

Siemens PLM Software

## HEEDS MDO 2019.1 Getting Started Guide

[www.redcedartech.com](http://www.redcedartech.com)



<b>Part 1 Introduction .....</b>	<b>1</b>
<b>Introducing HEEDS MDO .....</b>	<b>2</b>
What is Optimization? .....	2
Defining an Optimization Problem .....	3
How HEEDS MDO Accelerates Optimization .....	3
Parameter Optimization.....	6
Incorporating DOE into Your Optimization Search .....	6
Robustness and Reliability Testing .....	6
Evaluation .....	6
<b>Anatomy of a HEEDS MDO Project.....</b>	<b>7</b>
Starting Out: The Baseline Design .....	7
Process Automation .....	7
Exploration .....	11
<b>The HEEDS MDO Environment .....</b>	<b>15</b>
Starting HEEDS MDO .....	15
Starting a New Project .....	16
The HEEDS MDO Layout.....	17
Entering Information.....	18
Viewing Table Data.....	28
Saving and Retrieving HEEDS MDO Projects .....	28
Your Next Step .....	29
<b>Part 2 Example Problems.....</b>	<b>31</b>
<b>Example 1: A Simple Function Problem.....</b>	<b>33</b>
Step 1: Starting a New Project .....	34
Step 2: Defining the Process .....	34
Step 3: Defining the Parameters .....	37
Step 4: Tagging the Input and Output Files .....	41
Step 5: Defining the Study.....	42
Step 6: Running the Study .....	46
Step 7: Viewing Results Using HEEDS POST .....	50
Step 8: Looking at the Results Files .....	54
<b>Example 2: Design of a Cantilevered Beam.....</b>	<b>55</b>
Step 1: Starting a New Project .....	56
Step 2: Defining the Process .....	56
Step 3: Defining the Parameters .....	61
Step 4: Tagging the Input and Output Files .....	66
Step 5: Defining the Study.....	69
Step 6: Running the Study .....	74
Step 7: Viewing Results Using HEEDS POST .....	78
Step 8: Reviewing the Log Files .....	84
<b>Example 3: 3-Bar Truss Mechanism .....</b>	<b>86</b>
Step 1: Starting a New Project .....	88
Step 2: Defining the Process .....	88

Step 3: Defining the Parameters.....	92
Step 4: Tagging the Input and Output Files.....	96
Step 5: Defining the Study .....	103
Step 6: Running the Study.....	107
Step 7: Viewing Results Using HEEDS POST .....	110
Step 8: Animating the plot.....	116
<b>Example 4: DOE Study of a Coil Spring.....</b>	<b>117</b>
Step 1: Starting a New Project.....	118
Step 2: Defining the Process .....	118
Step 3: Defining the Parameters.....	119
Step 4: Tagging the Input and Output Files.....	122
Step 5: Defining the Study .....	124
Step 6: Running the Study.....	125
Step 7: Viewing Results Using HEEDS POST .....	126
<b>Example 5: Robustness and Reliability .....</b>	<b>135</b>
Step 1: Copying the Project .....	135
Step 2: Editing the Parameters .....	136
Step 3: Tagging the Input and Output Files.....	138
Step 4: Defining the Reliability Study.....	140
Step 5: Running the Reliability Study.....	140
Step 6: Viewing the Results of the Reliability Study.....	141
Step 7: Defining the Robustness Study .....	141
Step 8: Running the Robustness Study .....	142
Step 9: Viewing the Results of the Robustness Study .....	142
<b>Example 6: Multi-Objective Optimization.....</b>	<b>144</b>
Step 1: Starting a New Project.....	145
Step 2: Defining the Process .....	145
Step 3: Defining the Parameters.....	146
Step 4: Tagging the Input and Output Files.....	147
Step 5: Defining the Study .....	148
Step 6: Running the Study .....	150
Step 7: Viewing the Results .....	151
<b>Example 7: Population Modeling Using Curve Fitting.....</b>	<b>156</b>
Step 1: Starting a New Project.....	157
Step 2: Defining the Process .....	157
Step 3: Defining the Parameters.....	159
Step 4: Tagging the Input and Output Files.....	162
Step 5: Defining the Study .....	164
Step 6: Running the Study.....	165
Step 7: Viewing the Results .....	165
References .....	166
<b>Example 8: Designing a Pipe Network with Taguchi RPD.....</b>	<b>168</b>
Step 1: Starting a New Project.....	170
Step 2: Defining the Process .....	170
Step 3: Defining the Parameters.....	170
Step 4: Tagging the Input and Output Files.....	172
Step 5: Defining the Study .....	173

Step 6: Running the Study .....	175
Step 7: Viewing the Results .....	175
Step 8: Identifying the Best Design .....	177
References.....	177



# **Part 1**

## **Introduction**

# Introducing HEEDS MDO

---

HEEDS® MDO is a robust multidisciplinary design optimization software package that automates the search for better solutions within a given design space and dramatically reduces design time.

