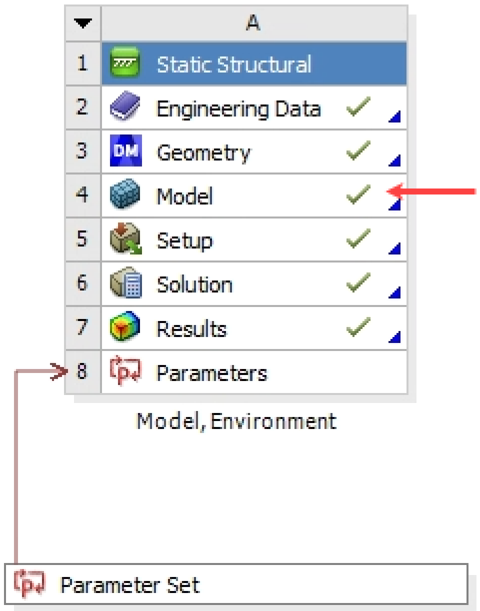
10. Close the **Parameter Set** bar.

11. On the **Project Schematic**, update the project.

Viewing Results in Mechanical

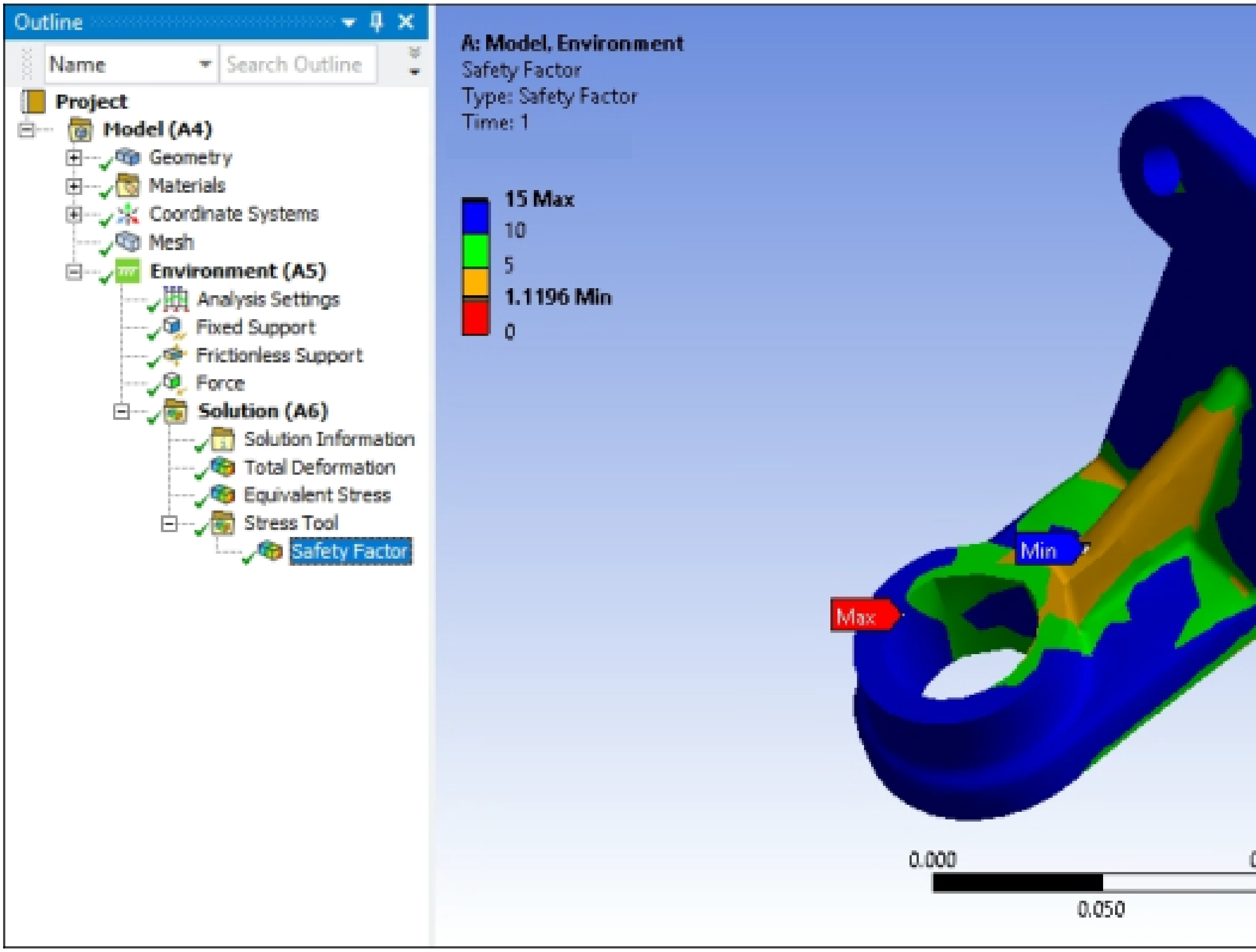
To view results in Mechanical:

1. In the **Static Structural** system, double-click the **Model** cell.



Mechanical starts.

2. In the **Outline** view under **Solution**, select **Stress Tool** > **Safety Factor** to view the results.



3. Exit Mechanical.

4. Exit Workbench, saving changes to the project.

Using ACT to Expose MATLAB Optimizers in DesignXplorer

DesignXplorer can delegate the resolution of an optimization to third-party optimizers. The optimizers are installed as ACT extensions and hosted in the DesignXplorer environment. When invoked, the installed optimizer becomes responsible for the resolution of the optimization. The complete definition of the optimization, along with utilities to trigger design point updates and provide progress information, is passed to the installed optimizer. This architecture enables the optimizer to implement any optimization strategy or workflow, from the simplest to the most complicated.

From the Ansys App Store, you can download the free app **MATLab Optmizers for DX** for your Ansys version. In the downloaded folder, the child folder `doc` contains a PDF file that explains how to install the extension and use the MATLAB optimizers that this ACT extension exposes in DesignXplorer.

You need no knowledge of ACT to use the MATLAB optimizers in this extension. You can use them exactly as they are delivered, without having to change or recompile the ACT extension. However, if these optimizers do not meet your need, the supplied PDF file explains how you can either customize the generic optimizer or modify source code to create your own ACT extension with custom MATLAB optimizers.

Using Adaptive Single-Objective Optimization

To choose the correct optimization method for a given problem, you must understand the problem. To understand the problem, you must first explore it, which requires the selection of an optimization method. DesignXplorer's *Adaptive Single-Objective (ASO)* optimization method is a robust, adaptive algorithm that simplifies this process, allowing you to explore your design space during an actual optimization run.

In this advanced tutorial, you'll use four different optimization scenarios, including one that uses ASO, to explore the design space and find the global optimum for the same problem. You'll examine the results and benefits of each method for solving this particular problem, learning how the performance of different algorithms in combination compare with the performance of ASO.

Note:

This advanced tutorial assumes that you are familiar with Ansys Workbench and with DesignXplorer's goal-driven optimization functionality. For an introduction to such optimizations, see the tutorial Performing a Goal-Driven Optimization (p. 79).

This tutorial is divided into the following sections:

What is Adaptive Single-Objective Optimization?

Problem Definition

Problem Setup

Scenario 1: Kriging-NLPQL Response Surface Optimization to NLPQL Direct Optimization

Scenario 2: NLPQL Direct Optimization to NLPQL Direct Optimization

Scenario 3: Screening Direct Optimization to NLPQL Direct Optimization

Scenario 4: Adaptive Single-Objective Direct Optimization

Time to Spare?

What Have You Learned?

What is Adaptive Single-Objective Optimization?

Adaptive Single-Objective (ASO) is a gradient-based mathematical optimization method that is available only for direct optimizations. It combines an Optimal Space-Filling (OSF) DOE, a Kriging response surface, and the Mixed-Integer Sequential Quadratic Programming (MISQP) optimization algorithm with domain reduction to locate the global optima.

When an optimization method is *adaptive*, it is internally powered by response surface technology. When the level of accuracy is not acceptable, ASO performs design point updates and refines the surface. When the level of accuracy is good enough, it uses approximation instead.

Problem Definition

For this tutorial, the problem is a non-convex analytic function with two input parameters. The definition of the problem is as follows:

Minimize

$$f(x_1,x_2)$$

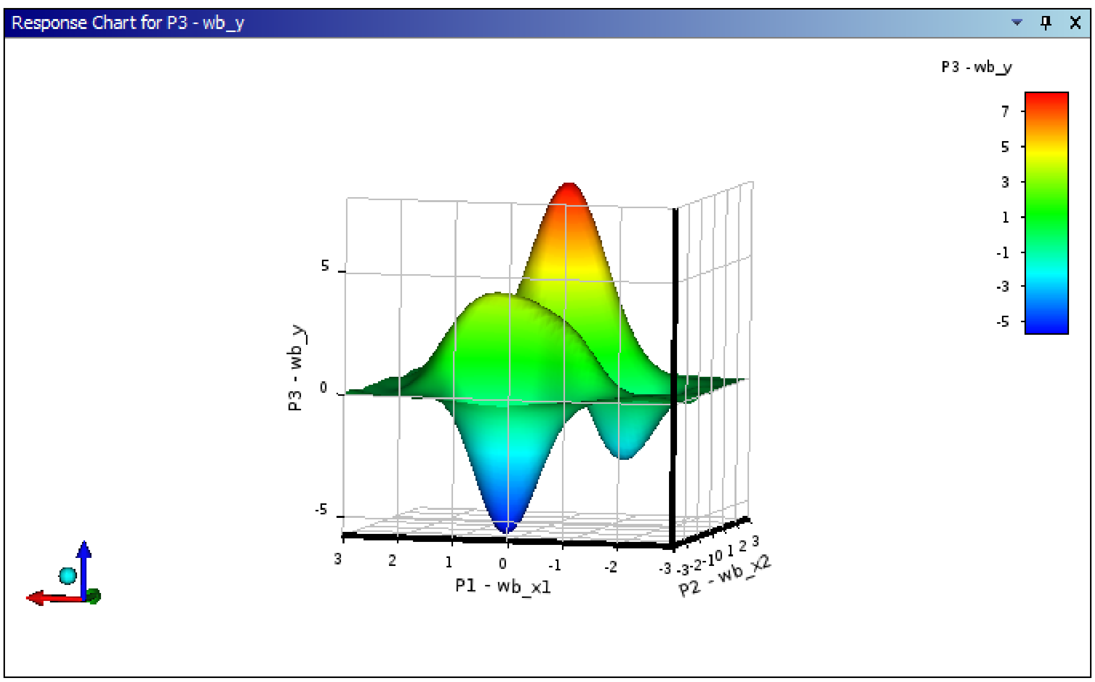
Where

$$-3.0 \le x_1, x_2 \le 3$$

And

$$f(x_1,x_2)=3(1-x_1)^2 e^{[-x_1^2-(x_2+1)^2]}-10\left(\frac{x_1}{5}-x_1^3-x_2^5\right)e^{[-x_1^2-x_2^2]}-\frac{1}{3}e^{[-(x_1+1)^2-x_2^2]}$$

This analytic function has three local maxima, one local minimum, and one global minimum point at (**0.2282**;**-1.6256**), with a corresponding objective function value of **-6.5511**.



Problem Setup

To create this project, you'll add an analysis system to the **Project Schematic** and then link the **Analysis** cell to an input file.

1. Start Workbench.

2. From under **Component Systems** in the **Toolbox**, add the system to the **Project Schematic**:

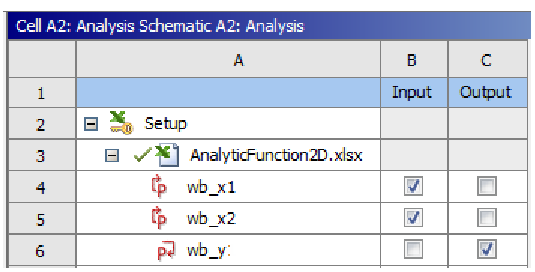
   • For Windows, add a **Microsoft Office Excel** system.

- For Linux, add a **Mechanical APDL** system.

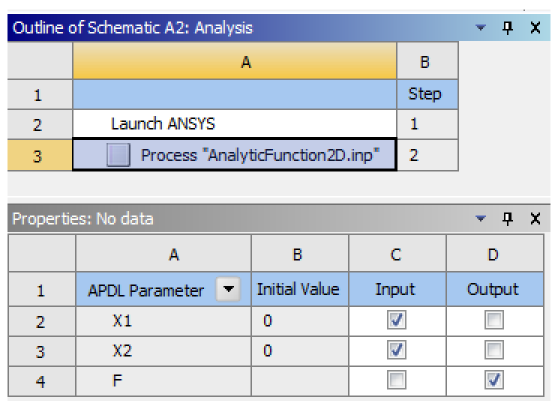
3. In this system, double-click the **Analysis** cell to open it.
4. In the **Outline** pane, right-click in the **Input** column for **Setup** and select **Add File** → **Browse**.
5. In the file browser, navigate to the directory to which you extracted the input files (p. 27) and open the appropriate file:
   - For Windows, open `AnalyticFunction2D.xlsx`.
   - For Linux, open `AnalyticFunction2D.inp`.
6. Define input and output parameters as shown.

> **Note:**
>
> The images in the remainder of this tutorial are for the XLSX file. The order in which you define parameters determines the system-generated names for the parameters. For input names to match those shown, you must select the **Input** check boxes for **wb_x1** first and **wb_x2** second. You then select the **Output** check box for **wb_y** last.

Excel



MAPDL



7. Close the **Analysis** cell. The **Project Schematic** now includes the **Parameter Set** bar.
8. Update the project.
9. Save the project as `DX_ASO.wbpj` in either this same directory or another directory.

In this project, you'll run four different optimization scenarios, compare their results, and determine which optimization method was best for this particular problem.

Scenario 1: Kriging-NLPQL Response Surface Optimization to NLPQL Direct Optimization

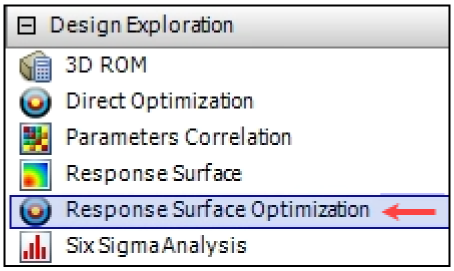
For the first scenario, you'll run a response surface optimization. You'll then use its results to run a direct optimization.

Run the Response Surface Optimization

You'll begin by adding a **Response Surface Optimization** system to the **Project Schematic**. You'll then configure each of its cells, run the optimization, and view results.

Add the Goal-Driven Optimization System

On the **Project Schematic**, in the **Toolbox** under **Design Exploration**, double-click **Response Surface Optimization**.



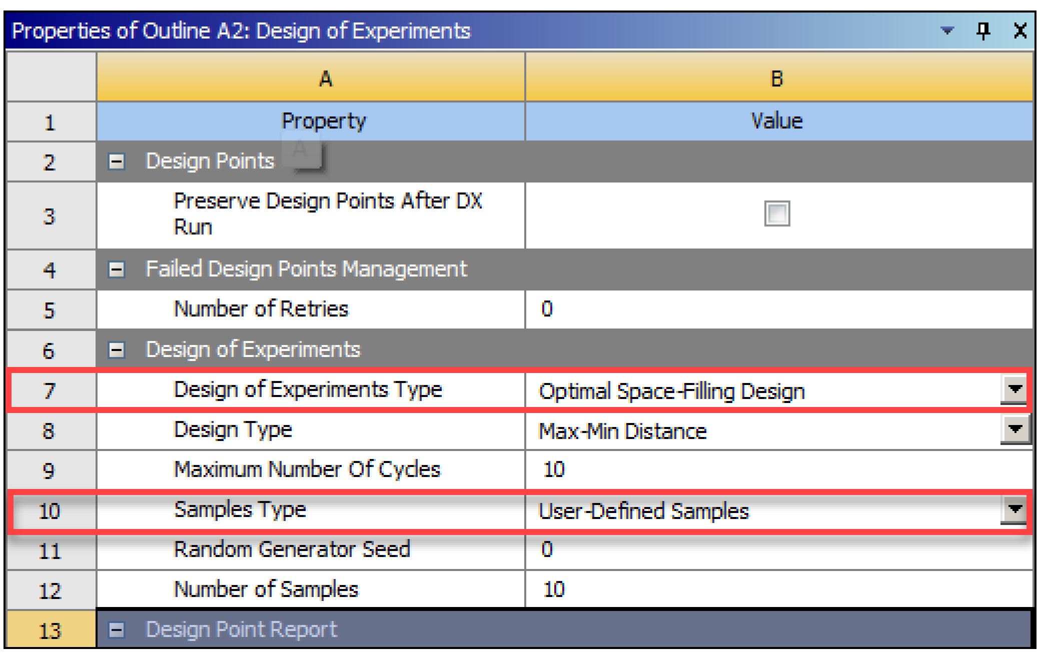
This system is added below the **Parameter Set** bar.

Configure and Update the Design of Experiments Cell

1. Double-click the **Design of Experiments** cell to open it.

2. In the **Outline** pane, select **Design of Experiments**.

3. In the **Properties** pane, set DOE properties as shown:



Properties of Outline A2: Design of Experiments

	A	B
1	Property	Value
2	Design Points	
3	Preserve Design Points After DX Run	☐
4	Failed Design Points Management	
5	Number of Retries	0
6	Design of Experiments	
7	Design of Experiments Type	Optimal Space-Filling Design
8	Design Type	Max-Min Distance
9	Maximum Number Of Cycles	10
10	Samples Type	User-Defined Samples
11	Random Generator Seed	0
12	Number of Samples	10
13	Design Point Report	

4. In the **Properties** pane for each input parameter, set lower and upper bounds as shown here for **P1 - wb_x1**:

Properties of Outline : P1 - wb_x1

	A	B
1	Property	Value
2	General	
3	Units	
4	Type	Design Variable
5	Classification	Continuous
6	Values	
7	Lower Bound	-3
8	Upper Bound	3
9	Allowed Values	Any

5. Update the **Design of Experiments** cell. Once the update completes, the **Table** pane displays the solved design points to be used as inputs for the response surface.

Table of Outline A6: Design Points of Design of Experiments

	A	B	C	D
1	Name	P1 - wb_x1	P2 - wb_x2	P3 - wb_y
2	1	-0.3	-0.3	2.9074
3	2	1.5	-2.7	-0.096589
4	3	-2.7	-1.5	0.00064371
5	4	0.9	0.9	2.2514
6	5	-2.1	0.3	-1.0081
7	6	2.7	1.5	0.01924
8	7	0.3	2.7	0.89455
9	8	2.1	-0.9	0.48981
10	9	-0.9	-2.1	-0.80545
11	10	-1.5	2.1	0.48081

6. Close the **Design of Experiments** cell.

Configure and Update the Response Surface Cell

Because the problem is a type of function that cannot be approximated with a quadratic response surface, you must select an alternate type of response surface. Kriging is a good choice because it can approximate the function by using automatic refinement to enrich the response surface and obtain the required accuracy.

1. On the **Project Schematic**, double-click the **Response Surface** cell to open it.

2. In the **Outline** pane, select **Response Surface**.

3. In the **Properties** pane, set properties as shown:

Properties of Outline A2: Response Surface		
	A	B
1	Property	Value
2	Design Points	
3	Preserve Design Points After DX Run	☐
4	Failed Design Points Management	
5	Number of Retries	0
6	Meta Model	
7	Response Surface Type	Kriging
8	Kernel Variation Type	Variable
9	Refinement	
10	Refinement Type	Auto
11	Maximum Number of Refinement Points	100
12	Number of Refinement Points	0
13	Maximum Predicted Relative Error (%)	5
14	Verification Points	

4. In the **Outline** pane under **Output Parameters**, ensure that the check box for **Min-Max Search** is selected.

5. Update the **Response Surface** cell. When the update is complete, the **Table** pane displays a response point. The value for output **P3 - wb_y** is **3.0293**.