HEEDS MDO enables you to quickly find designs that perform extremely well according to multiple criteria, while simultaneously satisfying multiple constraints and using a large number of variables. In an overnight run, HEEDS can produce results that would take weeks to accomplish manually.

With HEEDS MDO, you have access to all of the following design exploration and optimization capabilities:

- Identifying optimal solutions within a given design space using advanced mathematical search algorithms and strategies.
- Performing multiple-objective, Pareto-optimization studies to generate a Pareto set of optimal designs which let you perform trade-off studies between conflicting objectives.
- Performing design of experiments (DOE) studies to examine how selected variables affect a design.
- Testing designs for robustness and reliability to determine how sensitive a design is to expected variations in designated project variables.
- Executing and post-processing designs that you specify.

Depending on your project requirements, you may use HEEDS MDO to perform any of the above functions. On a given project, you might use more than one, selecting the one that is most appropriate for each stage of your project's development. For example, you might set up a parameter optimization study and search for an optimized design that meets all of your objectives and constraints. After that, you might take your best design(s) and use a DOE study to identify critical tolerances based on the sensitivity of the design to certain variables. Individual designs could be studied further by testing their robustness and reliability in yet another study in the project.

Although the technology behind HEEDS MDO and its powerful search engine is complex, the software itself is user-friendly and easy to use. You do not need to be an expert in optimization theory in order to use it effectively. Simply create your design model with the analysis software you normally use. Then, set up your design study in HEEDS MDO. When your study runs, HEEDS MDO automates and accelerates the natural design iteration process, using the same analysis tool(s) you used to create your design.

## What is Optimization?

---

As an engineer, your goal in creating something new is typically to design it to be *as good as it can be* in some sense. That may mean that it should last as long as possible, perform as efficiently as possible, or be as easy to maintain as possible. Or it may mean that it should be ready for market in as short a time as possible or be as inexpensive to manufacture as possible. If the product is to replace an existing one, it needs to be at least as good as the old one in many respects and *better* in others. All of these reasons for creating a new design have one common goal—to improve one or more aspects of the design so that it results in a better product. This is what optimization is all about.

*Webster's Collegiate Dictionary* defines optimization as “an act, process, or methodology of making something (a design, system, or decision) as fully perfect, functional, or effective as possible.” Based on this definition, true optimization is making something *as good as it can possibly be*. Closely related to optimization is *improvement*—that is, making something *better than it currently is*. Traditional optimization has focused on improvement rather than true optimization. The result is not necessarily the best design possible, but merely a design that is improved over the original. True optimization, on the other hand, results in the best design possible under a given set of circumstances.

When you set out to optimize a design, you must decide what aspect(s) of the design need to be improved (the *objective* or goal) and what restrictions or limiting factors need to be applied (the *constraints*). The *objective* is whatever property of the design is to be maximized (e.g., strength or durability) or minimized (e.g., cost, weight, etc.). The *constraints* are the limitations you want to place on the design. The constraints may specify how big the part must be, how it must connect to various other objects, what the most extreme operating conditions that it can tolerate without failing must be, how much it can cost, etc. The best possible design will be the one that most effectively meets the objective while staying within the constraints.

There are often many different ways to look at the same problem. For example, you might specify the maximum allowable weight of a part (a constraint) and try to maximize the load it can tolerate without failing (the goal or objective). Alternatively, you might seek to minimize the weight (the objective) while enforcing that it cannot break at a certain load (a constraint). The solutions will likely be different, but each might be appropriate for a given situation.

## Defining an Optimization Problem

To optimize a design, you need to identify its measurable parts. Every optimization problem can be specified in terms of the following:

- One or more *objectives* to determine how good a design is. The objective is what you are trying to maximize or minimize.
- A set of *constraints* that will determine whether or not a design is feasible.
- A *baseline design* to use as the frame of reference.
- A set of *variables* that specifies what values can be changed from the baseline design in order to define a new design.

Once you have identified your variables, you need to determine their allowable values. The combined allowable range of all your variables determines the *design space* that will be searched.

A design that simply meets or satisfies all of the constraints is called a *feasible design*, regardless of how well it meets its objective. There may be many feasible designs that are unsatisfactory, but what we look for in optimization is the *best* of the feasible designs—the one that maximizes or minimizes the objective as much as possible while still satisfying the constraints.