View Results

1. In the **Outline** pane, select **Response Surface**.

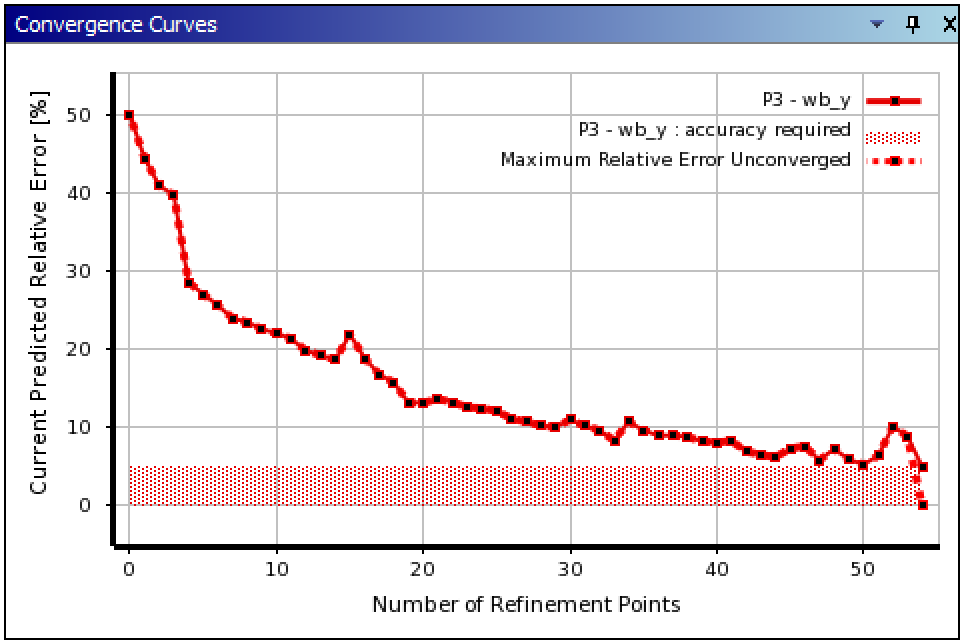
   In the **Properties** pane under **Refinement**, **Number of Refinement Points** displays **54**. This indicates that Kriging with automatic refinement converged after 54 additional refinement points were created.

2. In the **Outline** pane under **Output Parameters**, select **Min-Max Search**.

   The **Table** pane displays the approximate value of the objective function (**-5.8017**) and the parameter minimums (**P1 = 0.35346** and **P2 = –1.5925**). You will use these minimum values to initialize the **Optimization** cell of the **Response Surface Optimization** system.

Table of Outline A10: Min-Max Search				
	A	B	C	D
1	Name	P1 - wb_x1	P2 - wb_x2	P3 - wb_y
2	Output Parameter Minimums			
3	P3 - wb_y	0.35346	-1.5925	-5.8017
4	Output Parameter Maximums			
5	P3 - wb_y	0.017654	1.5878	8.1246

3. In the **Outline** pane, select **Refinement**. The **Graph** pane displays the Convergence Curves chart.



4. To view the three-dimensional Kriging response surface:

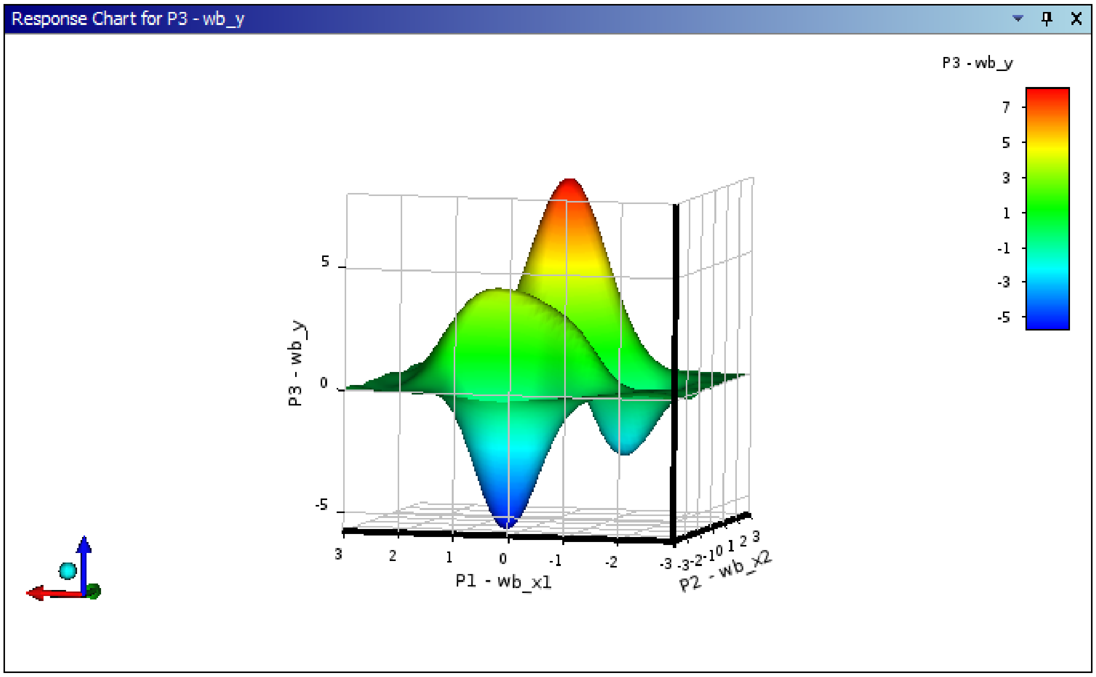
   a. In the **Outline** pane under **Response Points** → **Response Point**, select **Response** to display this chart.

   b. In the **Properties** pane:

      - Set **Mode** to **3D**.
      - Set **Chart Resolution Along X** and **Chart Resolution Along Y** to **50**.

      The chart displays the three-dimensional response surface.

   c. In the **Chart** pane, rotate the chart so that only the Y axis and Z axis are visible.



5. To view the design point distribution of the Kriging response surface:

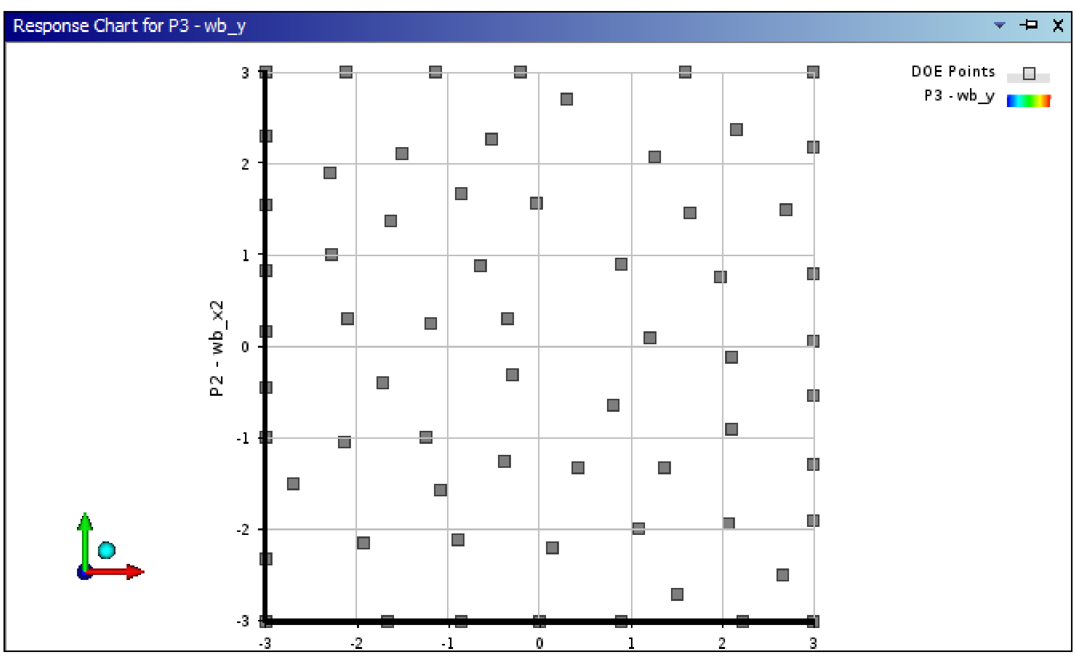
   a. In the **Properties** pane, select the **Show Design Points** check box.

   b. Right-click the response surface and select **Edit Properties**.

   c. In the **Properties** pane for the chart, set **Display As** to **None**.

   d. In the **Chart** pane, rotate the chart so that only the X axis and Y axis are visible.

      The chart displays the distribution of design points in the response surface as shown:



6. Close the **Response Surface** cell.

Configure and Update the Optimization Cell

1. On the **Project Schematic**, double-click the **Optimization** cell to open it.
2. In the **Outline** pane, select **Optimization**.
3. In the **Properties** pane, set optimization properties as shown:

Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	Design Points	
3	Preserve Design Points After DX Run	☐
4	Failed Design Points Management	
5	Number of Retries	0
6	Optimization	
7	Method Selection	Manual
8	Method Name	NLPQL
9	Estimated Number of Evaluations	100
10	Tolerance Settings	☑
11	Verify Candidate Points	☐
12	Finite Difference Approximation	Central
13	Allowable Convergence (%)	0.0001
14	Maximum Number of Iterations	20
15	Maximum Number of Candidates	3

4. In the **Outline** pane, select **Objectives and Constraints**.
5. In the **Table** pane, for **Parameter**, select **P3 - wb_y** and then set the objective type to **Minimize** and **Target** to **-6**:

Table of Schematic B4: Optimization

	A	B	C	D	E	F
1	Name	Parameter	Objective			
2			Type	Target	Tolerance	Type
3	Minimize P3	P3 - wb_y	Minimize	-6		No Constraint
*		Select a Parameter				

6. In the **Outline** pane, select **Domain**.
7. In the **Table** pane, set **Starting Value** for each input parameter to the minimum found earlier by the Min-Max Search:

Table of Schematic B4: Optimization

	A	B	C	D
1	⊟ Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - wb_x1	-3	3	0.35346
4	P2 - wb_x2	-3	3	-1.5925
5	⊟ Parameter Relationships			
6	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	*<=*	*New Expression*

8. Update the **Optimization** cell.

View Results

1. When the update is complete, in the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, **Number of Evaluations** shows that the optimization has converged after 14 evaluations.

2. In the **Outline** pane under **Results**, select **Candidate Points**. The **Table** pane shows that the best candidate is the original NLPQL **Starting Point**. This is expected because the Min-Max Search is based on NLPQL.

Table of Schematic B4: Optimization , Candidate Points

	A	B	C	D	E	F
1	Reference	Name	P1 - wb_x1	P2 - wb_x2	P3 - wb_y	
2					Parameter Value	Variation from Reference
3	◉	Starting Point	0.35346	-1.5925	-5.8017	0.00%
4	○	Candidate Point 1	0.35346	-1.5925	-5.8017	0.00%
5	○	Candidate Point 2	0.35346	-1.5925	-5.8017	0.00%
6	○	Candidate Point 3	0.35346	-1.5925	-5.8017	0.00%
*		*New Custom Candidate Point*	*0*	*0*		

3. Right-click **Candidate Point 1** and select **Verify by Design Point Update**.

   The objective function value for the verified point is **-6.4009**.

Table of Schematic B4: Optimization , Candidate Points

	A	B	C	D	E	F
1	Reference	Name	P1 - wb_x1	P2 - wb_x2	P3 - wb_y	
2					Parameter Value	Variation from Reference
3	◉	Starting Point	0.35346	-1.5925	★★★ -5.8017	0.00%
4	○	Candidate Point 1	0.35346	-1.5925	★★★ -5.8017	0.00%
5	○	Candidate Point 1 (verified)			★★★ -6.4009	-10.33%
6	○	Candidate Point 2	0.35346	-1.5925	★★★ -5.8017	0.00%
7	○	Candidate Point 3	0.35346	-1.5925	★★★ -5.8017	0.00%
*		*New Custom Candidate Point*	*0*	*0*		

4. Close the **Optimization** cell.

Run the Direct Optimization

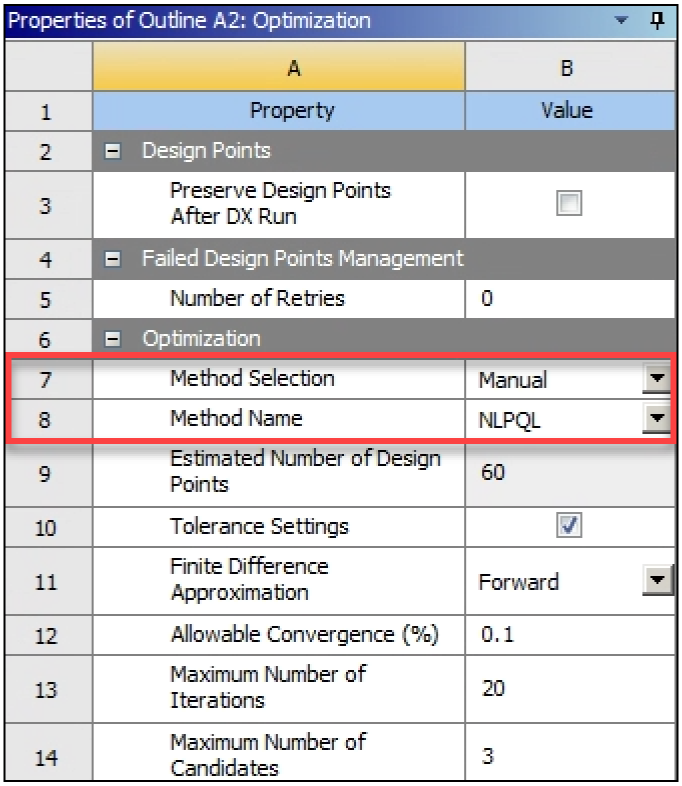
Next, you'll add, configure, and run a **Direct Optimization** system that uses NLPQL. Although NLPQL is dependent on the **Starting Point**, you can get reasonable starting points for the inputs by using the results of the response surface optimization. Also, you can use response surface exploration to reduce the domain of the direct optimization +/-0.3 in each direction.

1. From under **Design Exploration Systems** in the **Toolbox**, double-click **Direct Optimization** to add this system to the **Project Schematic**.

2. In the **Direct Optimization** system, double-click the **Optimization** cell to open it.

3. In the **Outline** pane, select **Optimization**.

4. In the **Properties** pane, set optimization properties as shown:



Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	Design Points	
3	Preserve Design Points After DX Run	☐
4	Failed Design Points Management	
5	Number of Retries	0
6	Optimization	
7	Method Selection	Manual
8	Method Name	NLPQL
9	Estimated Number of Design Points	60
10	Tolerance Settings	☑
11	Finite Difference Approximation	Forward
12	Allowable Convergence (%)	0.1
13	Maximum Number of Iterations	20
14	Maximum Number of Candidates	3

5. In the **Outline** pane, select **Objectives and Constraints**.

6. In the **Table** pane, for **Parameter**, select **P3 - wb_y** and then set the objective type to **Minimize** and **Target** to **-6**:

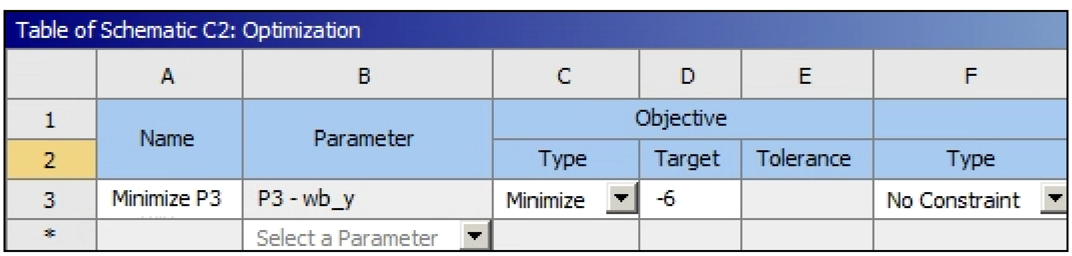


Table of Schematic C2: Optimization

	A	B	C	D	E	F
1	Name	Parameter	Objective			
2			Type	Target	Tolerance	Type
3	Minimize P3	P3 - wb_y	Minimize	-6		No Constraint
*		Select a Parameter				

7. In the **Outline** pane, select **Domain**.

8. In the **Table** pane, assign values for the input parameters as shown:

Table of Schematic C2: Optimization				
	A	B	C	D
1	Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - wb_x1	0.053447	0.65345	0.35346
4	P2 - wb_x2	-1.8925	-1.2925	-1.5925
5	Parameter Relationships			
6	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	<=	*New Expression*

Note:

The starting values are set to the minimums found earlier by the response surface Min-Max Search. The lower and upper bounds are set to +0.3 and –0.3 of the starting values.

9. Update the **Optimization** cell.

View Results

1. When the update is complete, in the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, you can see that the optimization has converged. Four iterations and 15 design points were needed to find the minimum.

2. In the **Outline** pane under **Results**, select **Candidate Points**. The **Table** pane shows that **Candidate Point 1** now has an objective function value of **-6.5511**.

Table of Schematic C2: Optimization , Candidate Points						
	A	B	C	D	E	F
1	Reference	Name	P1 - wb_x1	P2 - wb_x2	P3 - wb_y	
2					Parameter Value	Variation from Reference
3	◉	Starting Point	0.35346	-1.5925	★★★ -6.4009	0.00%
4	○	Candidate Point 1	0.22761	-1.6275	★★★ -6.5511	-2.35%
5	○	Candidate Point 2	0.27816	-1.6678	★★★ -6.4951	-1.47%
6	○	Candidate Point 3	0.35346	-1.5925	★★★ -6.4009	0.00%
*		*New Custom Candidate Point*	*0.35345*	*-1.5925*		

3. Close the **Optimization** cell.

How Effective was the Approach Used for Scenario 1?

A response surface optimization is a good way to explore the design space. However, for this problem, it is expensive in terms of the number of design points required (80):

- 10 to build the DOE
- 54 to enrich the Kriging response surface
- 1 to verify the candidate
- 15 to run the NLPQL direct optimization

Once built, the Kriging response surface does allow you to find the area containing the global minimum, but the response surface optimization alone cannot obtain an accurate candidate point unless more design points are generated to further enrich the Kriging response surface.

Running an NLPQL direct optimization afterward, with the candidate from the response surface optimization as the starting point and with a reduced domain, is a good way to get more accuracy from the response surface-based approach.

Scenario 2: NLPQL Direct Optimization to NLPQL Direct Optimization

In Scenario 2, you'll begin by adding and running a **Direct Optimization** system that uses NLPQL. You'll then run a second direct optimization that is the same as the first, except with a different starting point.

Run the First NLPQL Direct Optimization

You'll begin by adding, configuring, and running the first NLPQL direct optimization:

1. From under **Design Exploration** in the **Toolbox**, double-click **Direct Optimization** to add this system to the **Project Schematic**.
2. Rename the system to `Scenario 2: First NLPQL Direct Optimization`.
3. In the system, double-click the **Optimization** cell to open it.
4. In the **Outline** pane, select **Optimization**.
5. In the **Properties** pane, specify optimization properties as shown:

Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	Design Points	
3	Preserve Design Points After DX Run	☐
4	Failed Design Points Management	
5	Number of Retries	0
6	Optimization	
7	Method Selection	Manual
8	Method Name	NLPQL
9	Estimated Number of Design Points	60
10	Tolerance Settings	☑
11	Finite Difference Approximation	Forward
12	Allowable Convergence (%)	0.1
13	Maximum Number of Iterations	20
14	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.

7. In the **Table** pane, for **Parameter**, select **P3 - wb_y** and then set the objective type to **Minimize** and **Target** to **-6**:

Table of Schematic D2: Optimization

	A	B	C	D	E	F
1	Name	Parameter	Objective			
2			Type	Target	Tolerance	Type
3	Minimize P3	P3 - wb_y	Minimize	-6		No Constraint
*		Select a Parameter				

8. In the **Outline** pane, select **Domain**.

9. In the **Table** pane, set input parameter properties as shown:

Table of Schematic D2: Optimization

	A	B	C	D
1	Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - wb_x1	-3	3	0
4	P2 - wb_x2	-3	3	0
5	Parameter Relationships			
6	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	<=	*New Expression*

10. Update the **Optimization** cell.

When the update is complete, the **Table** pane displays 14 design points and their results.

View Results

1. In the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, you can see that 4 iterations and 14 design points were needed to find the minimum.

   In the **Table** pane, you can see that the objective function of **Candidate Point 1** is **0.012741**.

Table of Schematic D2: Optimization

	A	B	C	D	E
1	Optimization Study				
2	Minimize P3	Goal, Minimize P3 (Default importance)			
3	Optimization Method				
4	NLPQL	The NLPQL method (Nonlinear Programming by Quadratic Lagrangian) is a gradient-based algorithm to provide a refined, local, optimization result. It supports a single objective, multiple constraints and is limited to continuous parameters. The starting point must be specified to determine the region of the design space to explore.			
5	Configuration	Approximate derivatives by Forward difference and find 3 candidates in a maximum of 20 iterations.			
6	Status	Converged after 14 evaluations.			
7	Candidate Points				
8		Starting Point DP 0	Candidate Point 1	Candidate Point 2	Candidate Point 3 DP 0
9	P1 - wb_x1	0	1.7833	1.821	0
10	P2 - wb_x2	0	3	1.6633	0
11	P3 - wb_y	0.98101	0.012741	0.42007	0.98101

2. Close the **Optimization** cell.

Run the Second NLPQL Direct Optimization

Next, you'll add, configure, and run the second NLPQL direct optimization. While you are going to change the starting point, you will not be using the results of the last optimization because the objective function obtained was not close enough to the expected value to be usable.

1. From under **Design Exploration** in the **Toolbox**, double-click **Direct Optimization** to add this system to the **Project Schematic**.
2. Rename the system to **`Scenario 2: Second NLPQL Direct Optimization`**.
3. In the system, double-click the **Optimization** cell to open it. The next four steps are the same as for the first NLPQL direct optimization.
4. In the **Outline** pane, select **Optimization**.
5. In the **Properties** pane, specify optimization properties as shown:

Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	NLPQL
9	Estimated Number of Design Points	60
10	Tolerance Settings	☑
11	Finite Difference Approximation	Forward
12	Allowable Convergence (%)	0.1
13	Maximum Number of Iterations	20
14	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.

7. In the **Table** pane, for **Parameter**, select **P3 - wb_y** and then set the objective type to **Minimize** and set **Target** to **-6**:

Table of Schematic D2: Optimization

	A	B	C	D	E	F
1	Name	Parameter	Objective			
2			Type	Target	Tolerance	Type
3	Minimize P3	P3 - wb_y	Minimize	-6		No Constraint
*		Select a Parameter				

8. In the **Outline** pane, select **Domain**.

9. In the **Table** pane, set input parameter parameters as shown:

Table of Schematic E2: Optimization				
	A	B	C	D
1	⊟ Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - wb_x1	-3	3	0
4	P2 - wb_x2	-3	3	-2
5	⊟ Parameter Relationships			
6	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	<=	*New Expression*

Note:

The starting values are randomly selected. They are not based on the results of the previous direct optimization.

---

10. Update the **Optimization** cell.

When the update is complete, the **Table** pane displays 88 design points and their results.

View Results

1. In the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, you can see that 13 iterations and 88 design points were needed to find the global minimum.

   In the **Table** pane, you can see that the objective function of **Candidate Point 1** is **-6.5511**, which is the expected value.

Table of Schematic E2: Optimization					
	A	B	C	D	E
1	⊟ Optimization Study				
2	Minimize P3	Goal, Minimize P3 (Default importance)			
3	⊟ Optimization Method				
4	NLPQL	The NLPQL method (Nonlinear Programming by Quadratic Lagrangian) is a gradient-based algorithm to provide a refined, local, optimization result. It supports a single objective, multiple constraints and is limited to continuous parameters. The starting point must be specified to determine the region of the design space to explore.			
5	Configuration	Approximate derivatives by Forward difference and find 3 candidates in a maximum of 20 iterations.			
6	Status	Converged after 88 evaluations.			
7	⊟ Candidate Points				
8		Starting Point	Candidate Point 1	Candidate Point 2	Candidate Point 3
9	P1 - wb_x1	0	0.22832	0.51723	0.65805
10	P2 - wb_x2	-2	-1.6255	-1.6528	-1.4529
11	P3 - wb_y	★★ -4.7596	★★★ -6.5511	★★★ -5.7805	★★ -4.7827

2. Close the **Optimization** cell.

How Effective was the Approach Used for Scenario 2?

In this scenario, the problem required 102 design points, 14 for the first NLPQL direct optimization and 88 for the second NLPQL direct optimization. If you didn't know the global minimum ahead of time, you might think that the results of the first NLPQL direct optimization are good—until you run the second one and achieve a better result.

The two optimizations in this scenario illustrate the importance of the **Starting Value** in a gradient-based method optimization such as NLPQL, especially when the objective function is not convex and contains several local optima. Because this problem has one local minimum and one global minimum, NLPQL alone cannot find the global optimum without a good starting point. This is also true for the Mixed-Integer Sequential Quadratic Programming (MISQP) optimization method.

Scenario 3: Screening Direct Optimization to NLPQL Direct Optimization

In Scenario 3, you'll use what you learned from Scenario 2: NLPQL needs to have a good starting point. You'll begin by running a Screening direct optimization to explore the design space. You will then use results from this optimization as the starting point for running an NLPQL direct optimization.

Run the Screening Direct Optimization

You'll begin by adding, configuring, and updating the Screening direct optimization:

1. From under **Design Exploration** in the **Toolbox**, double-click **Direct Optimization** to add this system to the **Project Schematic**.
2. Rename the system to `Scenario 3: Screening Direct Optimization`.
3. In the system, double-click the **Optimization** cell to open it.
4. In the **Outline** pane, select **Optimization**.
5. In the **Properties** pane, specify optimization properties as shown:

Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	Screening
9	Estimated Number of Design Points	20
10	Tolerance Settings	☑
11	Number of Samples	20
12	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.

7. In the **Table** pane, for **Parameter**, select **P3 - wb_y** and then set the objective type to **Minimize** and **Target** to **-6**:

Table of Schematic F2: Optimization

	A	B	C	D	E	F
1	Name	Parameter	Objective			
2			Type	Target	Tolerance	Type
3	Minimize P3	P3 - wb_y	Minimize	-6		No Constraint
*		Select a Parameter				

8. In the **Outline** pane, select **Domain**.

9. In the **Properties** pane, set upper and lower bounds as shown:

Table of Schematic F2: Optimization

	A	B	C	D
1	⊟ Input Parameters			
2	Name	Lower Bound	Upper Bound	
3	P1 - wb_x1	-3	3	
4	P3 - wb_x2	-3	3	
5	⊟ Parameter Relationships			
6	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	<=	*New Expression*

10. Update the **Optimization** cell.

When the update is complete, the **Table** pane displays 20 design points and their results.

View Results

1. In the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, you can see that Screening generated a sample set of 20 design points and identified 3 candidate points.

   In the **Table** pane, you can see **Candidate Point 1** is the best candidate. For this candidate point, parameter **P1 - wb_x1** has a value of **0.75**, parameter **P2 - wb_x2** has a value of **–1.725**, and the objective value of the function (output **P3 - wb_y**) is **-4.2983**.

   Table of Schematic F2: Optimization

	A	B	C	D
1	⊟ Optimization Study			
2	Minimize P3	Goal, Minimize P3 (Default importance)		
3	⊟ Optimization Method			
4	Screening	The Screening optimization method uses a simple approach based on sampling and sorting. It supports multiple objectives and constraints as well as all types of input parameters. Usually it is used for preliminary design, which may lead you to apply other methods for more refined optimization results.		
5	Configuration	Generate 20 samples and find 3 candidates.		
6	Status	Converged after 20 evaluations.		
7	⊟ Candidate Points			
8		Candidate Point 1	Candidate Point 2	Candidate Point 3
9	P1 - wb_x1	0.75	0.15	-1.35
10	P2 - wb_x2	-1.725	-0.975	0.9
11	P3 - wb_y	★★ -4.2983	-1.3467	-1.209

2. Close the **Optimization** cell.

Run the NLPQL Direct Optimization

Next, you'll add, configure, and update the NLPQL direct optimization:

1. From under **Design Exploration** in the **Toolbox**, double-click **Direct Optimization** to add this system to the **Project Schematic**.
2. Rename the system to **`Scenario 3: NLPQL Direct Optimization`**.
3. In the system, double-click the **Optimization** cell to open it.
4. In the **Outline** pane, select **Optimization**.
5. In the **Properties** pane, set optimization properties as shown:

Properties of Outline A2: Optimization		
	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	NLPQL
9	Estimated Number of Design Points	60
10	Tolerance Settings	☑
11	Finite Difference Approximation	Forward
12	Allowable Convergence (%)	0.1
13	Maximum Number of Iterations	20
14	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.
7. In the **Table** pane, for **Parameter**, select **P3 - wb_y** and then set the objective type to **Minimize** and **Target** to **-6**:

Table of Schematic G2: Optimization						
	A	B	C	D	E	F
1	Name	Parameter	Objective			
2			Type	Target	Tolerance	Type
3	Minimize P3	P3 - wb_y	Minimize	-6		No Constraint
*		Select a Parameter				

8. In the **Outline** pane, select **Domain**.
9. In the **Properties** pane, set input parameter properties as shown:

Table of Schematic G2: Optimization				
	A	B	C	D
1	⊟ Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - wb_x1	-0.25	1.75	0.75
4	P2 - wb_x2	-2.725	-0.725	-1.725
5	⊟ Parameter Relationships			
6	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	<=	*New Expression*

10. Update the **Optimization** cell.

When the update is complete, the **Table** pane displays 38 design points and their results.

View Results

1. In the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, you can see that 5 iterations and 38 new design points were needed to find the global minimum.

   From the Convergence Criteria chart and the Raw Optimization Data table, you can see that the candidate was already found at the third iteration with 13 design points.

   In the **Table** pane, you can see **Candidate Point 1** is the best candidate. For this candidate point, the objective value of the function is **-6.5511**, which again matches the expected value.

Table of Schematic G2: Optimization

	A	B	C	D	E
1	⊟ Optimization Study				
2	Minimize P3	Goal, Minimize P3 (Default importance)			
3	⊟ Optimization Method				
4	NLPQL	The NLPQL method (Nonlinear Programming by Quadratic Lagrangian) is a gradient-based algorithm to provide a refined, local, optimization result. It supports a single objective, multiple constraints and is limited to continuous parameters. The starting point must be specified to determine the region of the design space to explore.			
5	Configuration	Approximate derivatives by Forward difference and find 3 candidates in a maximum of 20 iterations.			
6	Status	Converged after 38 evaluations.			
7	⊟ Candidate Points				
8		Starting Point	Candidate Point 1	Candidate Point 2	Candidate Point 3
9	P1 - wb_x1	0.75	0.22843	0.4844	0.75
10	P2 - wb_x2	-1.725	-1.6257	-1.4594	-1.725
11	P3 - wb_y	✗✗✗ -4.2983	★★★ -6.5511	★★ -5.7007	✗✗✗ -4.2983

2. Close the **Optimization** cell.

How Effective was the Approach Used for Scenario 3?

Scenario 3 is quite reliable because you began by running a Screening direct optimization to explore the design space and find the best global candidate point, which you then used as the starting point for the NLPQL direct optimization. The best candidate was actually found after 13 design points in the NLPQL, so this approach is more effective than the approaches in the two previous scenarios. It requires only 33 design points, 20 to run the Screening and 13 more for NLPQL.

However, remember that the candidate point found by Screening must be good enough to guarantee convergence of NLPQL. Convergence depends on the space-filling ability of Screening to create enough samples to adequately explore the parameter space. Also, a Screening direct optimization can be expensive when you have a large number of input parameters.

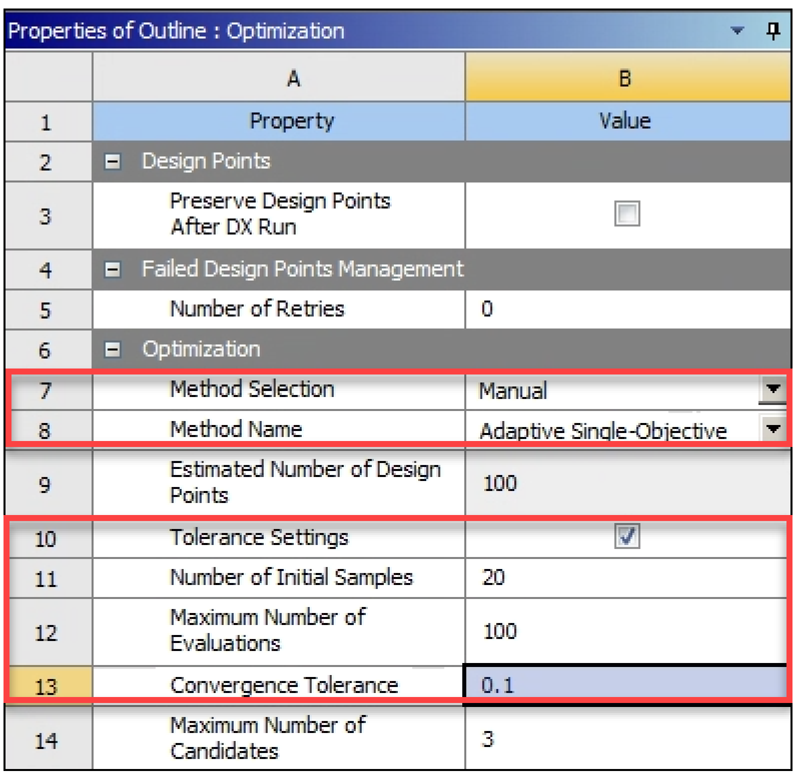
Scenario 4: Adaptive Single-Objective Direct Optimization

In Scenario 4, you will run only a single Adaptive Single-Objective (ASO) direct optimization and then view results.

Run the ASO Direct Optimization System

Add, configure, and update the ASO direct optimization:

1. From under **Design Exploration** in the **Toolbox**, double **Direct Optimization** to add this system to the **Project Schematic**.
2. Rename the system to **`Scenario 4: ASO Direct Optimization`**.
3. In the system, double-click the **Optimization** cell to open it.
4. In the **Outline** pane, select **Optimization**.
5. In the **Properties** pane, set optimization properties as shown:



Properties of Outline : Optimization

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	Adaptive Single-Objective
9	Estimated Number of Design Points	100
10	Tolerance Settings	☑
11	Number of Initial Samples	20
12	Maximum Number of Evaluations	100
13	Convergence Tolerance	0.1
14	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.
7. In the **Table** pane, for **Parameter**, select **P3 - wb_y** and then set the objective type to **Minimize** and **Target** to **-6**:

Table of Schematic H2: Optimization						
	A	B	C	D	E	F
1	Name	Parameter	Objective			
2			Type	Target	Tolerance	Type
3	Minimize P3	P3 - wb_y	Minimize	-6		No Constraint
*		Select a Parameter				

8. In the **Outline** pane, select **Domain**.

9. In the **Properties** pane, set the lower and upper bounds as shown:

Table of Schematic H2: Optimization				
	A	B	C	D
1	Input Parameters			
2	Name	Lower Bound	Upper Bound	
3	P1 - wb_x1	-3	3	
4	P2 - wb_x2	-3	3	
5	Parameter Relationships			
6	Name	Left Expression	Operator	Right Expression

10. Update the **Optimization** cell.

View Results

1. In the **Outline** pane, select **Optimization**.

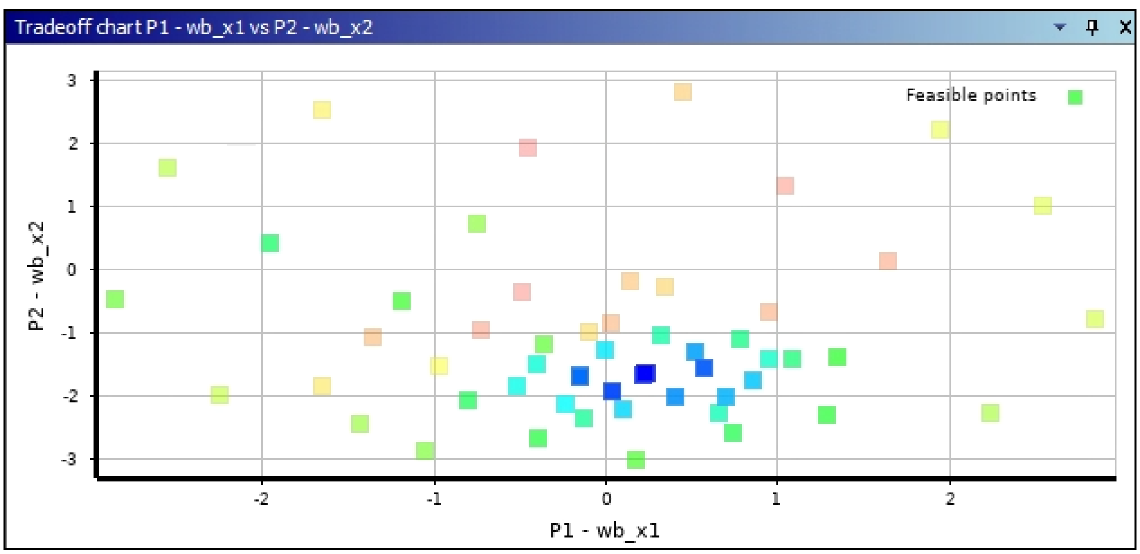
   In the **Properties** pane under **Optimization Status**, you can see that the optimization converged after 56 evaluations and 3 candidates were identified.

   In the **Table** pane, you can see **Candidate Point 1** is the best candidate. For this candidate point, parameter **P1** has a value of **0.23301**, parameter **P2** has a value of **-1.6264**, and the corresponding objective function value for **P3** is **–6.5509**. These values show that the optimization has reached the expected global minimum.