## How HEEDS MDO Accelerates Optimization

To solve an optimization problem manually, you would need to choose combinations of the variables and perform an analysis for each combination. Then you would examine the output to evaluate how well it fulfills the objectives and constraints, and determine what a good set of variables for the next run might be. Eventually, given enough time, you would come up with a solution that was “more optimal” than the baseline design. For a simple problem, you might even come up with the “best” solution.

However, given the nature of most engineering design environments, limitations on time and resources prevent manual search methods from reliably producing the best possible design. Instead, you must settle for “better” or “good enough.” This is where HEEDS MDO comes in. Taking the same software tools that you would otherwise use in a manual design process (a finite element package, a computational fluid dynamics package, a dynamic systems package, a circuit simulation package, etc.), HEEDS MDO, running unattended, can search the design space for the best design by automating the evaluation process and executing it numerous times, varying the variables each time, based on an intelligent search strategy, until it has performed the number of evaluations you specify.

You simply create your baseline design, run it through your normal analysis/simulation tools (the tools you would use if HEEDS MDO were not available), and save the input and output files. Then, using HEEDS MDO, you (1) identify your project variables and the responses that contain your objective(s) and your constraints and (2) define the con-

straints (whether a response should be  $\geq$  or  $\leq$  a particular value) and objectives (whether a response should be maximized or minimized).

## How HEEDS MDO optimization works

HEEDS MDO starts by picking values for the project variables within the limitations you specify. It evaluates each design with the same analysis software tool(s) that produced the baseline input and output files. During an evaluation, HEEDS MDO will automatically pick which hybrid search procedures to use, depending on the results that are returned, unless you override it with your own specific directions. As the search progresses, it uses intelligent selection processes, based on what worked best (or did not work) in previous designs.

HEEDS MDO writes the data for each design to its working files. During a given run, HEEDS MDO can evaluate hundreds of designs—a far greater number than can typically be produced in a manual design search. It doesn't take an engineer to figure out how much more likely you are to find the "optimal" design with HEEDS MDO than with the manual method.

In HEEDS MDO a search is performed by setting up a study. HEEDS MDO allows multiple studies to be setup within the same project. Within each study, you can independently specify the number of variables and their representation, the search type (DOE, Optimization, Reliability, Evaluation), the goals of the search, the search algorithms to be used, the analysis model(s) to be evaluated, and more. The simplest HEEDS MDO projects use a single study to search for an optimal design and a single analysis tool in the process to run the evaluation. More complex problems can employ multiple analysis tools in the process. The different studies can be setup to look at different aspects of the problem, or to perform optimization, DOE and reliability on the same problem. All of this can be specified easily within HEEDS MDO.

## How HEEDS MDO judges designs

During optimization, HEEDS MDO gives each design a performance rating. The value returned for the objective(s) and the degree to which a design satisfies its constraints together determine the design's *performance value*. A high-performance design is one that satisfies all constraints and has a good rating on its objective(s).

All designs that satisfy the chosen constraints essentially ignore the margin by which they meet those constraints. Once the constraints are satisfied, only the objectives contribute numerically to the performance evaluation. The *performance value* of each design is calculated using the following equation:

$$\sum_{i=1}^{Nobj} \left( \frac{LinWt_i * S_i * Obj_i}{Norm_i} + \frac{QuadWt_i * S_i * Obj_i^2}{Norm_i^2} \right) - \sum_{j=1}^{Ncon} \left( \frac{LinWt_j * ConViol_j}{Norm_j} + \frac{QuadWt_j * ConViol_j^2}{Norm_j^2} \right)$$

OR

When the linear and quadratic weight variables are set to their default values:

$$\sum_{i=1}^{Nobj} \left( \frac{LinWt_i * S_i * Obj_i}{Norm_i} \right) - \sum_{j=1}^{Ncon} \left( \frac{QuadWt_j * ConViol_j^2}{Norm_j^2} \right)$$