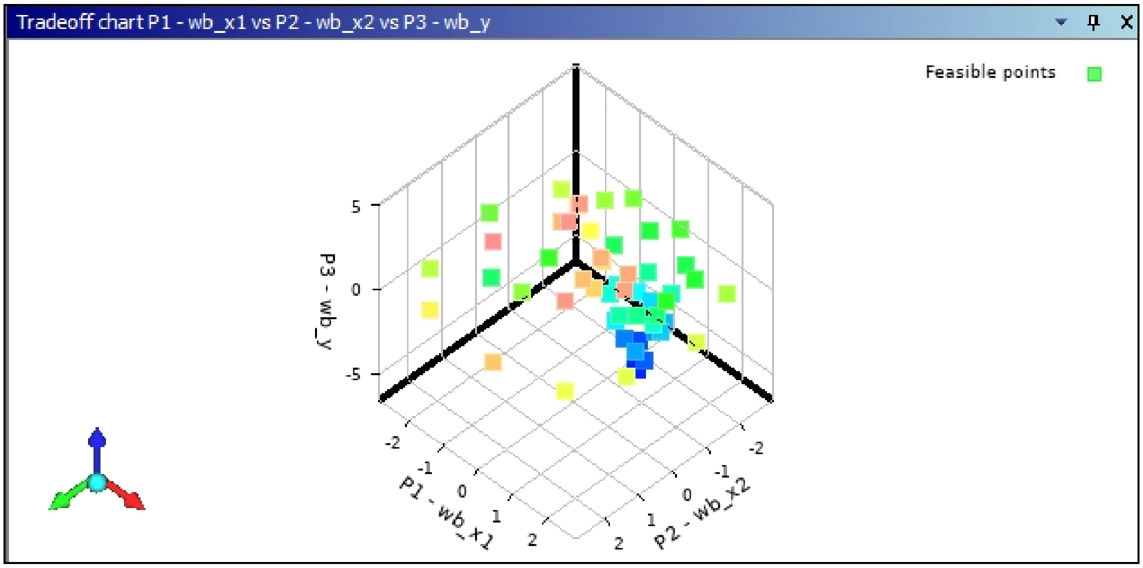
Table of Schematic H2: Optimization				
	A	B	C	D
1	⊟ Optimization Study			
2	Minimize P3	Goal, Minimize P3 (Default importance)		
3	⊟ Optimization Method			
4	Adaptive Single-Objective	The Adaptive Single-Objective method is a gradient-based algorithm to provide a refined, global, optimization result. It supports a single objective, multiple constraints and aims at finding the global optimum . It is limited to continuous and manufacturable input parameters.		
5	Configuration	Find 3 candidates in a maximum of 100 evaluations and 20 domain reductions.		
6	Status	Converged after 56 evaluations.		
7	⊟ Candidate Points			
8		Candidate Point 1	Candidate Point 2	Candidate Point 3
9	P1 - wb_x1	0.23301	0.70008	-1.95
10	P2 - wb_x2	-1.6264	-1.9738	0.45
11	P3 - wb_y	★★★ -6.5509	★★ -3.6408	-1.3161

2. In the **Outline** pane under **Results**, select **Tradeoff**. In the **Chart** pane, you can see that the refinement is targeted to a small area of the surface.

   For the following Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **2D**. Under **Axes**, **X Axis** is set to **P1 - wb_x1** and **Y Axis** is set to **P2 - wb_x2**.

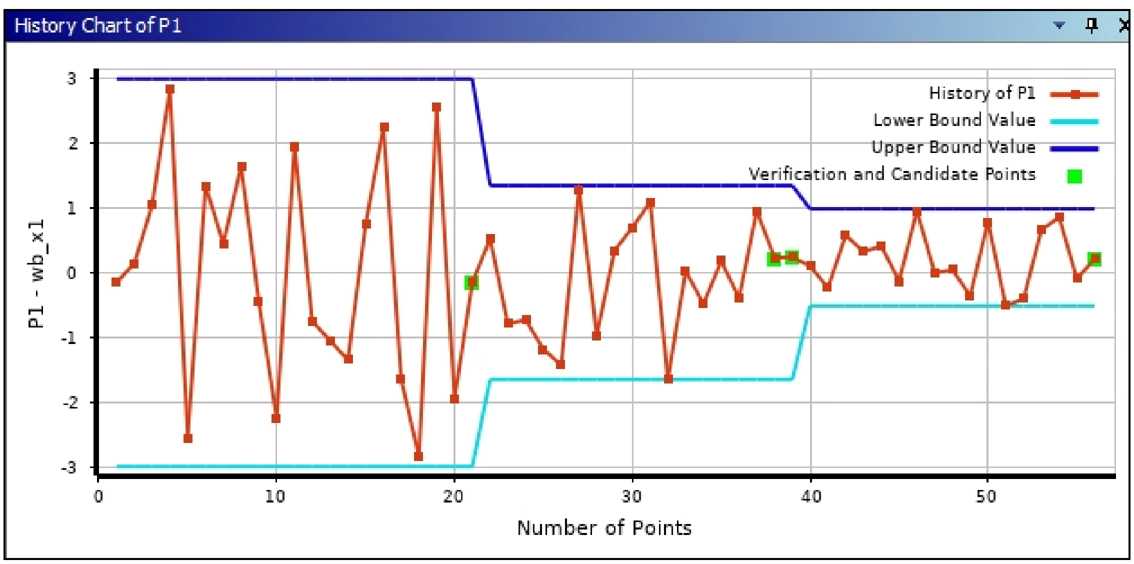


   For the next Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **3D**. Under **Axes**, the **X Axis** and **Y Axis**settings remain the same. The **Z Axis** is set to **P3 - wb_y**. This chart displays the refinement in the minimum area.

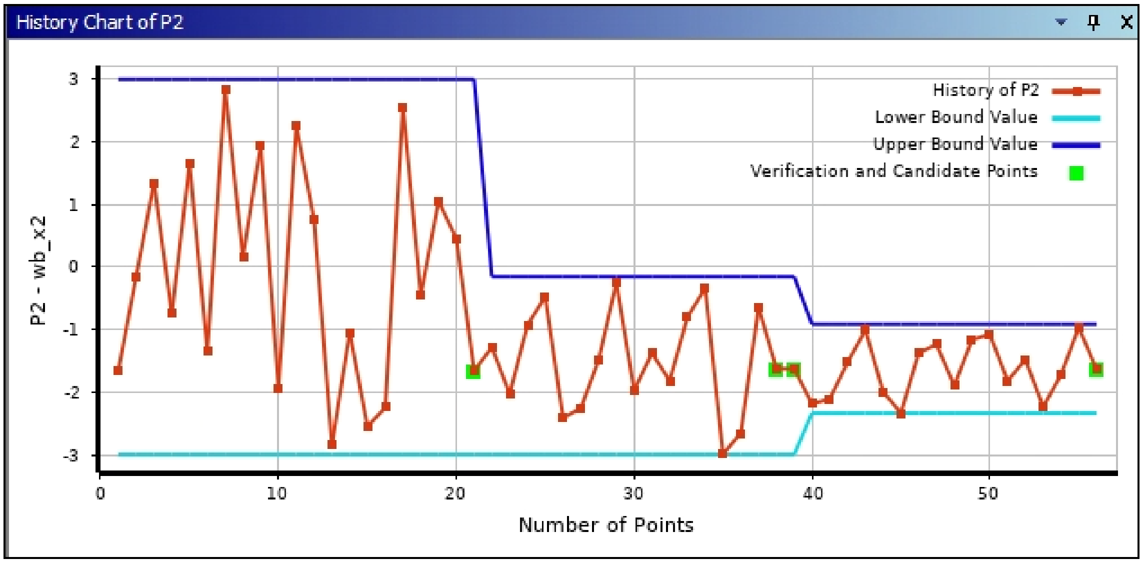


The History charts for the input parameters show the successive steps of the domain reduction performed by ASO.

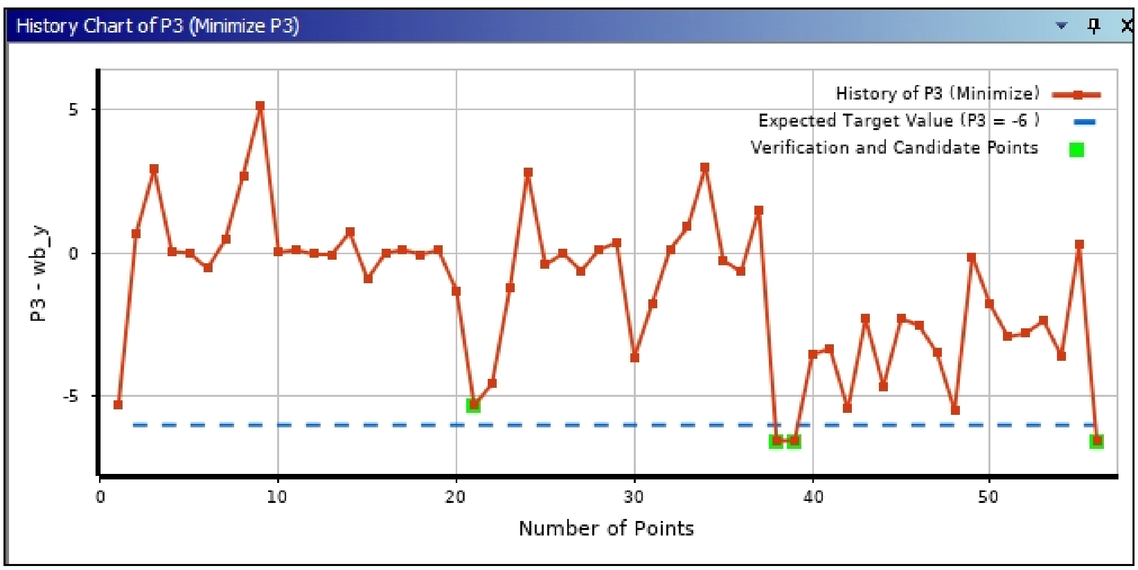
3. In the **Outline** pane under **Domain**, select **P1 - wb_x1** to view the History chart of this input parameter in the **Graph** pane:



4. In the **Outline** pane under **Domain**, select **P2 - wb_x2** to view the History chart of this input parameter in the **Graph** pane:



5. In the **Outline** pane under **Objectives and Constraints**, select **Minimize P3** to view the History chart of this output parameter in the **Graph** pane:



6. Close the **Optimization** cell.

7. Exit Workbench, saving project changes.

How Effective was the Approach Used for Scenario 4?

This scenario required 56 design points. Although ASO needed more points than the methods used in Scenarios 2 and 3, it has the added benefit of ease of use. ASO automatically zooms in on a solution by adaptive methods.

ASO has these advantages:

- Offers a fully automated method of finding the global optimum using sampling, a response surface, and the NLPQL algorithm.

- Employs targeted refinement (the refinement of the internal response surface is driven by the optimization objective), which means that it does not expend time or resources on refining the surface in areas not relevant to the optimization.
- Finds the optimal point without requiring results from a prior optimization.
- Reaches a high level of accuracy early in the optimization process, accelerating the optimization process by enabling you to accept intermediate results.

Time to Spare?

Open the saved **DX_ASO** project and try to find the global maximum of the objective function by using ASO.

What Have You Learned?

During this tutorial, you learned that DesignXplorer offers multiple ways to find the global optimum for a given function.

Different Types of Optimizations

You looked at different types of optimizations: response surface optimizations, direct optimizations, and an approach that combined both of these types.

Response Surface Optimization

- Allows you to select the type of DOE and response surface best suited to your problem.
- Quickly finds interesting areas of the design space.
- Offers candidate verification and is excellent for exploring sensitivities, determination, and responses.
- Excellent for design exploration and finding an approximated optimum quickly.
- Prevents targeted refinement.

  The response surface is built before optimization objectives are defined. This means that time and resources are spent on parts of the design space that are not relevant to the optimization, making it a less effective approach for optimization. Because the entire design space is refined, the optimization can be very expensive, requiring a large number of design point updates to obtain a response surface that is accurate all over.
- Sacrifices some accuracy to achieve greater efficiency.

  For very complex response surfaces, the number of trials to generate a high enough quality response surface can exceed the number that would have been required for a direct solve.

Direct Optimization

- Gives you the ability to reuse data.

Although it does not have a single associated response surface, data reuse is possible because it can retrieve information via data links from other DesignXplorer system cells that contain design point data.

- Good choice when a large number of parameters or problems with building a good response surface make response surface optimization infeasible.
- Provides targeted refinement.

  Refinement is driven by the objective, with the creation of a new response surface on a smaller domain with each iteration. The smaller the domain, the easier the surface construction and the more accurate the approximation.

- Although it uses real solves, each design point update is worth the expense.

  Each update is targeted on the area most relevant to the optimization, allowing the refinement process to progressively zoom in on the optimum.

Different Optimization Methods

Screening

- Good option for the initial exploration of a design space because its space-filling abilities allow it to locate a viable candidate point, which can possibly be used as a starting point for an NLPQL optimization.
- Can be expensive when there are many input parameters.

NLPQL

- Can add accuracy to the response surface-based approach.
- Highly dependent on the quality of the starting point.

ASO

- Adaptive method that uses actual solves to refine the response surface, but it uses approximation when accuracy is good enough.
- Combines the best of DesignXplorer technologies: a DOE, an internal response surface, domain reduction, and error prediction.
- Provides both accuracy and speed without needing prior results to initialize the optimization.
- Allows you to balance your available time and resources with your desired level of accuracy.
- While a response surface optimization or NLPQL might be sufficient for exploring problems that are convex or smooth, the ASO is a better optimization choice when you are not already very familiar with your problem.

Using Adaptive Multiple-Objective Optimization

In this advanced tutorial, you'll use DesignXplorer's *Adaptive Multiple-Objective (AMO)* optimization method as part of a direct optimization to locate a Pareto front. First, you'll create two separate projects to examine different functions and apply both the Multiple-Objective Genetic Algorithm (MOGA) and AMO to the problem. Then, you'll review the results and examine why AMO is better suited to finding the Pareto front for the given problem.

Note:

This advanced tutorial assumes that you are familiar with Ansys Workbench and with DesignXplorer's goal-driven optimization functionality. For an introduction to such optimizations, see the tutorial Performing a Goal-Driven Optimization (p. 79).

This tutorial is divided into the following sections:

What is Adaptive Multiple-Objective Optimization?
Problem 1
Problem 2
What Have You Learned?

What is Adaptive Multiple-Objective Optimization?

Adaptive Multiple-Objective (AMO) is a hybrid optimization method that combines a Kriging response surface optimization and MOGA direct optimization. It uses the same general approach as MOGA but extends it by using the Kriging error predictor to reduce the number of evaluations needed to locate the global optimum.

Problem 1

This tutorial covers the following topics for Problem 1:

Problem 1 Definition
Problem 1 Project Setup
Run the MOGA Direct Optimization
Run the AMO Direct Optimization

Problem 1 Definition

The analytic function you'll examine for Problem 1 has two input parameters and four output parameters (with two objectives and two constraints). The definition of the problem is as follows:

Minimize

$$f_1(x_1,x_2) = (x_1 + x_2 - 7.5)^2 + (x_2 - x_1 + 3)^2 / 4$$

$$f_2(x_1,x_2) = (x_1 - 1)^2 / 4 + 2$$

With

$$g_1(x_1,x_2) = (x_1 - 2)^3 / 2 + x_2 - 2.5 \leq 0$$

$$g_2(x_1,x_2) = x_1 + x_2 - 8(x_2 - x_1 + 0.65)^2 - 3.85 \leq 0$$

Where $0 \leq x_1 \leq 5$ and $0 \leq x_2 \leq 3$.

Problem 1 Project Setup

1. Start Workbench.
2. From under **Component Systems** in the **Toolbox**, double-click **Microsoft Office Excel** to add this system to the **Project Schematic**.
3. Right-click the **Analysis** cell and select **Add File** → **Browse**
4. In the file browser, locate and open the file `DX_AMO_input1.xlsx`, which is in the directory to which you extracted the input files (p. 27).
5. On the **Project Schematic**, double-click the **Analysis** cell to open it.

Note:

The order in which you define parameters determines the system-generated names for the parameters. For this tutorial, you are using a specific order so that the parameter names in your project will match the parameter names in the tutorial.

6. In the **Outline** pane, define parameters in the following order:
   - In the **Input** column, select the check boxes for **x1_** and **x2_**, which appear at the bottom of the list.
   - In the **Output** column, select the check boxes for **f1_**, **f2_**, **g1_**, and **g2_**.

   The **Outline** pane will look like this.

Outline of Schematic A2: Analysis			
	A	B	C
1		Input	Output
2	⊟ Setup		
3	⊟ DX_AMO_input1.xlsx		
4	f1_	☐	☑
5	f2_	☐	☑
6	g1_	☐	☑
7	g2_	☐	☑
8	x1_	☑	☐
9	x2_	☑	☐

7. Close the **Analysis** cell. The **Project Schematic** now includes the **Parameter Set** bar.

8. Double-click the **Parameter Set** bar to open it. Given that you added the parameters in the order indicated, the **Outline** pane looks like this:

Outline of Schematic A3: Parameters				
	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	⊟ Input Parameters			
3	⊟ Microsoft Office Excel (A1)			
4	P1	x1_	0.5	
5	P2	x2_	0.5	
*	New input parameter	New name	New expression	
7	⊟ Output Parameters			
8	⊟ Microsoft Office Excel (A1)			
9	P3	f1_	⚡	
10	P4	f2_	⚡	
11	P5	g1_	⚡	
12	P6	g2_	⚡	
*	New output parameter		New expression	
14	Charts			

9. Close the **Parameter Set** bar.

10. Update the project.

11. Save the project as `DX_AMO_Problem1.wbpj` in either this same directory or another directory.

In this project, you'll add, configure, and update two **Direct Optimization** systems. One will use MOGA, and the other will use AMO.

Run the MOGA Direct Optimization

First, add, configure, and update the MOGA **Direct Optimization** system.

1. From under **Design Exploration** in the **Toolbox**, double **Direct Optimization** to add this system to the **Project Schematic**.

2. Rename the system to **`MOGA Direct Optimization`**.

3. In the system, double-click the **Optimization** cell to open it.

4. In the **Outline** pane, select **Optimization**.

5. In the **Properties** pane, set optimization properties as shown:

Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	MOGA
9	Estimated Number of Design Points	1050
10	Tolerance Settings	☑
11	Number of Initial Samples	100
12	Number of Samples Per Iteration	50
13	Maximum Allowable Pareto Percentage	70
14	Convergence Stability Percentage	2
15	Maximum Number of Iterations	20
16	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.

7. In the **Table** pane, specify objectives and constraints as shown:

Table of Schematic C2: Optimization

		B	C	D	E	F	G	H	I
1		Parameter	Objective			Constraint			
2			Type	Target	Tolerance	Type	Lower Bound	Upper Bound	Tolerance
3	P3	P3 - f1_	Minimize	0		No Constraint			
4	P4	P4 - f2_	Minimize	0		No Constraint			
5		P5 - g1_	No Objective			Values <= Upper Bound		0	0.001
6		P6 - g2_	No Objective			Values <= Upper Bound		0	0.001

8. In the **Outline** pane, select **Domain**.

9. In the **Table** pane, set the lower and upper bounds as shown:

Table of Schematic B2: Optimization				
	A	B	C	D
1	⊟ Input Parameters			
2	Name	Lower Bound	Upper Bound	
3	P1 - x1_	0	5	
4	P2 - x2_	0	3	

10. Update the **Optimization** cell.

The update may take a few minutes. While the update is in process, you can look at the History charts to monitor the progress of the input parameters and the objectives and constraints. In the **Outline** pane under **Objectives and Constraints** and **Domain**, a sparkline image of the History chart is displayed for each object in the **Monitoring** column. To view the full-sized History chart in the **Graph** pane, select the object in the **Outline** pane.

11. When the update is complete, in the **Outline** pane, select **Optimization**.

In the **Properties** pane under **Optimization Status**, you can see that **Converged** is set to **Yes**. Eight iterations and 374 design points were needed for convergence.

Note:

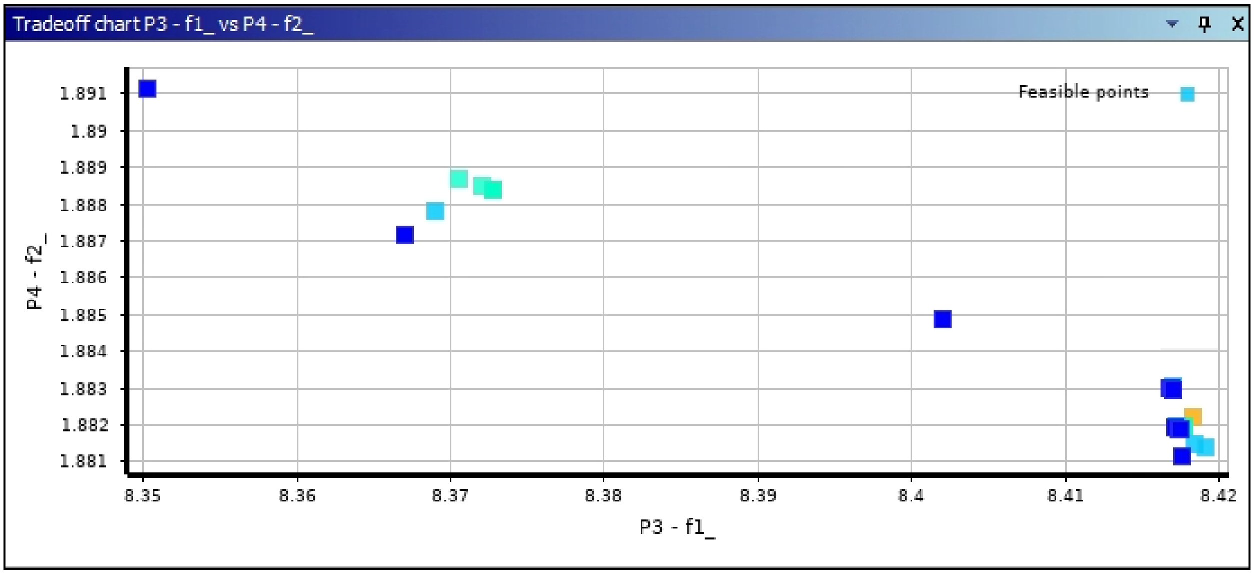
You can confirm this in the **Table** pane under **Optimization Method**, where **Status** displays **`Converged after 374 evaluations`**.

---

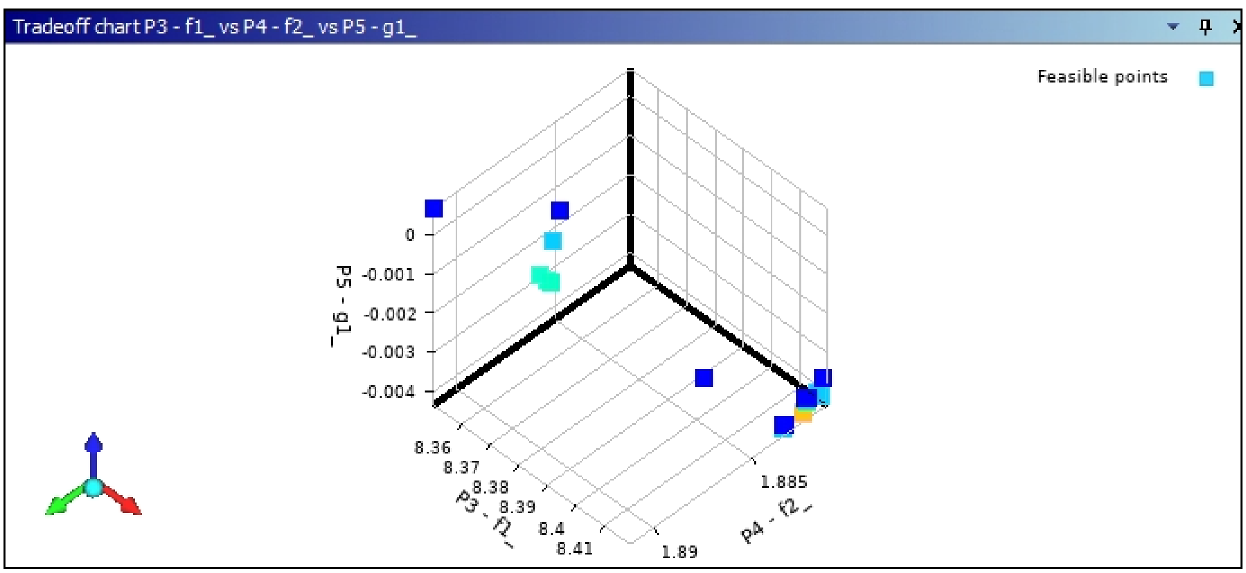
12. In the **Outline** pane under **Results**, select **Tradeoff**.

In the **Graph** pane, you can see the Tradeoff chart, which shows the Pareto front found by the optimization in blue by default.

For the following Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **2D**. Under **Axes**, **X Axis** is set to **P3 - f1_** and **Y Axis** is set to **P4 - f2_** .



For the next Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **3D**. Under **Axes**, **Z Axis** is set to **P5 - g1_** .



13. Close the **Optimization** cell.

Run the AMO Direct Optimization

Next, add, configure, and update the AMO **Direct Optimization** system.

1. From under **Design Exploration** in the **Toolbox**, double-click **Direct Optimization** to add this system to the **Project Schematic**.
2. Rename the system to **`AMO Direct Optimization`**.
3. In the system, double-click the **Optimization** cell to open it.
4. In the **Outline** pane, select **Optimization**.

5. In the **Properties** pane, set optimization properties as shown:

Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	Adaptive Multiple-Objective
9	Estimated Number of Design Points	825
10	Tolerance Settings	☑
11	Number of Initial Samples	100
12	Number of Samples Per Iteration	50
13	Maximum Allowable Pareto Percentage	70
14	Convergence Stability Percentage	2
15	Maximum Number of Iterations	20
16	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.

7. In the **Table** pane, specify the same objectives and constraints as you did for the MOGA optimization:

Table of Schematic C2: Optimization

		B	C	D	E	F	G	H
1		Parameter	Objective			Constraint		
2			Type	Target	Tolerance	Type	Lower Bound	Upper Bound
3	P3	P3 - f1_	Minimize	0		No Constraint		
4	P4	P4 - f2_	Minimize	0		No Constraint		
5		P5 - g1_	No Objective			Values <= Upper Bound		0
6		P6 - g2_	No Objective			Values <= Upper Bound		0

8. In the **Outline** pane, select **Domain**.

9. In the **Table** pane, set the lower and upper bounds the same as you did for the MOGA optimization:

Table of Schematic B2: Optimization

	A	B	C	D
1	Input Parameters			
2	Name	Lower Bound	Upper Bound	
3	P1 - x1_	0	5	
4	P2 - x2_	0	3	

10. Update the **Optimization** cell.

    While the update is in process, you can look at the History chart to monitor the progress of the input parameters and the objectives and constraints as before. The AMO optimization runs much more quickly than the previous MOGA optimization.

11. When the update is complete, in the **Outline** pane, select **Optimization**.

12. In the **Properties** pane under **Optimization Status**, **Converged** is set to **Yes**. You can see that 8 iterations and 131 design points were needed for convergence.

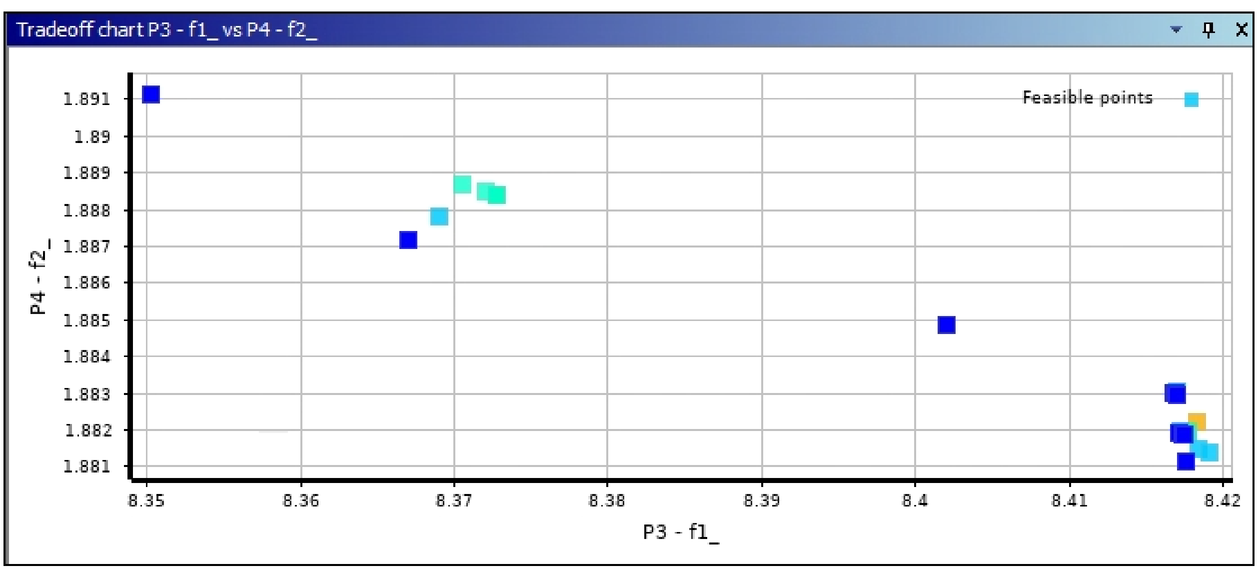
    Note:

    You can confirm this in the **Table** pane under **Optimization Method**, where **Status** displays `Converged after 131 evaluations`.

13. In the **Outline** pane under **Results**, select **Tradeoff**.

    As before, in the **Graph** pane you can see the Tradeoff chart, which shows the Pareto front found by the optimization in blue by default.

    For the following Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **2D**. Under **Axes**, **X Axis** is set to **P3 - f1_** and **Y Axis** is set to **P4 - f2_** .



The Pareto front found by the AMO optimization is similar to the one found by the MOGA optimization. While the results for the MOGA and AMO methods are similar, the number of design points needed to reach these results differ greatly. While MOGA had to run 374 design points, AMO had to run only 131 design points. Because AMO required 243 fewer points, the optimization process was much faster.

Note:

In this particular example, the Kriging response surface had excellent error prediction, so the optimization results are similar to those found by MOGA. This is not always the case. The quality of the Kriging error prediction can vary according to the type of function.

---

14. Close the **Optimization** cell.

15. Exit Workbench, saving your changes to this project.

Problem 2

This tutorial covers the following topics for Problem 2:

Problem 2 Definition
Problem 2 Project Setup
Run the MOGA Direct Optimization
Run the AMO Direct Optimization

Problem 2 Definition

The analytic function you'll examine for this problem has six input parameters, two objectives, and six constraints. The definition of the problem is as follows:

Minimize

$$f_1(x) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2$$

$$f_2(x) = \sum_{i=1}^{n} x_i^2$$

With

$$g_1(x) = 2 - (x_1 + x_2) \leq 0$$

$$g_2(x) = (x_1 + x_2) - 6 \leq 0$$

$$g_3(x) = 2 - (x_2 - x_1) - 2 \leq 0$$

$$g_4(x) = (x_1 - 3x_2) - 2 \leq 0$$

$$g_5(x) = (x_3 - 3)^2 + x^4 - 4 \leq 0$$

$$g_6(x) = 4 - \left((x_5 - 3)^2 + x_6\right) \leq 0$$

Where $0 \leq x_1, x_2, x_6 \leq 10$, $1 \leq x_3, x_5 \leq 3$, and $0 \leq x_4 \leq 6$

Problem 2 Project Setup

1. Start Workbench.
2. From under **Component Systems** in the **Toolbox**, double-click **Microsoft Office Excel** to add the system to the **Project Schematic**.
3. Right-click the **Analysis** cell and select **Add File** → **Browse**
4. In the file browser that opens, locate and open the file `DX_AMO_input2.xlsx`, which is in the directory to which you extracted the input files (p. 27).
5. On the **Project Schematic**, double-click the **Analysis cell** to open it.

Note:

The order in which you define parameters determines the system-generated names for the parameters. For this tutorial, you are using a specific order so that the parameter names in your project will match the parameter names in the tutorial.

6. In the **Analysis** cell, define inputs and outputs in the following order:
   - In the **Input** column, select the check boxes for **x1_** through **x6_**, which appear at the bottom of the list.
   - In the **Output** column, select the check boxes for **f1_** through **f8_**.

   The **Outline** pane will look like this.

Outline of Schematic A2: Analysis			
	A	B	C
1		Input	Output
2	⊟ Setup		
3	⊟ ✓ DX_AMO_input2.xlsx		
4	f1_	☐	☑
5	f2_	☐	☑
6	f3_	☐	☑
7	f4_	☐	☑
8	f5_	☐	☑
9	f6_	☐	☑
10	f7_	☐	☑
11	f8_	☐	☑
12	x1_	☑	☐
13	x2_	☑	☐
14	x3_	☑	☐
15	x4_	☑	☐
16	x5_	☑	☐
17	x6_	☑	☐

7. Close the **Analysis** cell. The **Project Schematic** now includes the **Parameter Set** bar.

8. Double-click the **Parameter Set** bar to open it. Given that you added the parameters in the order indicated, the **Outline** pane looks like this:

Outline of All Parameters				
	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	Input Parameters			
3	Microsoft Office Excel (A1)			
4	P1	x1_	3	
5	P2	x2_	3	
6	P3	x3_	3	
7	P4	x4_	3	
8	P5	x5_	3	
9	P6	x6_	3	
*	New input parameter	New name	New expression	
11	Output Parameters			
12	Microsoft Office Excel (A1)			
13	P7	f1_		
14	P8	f2_		
15	P9	f3_		
16	P10	f4_		
17	P11	f5_		
18	P12	f6_		
19	P13	f7_		
20	P14	f8_		
*	New output parameter		New expression	
22	Charts			

9. Close the **Parameter Set** bar.

10. Update the project.

11. Save the project as `DX_AMO_Problem2.wbpj` in either this same directory or another directory.

In this project, you'll add, configure, and update two **Direct Optimization** systems. Again, one will use MOGA, and the other will use AMO.

Run the MOGA Direct Optimization

First, you'll add, configure, and update the MOGA **Direct Optimization** system.

1. From under **Design Exploration** in the **Toolbox**, double-click **Direct Optimization** to add this system to the **Project Schematic**.

2. Rename the system to **`MOGA Direct Optimization`**.

3. In the system, double-click the **Optimization** cell to open it.

4. In the **Outline** pane, select **Optimization**

5. In the **Properties** pane, set optimization properties as shown:

Properties of Outline : Optimization		
	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	MOGA
9	Estimated Number of Design Points	1050
10	Tolerance Settings	☑
11	Number of Initial Samples	100
12	Number of Samples Per Iteration	50
13	Maximum Allowable Pareto Percentage	70
14	Convergence Stability Percentage	2
15	Maximum Number of Iterations	20
16	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.

7. In the **Table** pane, set objectives and constraints as shown:

Table of Schematic B2: Optimization								
	A	B	C	D	E	F	G	
1	Name	Parameter	Objective			Constraint		
2			Type	Target	Tolerance	Type	Lower Bound	Up
3	Minimize P7	P7 - f1_	Minimize	-100		No Constraint		
4	Minimize P8	P8 - f2_	Minimize	30		No Constraint		
5	P9 <= 0	P9 - f3_	No Objective			Values <= Upper Bound		0
6	P10 <= 0	P10 - f4_	No Objective			Values <= Upper Bound		0
7	P11 <= 0	P11 - f5_	No Objective			Values <= Upper Bound		0
8	P12 <= 0	P12 - f6_	No Objective			Values <= Upper Bound		0
9	P13 <= 0	P13 - f7_	No Objective			Values <= Upper Bound		0
10	P14 <= 0	P14 - f8_	No Objective			Values <= Upper Bound		0
*		Select a Parameter						

8. In the **Outline** pane, select **Domain**.

9. In the **Table** pane, set lower and upper bounds as shown:

Table of Schematic B2: Optimization			
	A	B	C
1	Input Parameters		
2	Name	Lower Bound	Upper Bound
3	P1 - x1_	0	10
4	P2 - x2_	0	10
5	P3 - x3_	1	5
6	P4 - x4_	0	6
7	P5 - x5_	1	6
8	P6 - x6_	0	10

10. Update the **Optimization** cell.

    The update takes several minutes. As before, you can watch the optimization process, either on the sparkline image in the **Outline** pane or on the full History chart in the **Graph** pane.

11. When the update is complete, in the **Outline** pane, select **Optimization**.

    In the **Properties** pane under **Optimization Status**, **Converged** is set to **Yes**. You can see that 10 iterations and 535 design points were needed for convergence.

    Note:

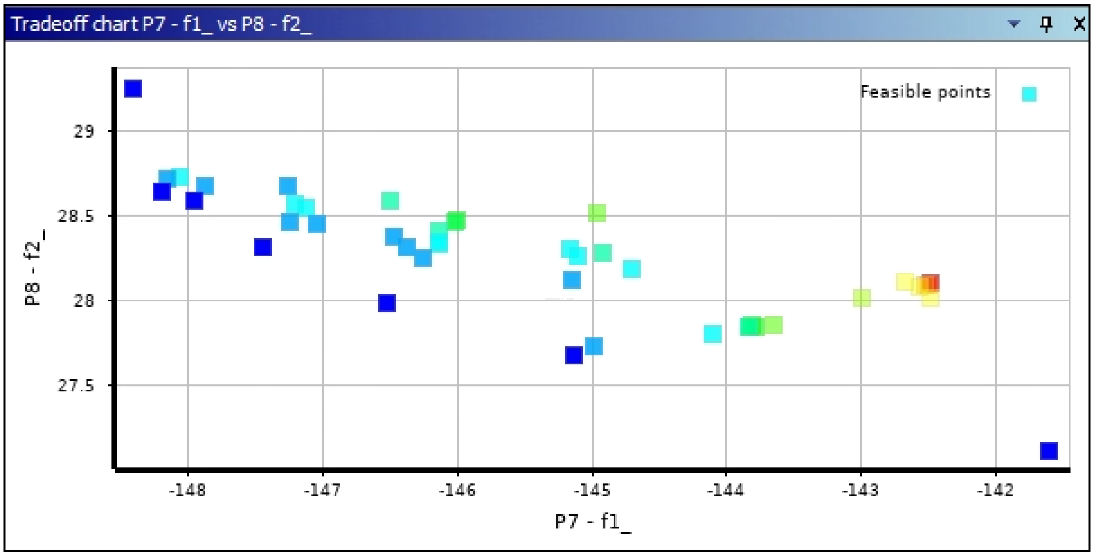
    You can confirm this in the **Table** pane under **Optimization Method**, where **Status** displays `Converged after 535 evaluations`.

    ---

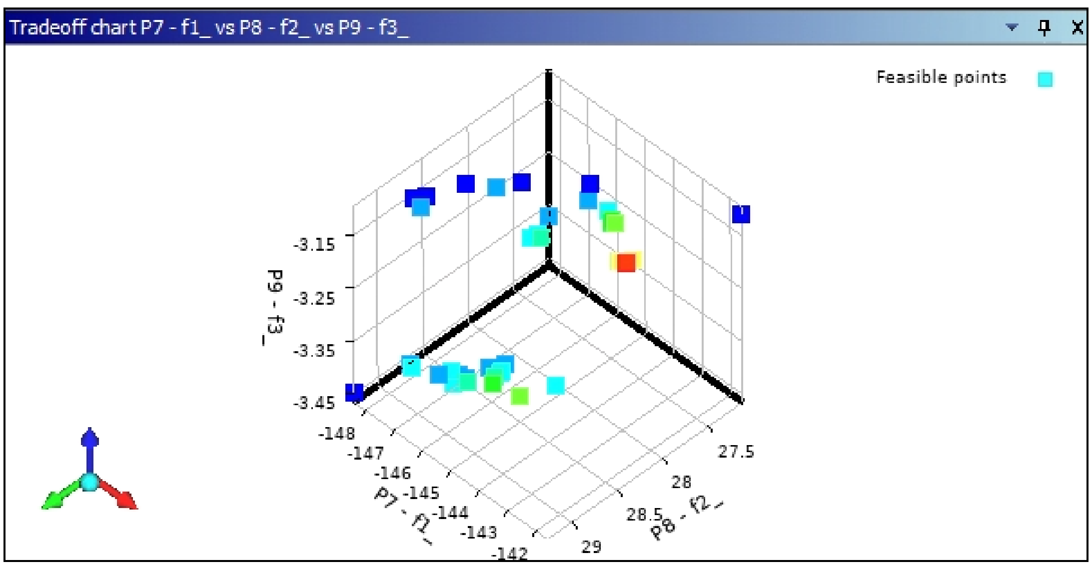
12. In the **Outline** pane under **Results**, select **Tradeoff**.

    As before, in the **Graph** pane, you can see the Tradeoff chart, which shows the Pareto front found by the optimization in blue by default.

    For the following Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **2D**. Under **Axes**, **X Axis** is set to **P7 - f1_** and **Y Axis** is set to **P8 - f2_** .



For the next Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **3D**. Under **Axes**, the **X Axis** and **Y Axis** settings remain the same. **Z Axis** is set to **P9 - f3_**.



13. Close the **Optimization** cell.

Run the AMO Direct Optimization

Next, you'll add, configure, and update the AMO **Direct Optimization** system.

1. Under **Design Exploration** in the **Toolbox**, double-click **Direct Optimization** to add the second system to the **Project Schematic**.
2. Rename the system to `AMO Direct Optimization`.
3. In the system, double-click the **Optimization** cell to open it.
4. In the **Outline** pane, select **Optimization**.

5. In the **Properties** pane, set optimization properties as shown:

Properties of Outline A2: Optimization

	A	B
1	Property	Value
2	⊟ Design Points	
3	Preserve Design Points After DX Run	☐
4	⊟ Failed Design Points Management	
5	Number of Retries	0
6	⊟ Optimization	
7	Method Selection	Manual
8	Method Name	Adaptive Multiple-Objective
9	Estimated Number of Design Points	825
10	Tolerance Settings	☑
11	Number of Initial Samples	100
12	Number of Samples Per Iteration	50
13	Maximum Allowable Pareto Percentage	70
14	Convergence Stability Percentage	2
15	Maximum Number of Iterations	20
16	Maximum Number of Candidates	3

6. In the **Outline** pane, select **Objectives and Constraints**.

7. In the **Table** pane, set the same objectives and constraints as you did for the MOGA optimization:

Table of Schematic C2: Optimization

	A	B	C	D	E	F	G	
1	Name	Parameter	Objective			Constraint		
2			Type	Target	Tolerance	Type	Lower Bound	Up
3	Minimize P7	P7 - f1_	Minimize	-100		No Constraint		
4	Minimize P8	P8 - f2_	Minimize	30		No Constraint		
5	P9 <= 0	P9 - f3_	No Objective			Values <= Upper Bound		0
6	P10 <= 0	P10 - f4_	No Objective			Values <= Upper Bound		0
7	P11 <= 0	P11 - f5_	No Objective			Values <= Upper Bound		0
8	P12 <= 0	P12 - f6_	No Objective			Values <= Upper Bound		0
9	P13 <= 0	P13 - f7_	No Objective			Values <= Upper Bound		0
10	P14 <= 0	P14 - f8_	No Objective			Values <= Upper Bound		0
*		Select a Parameter						

8. In the **Outline** pane, select **Domain**.

9. In the **Table** pane, set the lower and upper bounds as you did for the MOGA optimization:

Table of Schematic C2: Optimization

	A	B	C
1	Optimization Domain	Lower Bound	Upper Bound
2	P1 - x1_	0	10
3	P2 - x2_	0	10
4	P3 - x3_	1	5
5	P4 - x4_	0	6
6	P5 - x5_	1	5
7	P6 - x6_	0	10

10. Update the **Optimization** cell.

    As before, you can watch the optimization process, either on the sparkline image in the **Outline** pane or on the full History chart in the **Graph** pane. The AMO optimization runs much more quickly than the previous MOGA optimization.

11. When the update is complete, in the **Outline** pane, select **Optimization**.

    In the **Properties** pane under **Optimization Status**, **Converged** is set to **Yes**. You can see that 10 iterations and 144 design points were needed for convergence.

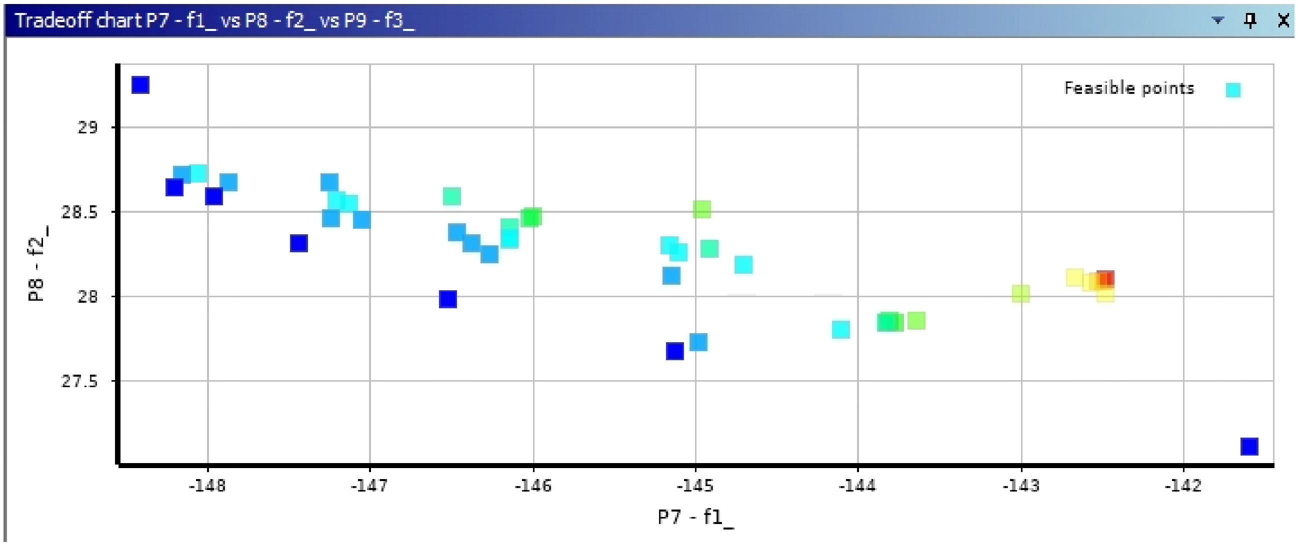
    Note:

    You can confirm this in the **Table** pane under **Optimization Method**, where **Status** displays **`Converged after 144 evaluations.`**.

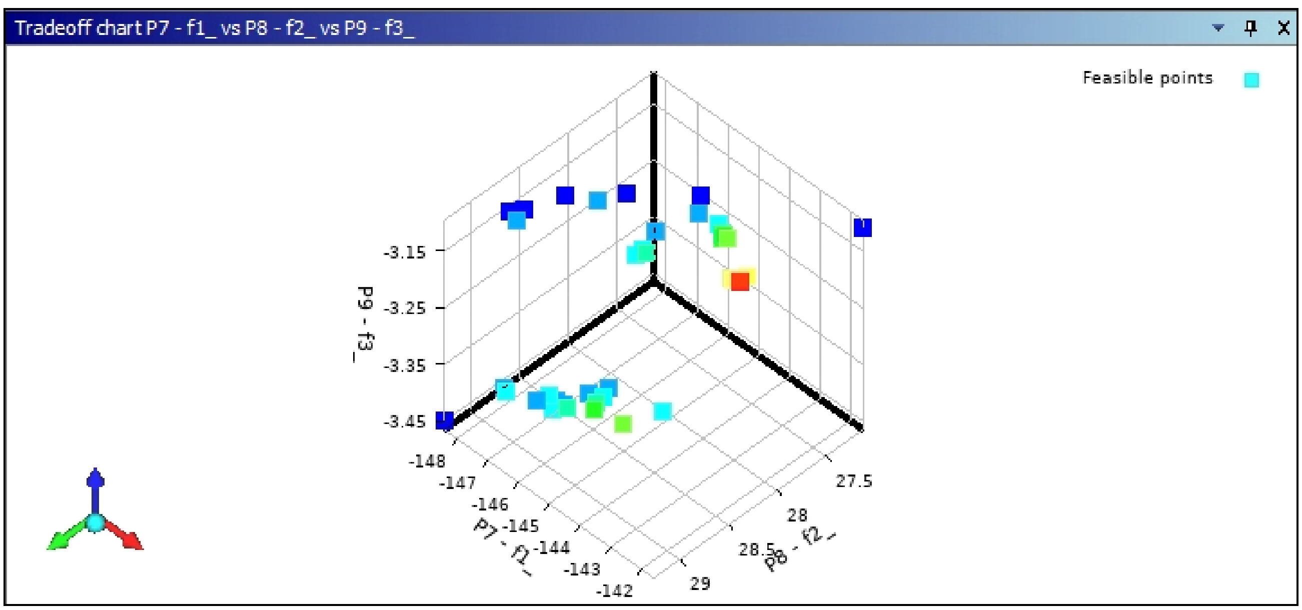
12. In the **Outline** pane under **Results**, select **Tradeoff**.

    As before, in the **Graph** pane, you can see the Tradeoff chart, which shows the Pareto front found by the optimization in blue by default.

    For the following Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **2D**. Under **Axes**, **X Axis** is set to **P7 - f1_** and **Y Axis** is set to **P8 - f2_**.



For the next Tradeoff chart, in the **Properties** pane under **Chart**, **Mode** is set to **3D**. Under **Axes**, the **X Axis** and **Y Axis** settings remain the same. **Z Axis** is set to **P9 - f3_**.



The AMO optimization found a Pareto front that is approximately the same as the one found by the MOGA optimization.

The number of design points needed to reach this result is especially significant. While MOGA had to run 535 design points, AMO had to run only 144. Because AMO required 391 fewer points, the optimization process was much faster.

13. Close the **Optimization** cell.

14. Exit Workbench, saving project changes.

What Have You Learned?

Both MOGA and AMO are designed to perform an optimization with multiple objectives and constraints. You used two separate problems to compare the ability of each method to find a Pareto front. You then reviewed Tradeoff charts to compare the Pareto fronts found by the two methods. As you worked through both problems, you observed that:

- MOGA obtains a high degree of accuracy, but it generally requires more design points to find the Pareto front, which increases the time.
- AMO obtains a high degree of accuracy with fewer design points. However, the accuracy depends on how well the Kriging error predictor is able to simulate the function being examined.
  - With some functions (as in Problems 1 and 2), results are extremely accurate. The AMO optimization finds the exact same Pareto front as the MOGA optimization, but the AMO optimization uses far fewer design points.
  - With other functions, the AMO optimization might find a Pareto front that *approximates* the one found by the MOGA optimization, but the AMO optimization uses far fewer design points.

During this tutorial, you have learned that AMO offers the accuracy of MOGA. AMO also applies the Kriging error predictor to reduce the number of evaluations. In general, when time and resources are considerations, AMO is the better choice for finding an accurate Pareto front.

Using Mixed-Integer Sequential Quadratic Programming (MISQP) Optimization

This advanced tutorial introduces you to the *Mixed-Integer Sequential Quadratic Programming (MISQP)* optimization algorithm and shows how it supports all three types of input parameters: continuous, continuous with manufacturable values, and discrete. To illustrate the effectiveness of MISQP, this tutorial has you apply several different optimization methods to the same problem.

Note:

This tutorial assumes that you are familiar with Ansys Workbench and DesignXplorer's goal-driven optimization functionality. For more information on goal-driven optimization, see the tutorial Performing a Goal-Driven Optimization (p. 79).

This tutorial is divided into the following sections:

Getting Started
Direct Optimization Using Screening
Response Surface Optimization to Direct Optimization Using Screening and MISQP
Direct Optimization Using MISQP
What Have You Learned?

Getting Started

Before beginning the tutorial, you should review the following topics:

What is MISQP?
Problem Definition
Project Setup

What is MISQP?

Adaptive Multiple-Objective (AMO) is a hybrid optimization method that combines a Kriging response surface and the MOGA algorithm in a **Direct Optimization** system. It uses the same general approach as MOGA but extends it by using the Kriging error predictor to reduce the number of evaluations needed to locate the global optimum.

MISQP is a multiple-objective optimization algorithm that is available for both **Direct Optimization** and **Response Surface Optimization** systems. It is used to solve mixed-integer nonlinear programming by modified *Sequential Quadratic programming (SQP)*, with the assumption that integer variables have a smooth influence on the model functions. MISQP has the same settings as NLPQL, but unlike NLPQL, it supports discrete parameters and continuous parameters with manufacturable values.

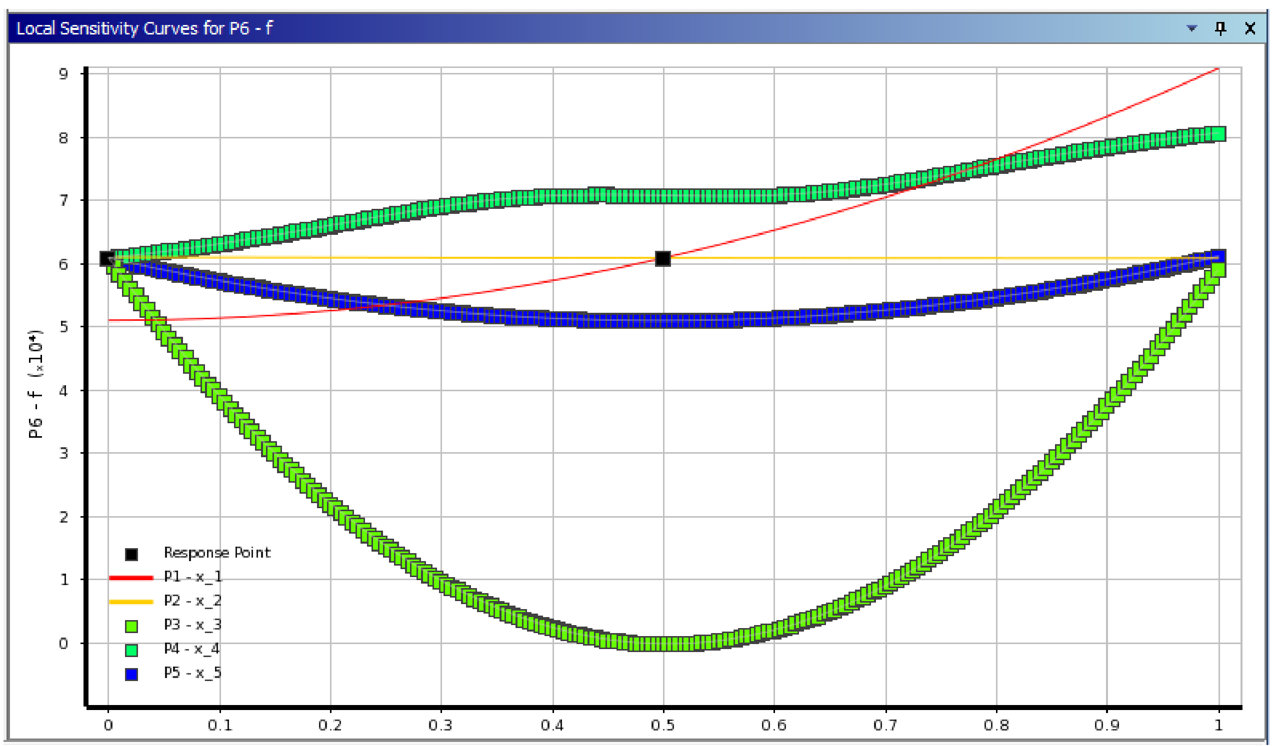
Problem Definition

MISQP can be used to drive any Workbench project containing any combination of solvers. Because almost everyone has access to Microsoft Office Excel, this tutorial uses an analytic function in Excel. The optimization problem has five input parameters: two continuous and three continuous with manufacturable values:

$$\text{Minimize } f(x,y)$$
$$g(x,y) \geq 0$$
$$x \in \mathbb{R}^2, y \in \mathbb{N}^3$$
$$-100 \leq x_i \leq 100, i = 1,2$$
$$-100 \leq y_i \leq 100, i = 1,2,3$$
$$f(x,y) = 5(y_1 - 1)^2 + 0.01y_2^2 + (y_3 + 0.7)^2 + (x_1 - y_1 - 0.33)^2 - x_1 + 0.22$$
$$g(x,y) = (x_1 + 1.25)^2 + 0.5y_3$$

The global minimum is at point **(1.33;100;1;-100;-1)** with a corresponding objective function value of **-10099.69** and a constraint function value of **6.1564**.

Below, the Local Sensitivity Curves chart for the function shows the impact of the five input parameters on the output parameter **P6 - f**.



Project Setup

1. Start Workbench.

2. Select **File** → **Open**.

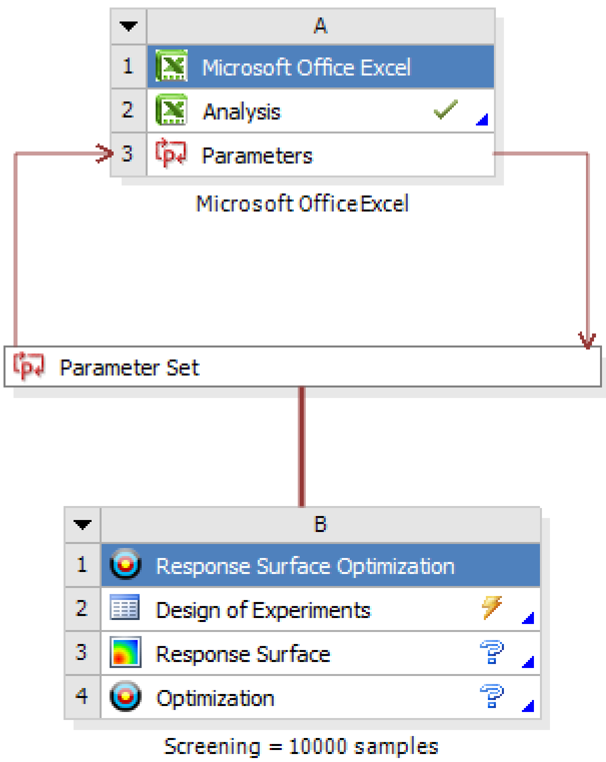
3. In the file browser, locate and open the archived Workbench project file `DX_MISQP.wbpz`, which is in the directory to which you extracted the input files (p. 27).

   A dialog box appears for saving this archived file to a standard Workbench project file.

4. Save the project as `DX_MISQP.wbpj` in either this same directory or another directory.

Direct Optimization Using Screening

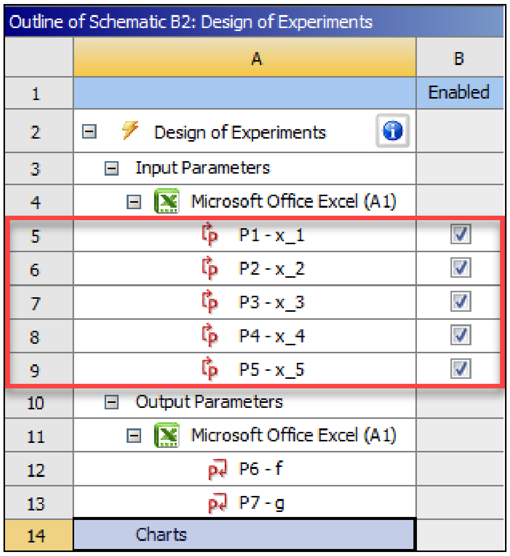
In this project, the **Project Schematic** already includes two systems: **Microsoft Office Excel** and **Response Surface Optimization**.



You will use the **Response Surface Optimization** system to explore the parameters that have already been defined and are used in both systems.

1. In the **Response Surface Optimization** system, double-click the **Design of Experiments** cell to open it.

   In the **Outline** pane under **Input Parameters**, you can see that five inputs are defined.

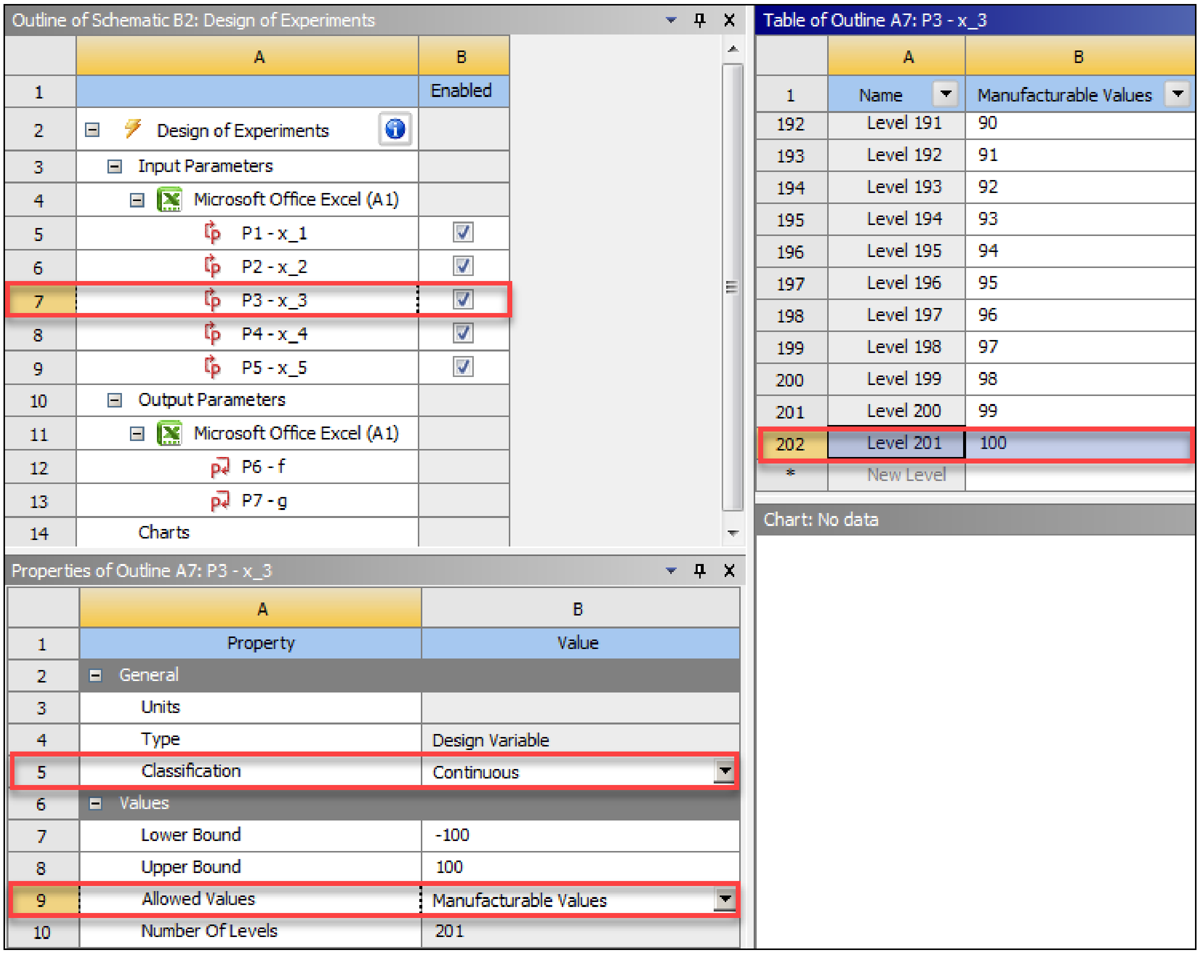


2. Select any one of these input parameters: **P3 - x_3**, **P4 - x_4**, or **P5 - x_5**.

   In the **Properties** pane, you can see that **Classification** is set to **Continuous** and **Allowed Values** is set to **Manufacturable Values**.

   In the **Table** pane for any of these three parameters, you can see levels that define the manufacturable values. Each of them has the same 201 levels.

   The following figure shows properties and manufacturable values for **P3 - x_3**.



The number of manufacturable values in the problem means that for full-factorial optimization of just these three parameters, the combination of values is equal to 8120601 (201^3). Unless each solve is very fast and you are running a Screening optimization on a cluster of machines via Ansys Remote Solve Manager, running all of the combinations is difficult. Consequently, you can conclude that for this problem, a **Direct Optimization** system using Screening is not likely to discover the optimum.

The only way to study all of these combinations is to assume that they have a smooth influence on the model functions and then either interpolate a response surface or use an optimization method that includes this assumption, such as MISQP.

3. Close the **Design of Experiments** cell.

Response Surface Optimization to Direct Optimization Using Screening and MISQP

For this problem, you'll use Screening and MISQP in two **Response Surface Optimization** systems to find a good starting point for MISQP in a **Direct Optimization** system. These are basically the steps that you'll take:

1. Run Screening in a **Response Surface Optimization** system to find a good candidate point.

2. Run MISQP in a **Response Surface Optimization** system, using the candidate point found by the Screening as its starting point.

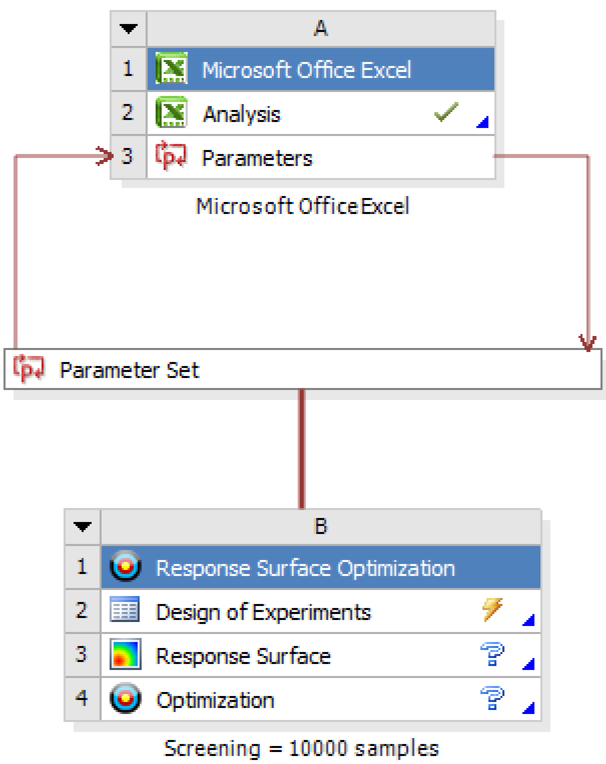
3. Run MISQP in a **Direct Optimization** system, using the candidate point found by the previous MISQP as a starting point.

Run Screening in a Response Surface Optimization

You can try this approach because the problem doesn't have a large number of input parameters and because you can evaluate the analytic functions at any point: both *on* manufacturable values and *between* manufacturable values—for example, at point **(1.33;100;1;-100;-1)** This means that it may be feasible to build a response surface.

First, you will run a Screening optimization to find a feasible candidate point to serve as a starting point for an MISQP optimization.

The **Project Schematic** already has a **Response Surface Optimization** system named **`Screening = 10000 samples`**.



For each cell, you need to configure properties and then update it and view results. For this first system, properties are already set. However, follow the instructions for setting properties and then update the cells as described.

Configure and Update the Design of Experiments Cell

1. In the **Project Schematic**, double-click the **Design of Experiments** cell to open it.

2. In the **Outline** pane, select **Design of Experiments**.

3. In the **Properties** pane:

   a. Set **Design of Experiments Type** to **Central Composite Design**.

   b. Set **Design Type** to **Auto Defined**.

4. Update the **Design of Experiments** cell.

   When the update is complete, the **Table** pane displays 27 design points and their results.

5. Close the **Design of Experiments** cell.

Configure and Update the Response Surface Cell

1. In the **Project Schematic**, double-click the **Response Surface** cell to open it.

2. In the **Outline** pane, select **Response Surface**.

3. In the **Properties** pane:

   a. Set **Response Surface Type** to **Kriging**.

   b. Set **Maximum Predicted Relative Error %** to **5**.

4. Update the **Response Surface** cell.

   The **Table** pane displays a response point.

5. Close the **Response Surface** cell.

Note:

Kriging with auto-refinement does not add new refinement points because, in this case, the response surface generated from the DOE is sufficiently accurate.

Configure and Update the Optimization Cell

1. In the **Project Schematic**, double-click the **Optimization** cell to open it.

2. In the **Outline** pane, select **Optimization**.

3. In the **Properties** pane:

   a. Set **Method Selection** to **Manual**.

   b. Set **Method Name** to **Screening**.

   c. Set **Number of Samples** to **10000**.

4. In the **Outline** pane, select **Objectives and Constraints**.

   The **Table** pane displays the objectives and constraints that have already been set:

Table of Schematic B4: Optimization

	A	B	C	D	E	F	G
1	Name	Parameter	Objective		Constraint		
2			Type	Target	Type	Lower Bound	Upper Bound
3	Minimize P6	P6 - f	Minimize		No Constraint		
4	P7 >= 0	P7 - g	No Objective		Values >= Lower Bound	0	
*		Select a Parameter					

5. Update the **Optimization** cell.

6. When the update is complete, in the **Outline** pane, select **Optimization**.

   In the **Table** pane, you can see that **Candidate Point 2** is the best candidate **(11.51;77.866;3;-94;7)**, with a corresponding objective value of **-9043.7** and a constraint value of **166.32**. This sample was chosen (screened) as the best of the 10,000 that were generated using the Kriging response surface meta-model, which was generated from the 27 design points that were actually solved for the DOE.

   The verification process has already been run. It takes the candidate input points and runs that design point through the actual solver (in this case, Excel). For this problem, verification found that the candidate was even better than predicted. The difference is due to the accuracy of the response surface prediction of the sample based on the 27 design points that were actually solved. However, because the Screening method is good at separating global maxima (or minima) from local maxima, this would be a good place to start an optimization with a more efficient method or even a direct solve method such as MISQP.

Table of Schematic B4: Optimization

	A	B	C	D	E
1	Optimization Study				
2	Minimize P6	Goal, Minimize P6 (Default importance)			
3	P7 >= 0	Strict Constraint, P7 values greater than or equals to 0 (Default importance)			
4	Optimization Method				
5	Screening	The Screening optimization method uses a simple approach based on sampling and sorting. It supports multiple objectives and constraints as well as all types of input parameters. Usually it is used for preliminary design, which may lead you to apply other methods for more refined optimization results.			
6	Configuration	Generate 10000 samples and find 3 candidates.			
7	Status	Converged after 10000 evaluations.			
8	Candidate Points				
9		Candidate Point 1	Candidate Point 2	Candidate Point 2 (verified)	Candidate Point 3
10	P1 - x_1	-19.99	11.51		-7.39
11	P2 - x_2	-95.351	77.866		-18.52
12	P3 - x_3	-7	3		-3
13	P4 - x_4	-100	-94		-92
14	P5 - x_5	5	7		0
15	P6 - f	-9322.3	-9043.7	-8237.3	-8678.8
16	P7 - g	353.69	166.32	166.32	37.7

7. Close the **Optimization** cell.

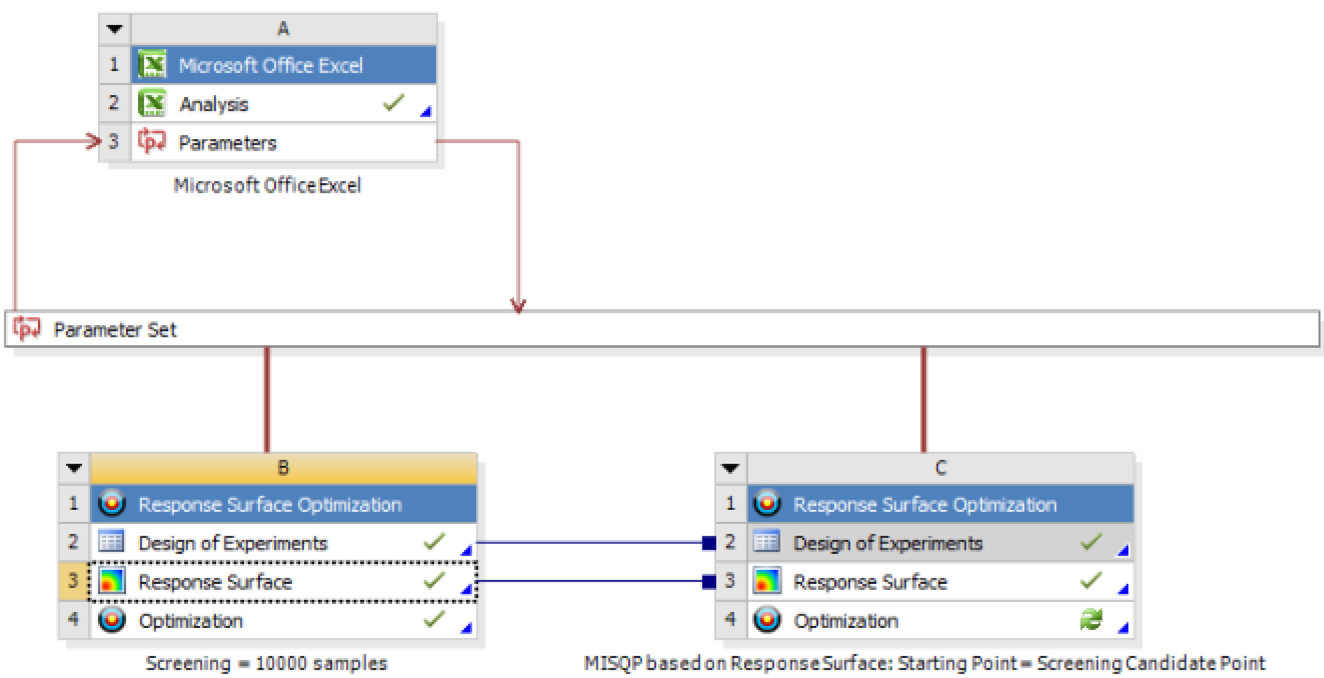
Run MISQP in a Response Surface Optimization

Now that Screening has found a suitable candidate, you'll use this candidate as the **Starting Value** in an MISQP response surface optimization.

Add Another Response Surface System

1. To add the second **Response Surface Optimization** system in the **Project Schematic**, right-click the header of the existing **Response Surface Optimization** system and select **Duplicate**.
2. Rename this new system to `MISQP based on Response Surface: Starting Point = Screening Candidate Point`.
3. Drag the **Design of Experiments** cell in the original system and drop it on the **Design of Experiments** cell in the new system so that these two cells share data.
4. In the dialog box that opens, click **OK** to confirm that the two cells are to share data.
5. Drag the **Response Surface** cell in the original system and drop it on the **Response Surface** cell in the new system so that these two cells share data.
6. In the dialog box that opens, click **OK** to confirm that the two cells are to share data.

   The **Project Schematic** will now look like this:



Configure and Update the Optimization Cell

1. In the second **Response Surface Optimization** system, double-click the **Optimization** cell to open it.
2. In the **Outline** pane, select **Optimization**.

3. In the **Properties** pane:

    a. Set **Method Selection** to **Manual**.

    b. Set **Method Name** to **MISQP**.

    c. Set **Allowable Convergence (%)** to **1E-06**.

    d. Set **Maximum Number of Iterations** to **55**.

4. In the **Outline** pane, select **Objectives and Constraints**.

5. In the **Table** pane, set the same objectives and constraints as in the other system:

Table of Schematic B4: Optimization

	A	B	C	D	E	F	G
1	Name	Parameter	Objective		Constraint		
2			Type	Target	Type	Lower Bound	Upper Bound
3	Minimize P6	P6 - f	Minimize		No Constraint		
4	P7 >= 0	P7 - g	No Objective		Values >= Lower Bound	0	
*		Select a Parameter					

6. In the **Outline** pane, select **Domain**.

    When **Optimization Type** is set to **MISQP**, the **Table** pane provides for entering not only lower and upper bounds but also starting values.

7. Adjust the starting values to match those for the candidate found by Screening **(11.51;77.866;3;-94;7)** as shown:

Table of Schematic C4: Optimization

	A	B	C	D
1	Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - x_1	-100	100	11.51
4	P2 - x_2	-100	100	77.866
5	P3 - x_3	-100	100	3
6	P4 - x_4	-100	100	-94
7	P5 - x_5	-100	100	7

8. Update the **Optimization** cell.

9. Once the update is complete, in the **Outline** pane, select **Optimization**.

    In the **Table** pane, you can see that **Candidate Point 1** is the best candidate **(1.323;94.688;1;-100;-1)**, with an objective value of **-10045** and a constraint value of **6.1205**. These values have been evaluated on the Kriging response surface.

Table of Schematic C4: Optimization

	A	B	C	D	E
1	⊟ Optimization Study				
2	Minimize P6	Goal, Minimize P6 (Default importance)			
3	P7 >= 0	Strict Constraint, P7 values greater than or equals to 0 (Default importance)			
4	⊟ Optimization Method				
5	MISQP	The MISQP method (Mixed-Integer Sequential Quadratic Programming) solves mixed-integer nonlinear programming problems by a modified sequential quadratic programming (SQP) method. Under the assumption that integer variables have a smooth influence on the model functions, i.e ., that function values do not change drastically when in- or decrementing an integer variable, successive quadratic approximations are applied. It supports a single objective and multiple constraints. The starting point must be specified to determine the region of the design space to explore.			
6	Configuration	Approximate derivatives by Central difference and find 3 candidates in a maximum of 55 iterations.			
7	Status	Converged after 559 evaluations.			
8	⊟ Candidate Points				
9		Starting Point	Candidate Point 1	Candidate Point 2	Candidate Point 3
10	P1 - x_1	11.51	1.323	6.8553	3.3107
11	P2 - x_2	77.866	94.688	-100	78.174
12	P3 - x_3	3	1	3	3
13	P4 - x_4	-94	-100	-100	-94
14	P5 - x_5	7	-1	7	7
15	P6 - f	★★★ -9043.7	★★★ -10045	★★★ -9753	★★★ -9111.2
16	P7 - g	★★★ 166.32	★★★ 6.1205	★★★ 69.196	★★★ 24.3

10. Right-click **Candidate Point 1** and select **Verify By Design Point Update**.

    In the following figure, these values have been verified by a design point update. For this verification point, the objective value is **-10094** and the constraint value remains the same at **6.1205**. Because this verification point (real solve) is very close to the predicted value, you now that the Kriging response surface is very accurate in this location.

Table of Schematic C4: Optimization

	A	B	C	D	E	F
1	⊟ Optimization Study					
2	Minimize P6	Goal, Minimize P6 (Default importance)				
3	P7 >= 0	Strict Constraint, P7 values greater than or equals to 0 (Default importance)				
4	⊟ Optimization Method					
5	MISQP	The MISQP method (Mixed-Integer Sequential Quadratic Programming) solves mixed-integer nonlinear programming problems by a modified sequential quadratic programming (SQP) method. Under the assumption that integer variables have a smooth influence on the model functions, i.e., that function values do not change drastically when in- or decrementing an integer variable, successive quadratic approximations are applied. It supports a single objective and multiple constraints. The starting point must be specified to determine the region of the design space to explore.				
6	Configuration	Approximate derivatives by Central difference and find 3 candidates in a maximum of 55 iterations.				
7	Status	Converged after 559 evaluations.				
8	⊟ Candidate Points					
9		Starting Point	Candidate Point 1	Candidate Point 1 (verified)	Candidate Point 2	Candidate Point 3
10	P1 - x_1	11.51	1.323		6.8553	3.3107
11	P2 - x_2	77.866	94.688		-100	78.174
12	P3 - x_3	3	1		3	3
13	P4 - x_4	-94	-100		-100	-94
14	P5 - x_5	7	-1		7	7
15	P6 - f	★★★ -9043.7	★★★ -10045	★★★ -10094	★★★ -9753	★★★ -9111.2
16	P7 - g	★★★ 166.32	★★★ 6.1205	★★★ 6.1205	★★★ 69.196	★★★ 24.3

Screening, which is somewhat random, missed this better location on the model because its resolution was not high enough. The benefit of Screening is that it spreads its samples across the entire domain so that it is better able to find the region of the global minimum or maximum. The region of the global minimum or maximum is an ideal starting point for an algorithm like MISQP, which is capable of a more active search but might otherwise have fallen into a local minimum.

11. Close the **Optimization** cell.

Run MISQP in a Direct Optimization

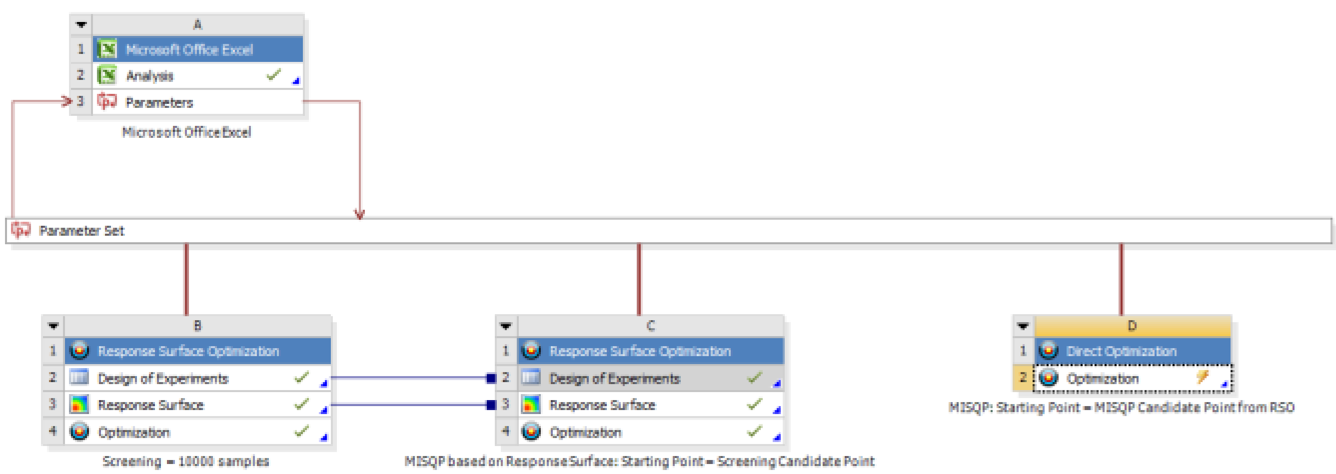
Finally, you'll use the candidate found by the MISQP response surface optimization as the starting value in an MISQP direct optimization.

Add a Direct Optimization System

1. Under **Design Exploration** in the **Toolbox**, drag the **Direct Optimization** system template and drop it on the **Project Schematic** to the right of the last system.

2. Rename it to `MISQP: Starting Point = MISQP Candidate Point from RSO`.

   The **Project Schematic** will now look like this:



Configure and Update the Optimization Cell

1. In the **Direct Optimization** system, double-click the **Optimization** cell to open it.

2. In the **Outline** pane, select **Optimization**.

3. In the **Properties** pane:

   a. Set **Method Selection** to **Manual**.

   b. Set **Method Name** to **MISQP**.

   c. Clear the **Tolerance Setting** check box.

   d. Set **Allowable Convergence (%)** to **1E-06**.

e. Set **Maximum Number of Iterations** to **20**.

4. In the **Outline** pane, select **Objectives and Constraints**.

5. In the **Table** pane, set the same objectives and constraints as in the other two systems:

Table of Schematic B4: Optimization

	A	B	C	D	E	F	G
1	Name	Parameter	Objective		Constraint		
2			Type	Target	Type	Lower Bound	Upper Bound
3	Minimize P6	P6 - f	Minimize		No Constraint		
4	P7 >= 0	P7 - g	No Objective		Values >= Lower Bound	0	
*		Select a Parameter					

6. In the **Outline** pane under **Domain**, do the following for input parameters **P3 - x_3**, **P4 - x_4**, and **P5 - x_5** in their respective **Properties** panes:

   a. Set **Allowed Values** to **Manufacturable Values**.

   b. Add the 201 manufacturable value levels by copying all manufacturable values from the **Design of Experiments** cell in the first **Response Surface Optimization** system and pasting them into this cell.

   To perform the paste operation, right-click in the **Manufacturable Values** column for the first row and select **Paste**. If a dialog box opens, indicating that an entered value falls outside of the current variation range and that existing results and design points will be lost, click **OK**.

7. In the **Outline** pane, select **Domain**.

   When **Optimization Type** is set to **MISQP**, the **Table** pane provides for entering not only lower and upper bounds but also starting values.

8. Adjust the starting values to match those for the candidate found by the previous MISQP optimization **(1.323;94.688;1;-100;-1)** as shown:

Table of Schematic D2: Optimization

	A	B	C	D
1	⊟ Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - x_1	-100	100	1.323
4	P2 - x_2	-100	100	94.688
5	P3 - x_3	-100	100	1
6	P4 - x_4	-100	100	-100
7	P5 - x_5	-100	100	-1
8	⊟ Parameter Relationships			
9	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	<=	*New Expression*

9. Update the **Optimization** cell.

Once the update is complete, in the **Outline** pane, select **Optimization**.

In the **Table** pane, the best candidate point is **(0.82299;95.014;1;-100;-1)**, with an objective value of **–10094** and a constraint value of **3.7973**.

Table of Schematic D2: Optimization

	A	B	C	D	E
1	Optimization Study				
2	Minimize P6	Goal, Minimize P6 (Default importance)			
3	P7 >= 0	Strict Constraint, P7 values greater than or equals to 0 (Default importance)			
4	Optimization Method				
5	MISQP	The MISQP method (Mixed-Integer Sequential Quadratic Programming) solves mixed-integer nonlinear programming problems by a modified sequential quadratic programming (SQP) method . Under the assumption that integer variables have a smooth influence on the model functions, i .e., that function values do not change drastically when in- or decrementing an integer variable, successive quadratic approximations are applied. It supports a single objective and multiple constraints. The starting point must be specified to determine the region of the design space to explore.			
6	Configuration	Approximate derivatives by Forward difference and find 3 candidates in a maximum of 20 iterations.			
7	Status	Converged after 102 evaluations.			
8	Candidate Points				
9		Starting Point	Candidate Point 1	Candidate Point 2	Candidate Point 3
10	P1 - x_1	1.323	0.82299	1.323	0.33
11	P2 - x_2	94.688	95.014	94.688	100
12	P3 - x_3	1	1	1	0
13	P4 - x_4	-100	-100	-100	-100
14	P5 - x_5	-1	-1	-1	-2
15	P6 - f	-10094	-10094	-10094	-10093
16	P7 - g	6.1203	3.7973	6.1203	1.4964

Response surface optimization allows you to approximate the objective and constraint values, explore them in the parameter space with a minimum number of design points, and find a good candidate point as starting point of the direct optimization. Essentially, this first step in a response surface-based exploration is an acceleration of a direct optimization. MISQP convergence can be very dependent on starting values, especially for non-convex problems.

10. Close the **Optimization** cell.

Direct Optimization Using MISQP

As demonstrated by the previous system, using response surface optimization can be an effective approach to finding a good starting value for MISQP. However, it's not always possible to build a response surface. Possible reasons include the following:

- Too many input parameters
- Too many combinations of discrete parameter levels
- Too difficult to approximate the objective or constraint values

When a response surface cannot be generated, a direct optimization is necessary. In this scenario, you will use MISQP for a direct optimization.

Run MISQP in a Direct Optimization with a Maximum of 20 Iterations

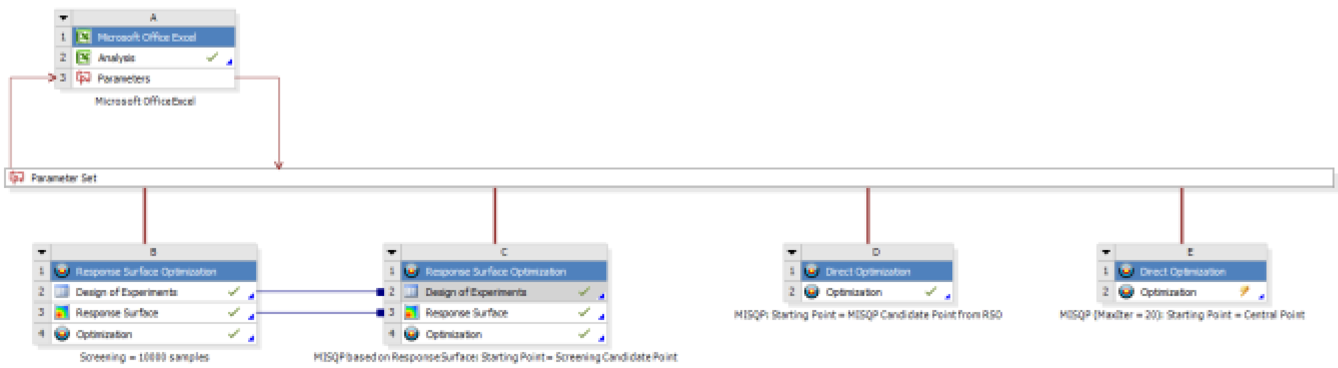
You'll now add a **Direct Optimization** system that uses the central point of the domain as the starting value for the MISQP optimization.

Add a Direct Optimization System

1. To add the system in the **Project Schematic**, right-click the header of the existing **Direct Optimization** system and select **Duplicate**.

2. Rename this new system to **`MISQP (MaxIter = 20): Starting Point = Central Point`**.

   The **Project Schematic** should now look like this:



Configure and Update the Optimization Cell

1. In this second **Direct Optimization** system, double-click the **Optimization** cell to open it.

2. In the **Outline** pane, select **Optimization**.

3. In the **Properties** pane:

   a. Set **Finite Difference Approximation** to **Central** to indicate the method for calculating the derivative.

   b. Set **Allowable Convergence (%)** to **1E-06**.

   c. Set **Maximum Number of Iterations** to **20**.

4. In the **Outline** pane, select **Objectives and Constraints**.

   The **Table** pane already displays the same objectives and constraints as in the other three systems:

Table of Schematic B4: Optimization

	A	B	C	D	E	F	G
1	Name	Parameter	Objective		Constraint		
2			Type	Target	Type	Lower Bound	Upper Bound
3	Minimize P6	P6 - f	Minimize		No Constraint		
4	P7 >= 0	P7 - g	No Objective		Values >= Lower Bound	0	
*		Select a Parameter					

5. In the **Outline** pane, select **Domain**.

6. Set all starting values to **0** because you are assuming a scenario with no response surface to find a starting point.

   The **Table** pane should look like this:

Table of Schematic E2: Optimization

	A	B	C	D
1	Input Parameters			
2	Name	Lower Bound	Upper Bound	Starting Value
3	P1 - x_1	-100	100	0
4	P2 - x_2	-100	100	0
5	P3 - x_3	-100	100	0
6	P4 - x_4	-100	100	0
7	P5 - x_5	-100	100	0
8	Parameter Relationships			
9	Name	Left Expression	Operator	Right Expression
*	*New Parameter Relationship*	*New Expression*	<=	*New Expression*

7. Update the **Optimization** cell.

8. Once the update is complete, in the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, you can see that MISQP reaches the maximum number of iterations (20) with 222 design points, and it does not converge.

   In the **Table** pane, the best candidate point is **(5.9531;100;2;-100;24)**, with an objective value of **–9416.1** and a constraint value of **34.119**.

Table of Schematic E2: Optimization

	A	B	C	D	E
1	⊟ Optimization Study				
2	Minimize P6	Goal, Minimize P6 (Default importance)			
3	P7 >= 0	Strict Constraint, P7 values greater than or equals to 0 (Default importance)			
4	⊟ Optimization Method				
5	MISQP	The MISQP method (Mixed-Integer Sequential Quadratic Programming) solves mixed-integer nonlinear programming problems by a modified sequential quadratic programming (SQP) method. Under the assumption that integer variables have a smooth influence on the model functions, i.e., that function values do not change drastically when in- or decrementing an integer variable, successive quadratic approximations are applied. It supports a single output parameter objective and multiple constraints. The starting point must be specified to determine the region of the design space to explore.			
6	Configuration	Approximate derivatives by Central difference and find 3 candidates in a maximum of 20 iterations.			
7	Status	Not Converged.			
8	⊟ Candidate Points				
9		Starting Point DP 0	Candidate Point 1	Candidate Point 2	Candidate Point 3
10	P1 - x_1	0	-5.9531	-1.0175	1.1165
11	P2 - x_2	0	100	100	100
12	P3 - x_3	0	2	1	1
13	P4 - x_4	0	-100	-37	-5
14	P5 - x_5	0	24	9	0
15	P6 - f	✗✗✗ 5.8189	★★★ -9416.1	✗✗✗ -506.71	✗✗✗ -100.49
16	P7 - g	★★★ 1.5625	★★★ 34.119	★★★ 4.554	★★★ 5.6002

9. Close the **Optimization** cell.

Run MISQP in a Direct Optimization with a Maximum of 50 Iterations

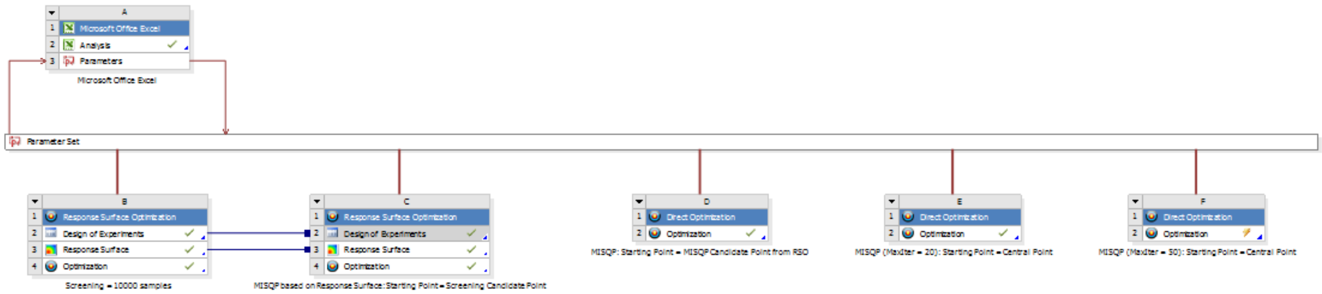
In the previous MISQP direct optimization, the candidate is very close to the optimum. To try to reach convergence, you can perform the same optimization but increase the maximum number of iterations.

Add Another Direct Optimization System

1. To add another **Direct Optimization** system in the **Project Schematic**, right-click the header of the previous **Direct Optimization** system and select **Duplicate**.

2. Rename this new system to **`MISQP (MaxIter = 50): Starting Point = Central Point`**.

   The **Project Schematic** should now look like this:



Configure and Update the Optimization Cell

1. In this third **Direct Optimization** system, double-click the **Optimization** cell to open it.

2. In the **Outline** pane, select **Optimization**.

3. In the **Properties** pane, set **Maximum Number of Iterations** to **50**, leaving the settings for all other properties the same.

4. Update the **Optimization** cell.

   Because you changed only **Maximum Number of Iterations**, this MISQP optimization reuses the design points generated by the previous one.

5. Once the update is complete, in the **Outline** pane, select **Optimization**.

   In the **Properties** pane under **Optimization Status**, you can see that MISQP converges in 40 iterations with 402 design points.

   In the **Table** pane, you can see that MISQP did now find the optimum point **(1.33;100;1;-100;-1)**, with an objective value of **–10100** and a constraint value of **6.1564**.

Table of Schematic F2: Optimization					
	A	B	C	D	E
1	Optimization Study				
2	Minimize P6	Goal, Minimize P6 (Default importance)			
3	P7 >= 0	Strict Constraint, P7 values greater than or equals to 0 (Default importance)			
4	Optimization Method				
5	MISQP	The MISQP method (Mixed-Integer Sequential Quadratic Programming) solves mixed-integer nonlinear programming problems by a modified sequential quadratic programming (SQP) method. Under the assumption that integer variables have a smooth influence on the model functions, i.e., that function values do not change drastically when in- or decrementing an integer variable, successive quadratic approximations are applied. It supports a single output parameter objective and multiple constraints. The starting point must be specified to determine the region of the design space to explore.			
6	Configuration	Approximate derivatives by Central difference and find 3 candidates in a maximum of 50 iterations.			
7	Status	Converged after 402 evaluations.			
8	Candidate Points				
9		Starting Point DP 0	Candidate Point 1	Candidate Point 2	Candidate Point 3
10	P1 - x_1	0	1.33	-8.8815	-1.0175
11	P2 - x_2	0	100	100	100
12	P3 - x_3	0	1	4	1
13	P4 - x_4	0	-100	-100	-37
14	P5 - x_5	0	-1	9	9
15	P6 - f	5.8189	-10100	-9786.1	-506.71
16	P7 - g	1.5625	6.1564	62.739	4.554

The candidate point found by this second MISQP is more accurate than the candidate point found by the first MISQP. To achieve the improved accuracy, the second optimization required more iterations and more design points (402 versus 222).

Tip:

DesignXplorer's History charts allow you to view the optimization history of objective and constraint values. You can stop the optimization at any time if you think that the current values are good enough. In this case, the candidate points will be extracted from the existing samples.

6. Close the **Optimization** cell.

7. Exit Workbench, saving project changes.

What Have You Learned?

In summary, for the analytic function problem, you tried three different optimization approaches:

1. **Running a Screening direct optimization**

Screening is an exhaustive method that distributes solutions across all Pareto fronts (across the entire design space, instead of focusing refinement on a section). A direct optimization uses real solves rather than estimates. In addition, 3 of the 5 input parameters each have 201 manufacturable value levels. Given these factors, a Screening direct optimization is not likely to find the optimum. The only way to study all of the combinations is to assume that they have a smooth influence on the model functions and then run either a DOE and interpolate a response surface or use an optimization method like MISQP that includes this assumption.

2. **Running Screening and MISQP response surface optimizations to find a starting value for an MISQP direct optimization**

   MISQP can be very dependent on the quality of its starting point, especially in a non-convex problem. The closer the starting point is to the optimum point, the better the probability of finding the optimum and the fewer the design points needed for convergence. You used a Screening response surface optimization to locate a starting point for an MISQP response surface optimization, which in turn found a starting point for an MISQP direct optimization. Because MISQP had a good starting point, the MISQP direct optimization needed only 222 design points to locate the optimum. In cases where it is possible to build a response surface, this is a good approach for solving problems of this type.

3. **Running an MISQP direct optimization**

   In cases where it is not possible to build a response surface, you can run MISQP in a direct optimization. MISQP still needs a good starting point, so in the absence of a response surface, you used the central point of the domain to initialize the optimization. In the first attempt, the optimization did not converge, but you could see that it was close to convergence. When you increased the maximum number of iterations, MISQP converged in 402 design points and obtained slightly more accurate results. Although more design points are needed for this approach, it is still possible to find the global minimum, even without a preliminary response surface.

The following table summarizes the results for each approach.

	x1	**x2**	**x3**	**x4**	**x5**	**f**	**g**	**Design Points Needed**
Global Minimum	1.33	100	1	-100	-1	-10099.69	6.1564	---
Response Surface Optimization Screening	11.51	77.866	3	-94	7	-9043.7	166.32	27+1
Response Surface Optimization MISQP (Starting Value = RSO Screening Candidate)	1.323	94.688	1	-100	-1	-10045	6.12095	53+1
Direct Optimization MISQP (Starting Value = RSO MISQP Candidate)	0.82299	95.014	1	-100	-1	-10094	3.7973	102
Direct Optimization MISQP	-5.9531	100	2	-100	24	-9416.1	34.119	222

	x1	x2	x3	x4	x5	f	g	Design Points Needed
(Max. Iterations = 20; Staring Value = Central Point)								
Direct Optimization MISQP (Max. Iterations = 50; Staring Value = Central Point)	1.33	100	1	-100	-1	-10100	6.1564	402

Multiple-Objective Optimization Problems

For multiple-objective optimization problems, you can define a derived output parameter that is a combination of your objective functions.