Variable	Definition
<b>Nobj</b>	Number of objectives in the optimization study.
<b>LinWt<sub>i</sub></b>	The linear weight for the $i^{\text{th}}$ objective. The default value is 1.
<b>S<sub>i</sub></b>	Sign for the $i^{\text{th}}$ objective. The value is -1 for objectives being minimized ( <b>Minimize</b> and <b>Minimize Difference</b> options) and +1 for objectives being maximized ( <b>Maximize</b> and <b>Maximize Difference</b> options).
<b>Obj<sub>i</sub></b>	For the <b>Minimize</b> and <b>Maximize</b> objective options, it's the response value for the $i^{\text{th}}$ objective for that design. For the <b>Minimize Difference</b> and <b>Maximize Difference</b> objective options, it's the square of the difference between the response value of the $i^{\text{th}}$ objective and the target value for the $i^{\text{th}}$ objective divided by the norm.
<b>Norm<sub>i</sub></b>	The normalizing value for the $i^{\text{th}}$ objective.
<b>QuadWt<sub>i</sub></b>	The quadratic weight for the $i^{\text{th}}$ objective. The default value is 0.
<b>Ncon</b>	Number of constraints in the optimization study.
<b>LinWt<sub>j</sub></b>	The linear weight for the $j^{\text{th}}$ constraint. The default value is 0.
<b>ConViol<sub>j</sub></b>	The amount by which the $j^{\text{th}}$ constraint is violated. This value is 0.0 if the constraint is met.
<b>Norm<sub>j</sub></b>	The normalizing value for the $j^{\text{th}}$ constraint.
<b>QuadWt<sub>j</sub></b>	The quadratic weight for the $j^{\text{th}}$ constraint. The default value is 10000.0.

### Design feasibility

A design that simply meets or satisfies all of the constraints is called a *feasible design*, regardless of how well it meets its objective. There may be many feasible designs that are unsatisfactory, but what we look for in optimization is the *best* of the feasible designs—the one that maximizes or minimizes the objective as much as possible while still satisfying the constraints.

As indicated by the performance value equation given above, for feasible designs (all constraints are satisfied), the performance function is a sum of the normalized objective values. When one or more constraints are violated, the performance value of the design is reduced by a value based on the violation of the constraint (the second term in the equation).

Feasibility is technically an either/or property—any violation of any constraint means that a design is infeasible. However, it is difficult to treat feasibility in that way for many real-world problems. Instead of treating all infeasible designs equally, HEEDS MDO computes a *penalty factor* for each constraint that is violated. For example, a constraint may say that the first natural frequency of a structural system must be greater than a specific fixed value. It does not matter how much greater, as long as it is that value or larger. Small violations of that constraint would be assigned smaller penalties than large violations. These penalties are combined with the objective rating to calculate the overall *performance value* of a design. Performance is good for designs with a good score on the objective and no violations of the constraints; performance gets worse when the objective score gets worse or constraints are violated.

Since the constraints are enforced using the penalty method, in certain cases, the reduction in performance of a design due to a small violation in constraint(s) can be offset by the increase in the performance because of the improvement in the objective the same time. This means that you can have slightly infeasible designs that have a higher performance value than designs that are feasible. In fact, it is possible that the best design reported is slightly infeasible for this reason. If no amount of infeasibility is acceptable, the weights for the constraint definitions should be increased and/or the normalizing factor value should be reduced such that even a small violation in the constraint results in a large reduction in the performance value.

By not eliminating designs on the basis of feasibility, HEEDS MDO can still provide results that are valuable in optimizing the design, even if no designs in a given run are feasible. HEEDS MDO provides the best of the infeasible designs, which you can use to recast your design problem (perhaps by extending the range or number of the variables or by relaxing some of the constraints) and let HEEDS MDO try again. This means that a run that does not produce desirable results is not a failed run but a step in the direction of optimizing the design.

## Parameter Optimization

---

In a HEEDS MDO parameter optimization study, optimized designs are sought by iteratively changing the values of tagged variables, executing the defined analyses, and extracting the responses for each design candidate. New design candidates are generated based on the mathematical search algorithms in HEEDS MDO.

## Incorporating DOE into Your Optimization Search

---

HEEDS MDO provides standard design of experiments (DOE) capabilities from within HEEDS MDO. A DOE study can add additional power to the optimization process and provide a greater understanding of the physics of the design by determining how variables affect the design's performance. After you run a parameter optimization study, you can set up a DOE study to identify which of the variables have the greatest influence on your responses and which have little or no effect.

When you set up a DOE, you choose the factors (variables) you want to test and the method you want to use to set up the DOE matrix. Using the DOE matrix, HEEDS MDO generates the designs and uses the responses you specify to evaluate the influence of your selected factors and also the influence of the interaction between these factors. The results are reported in post-processing plots, which can be displayed within HEEDS MDO.

## Robustness and Reliability Testing

---

To further evaluate your designs, you can test them to see how robust they are against standard tolerances and expected deviations in design parameters and environmental conditions. You can build your robustness and reliability study from your optimization study, or you can create a study to test a design that you created from a manual optimization. Setting up a robustness and reliability study is easy. You simply tell HEEDS MDO which variables you want to vary during the evaluation and set up stochastic distributions for them. HEEDS MDO will select values for these variables based on the defined distribution, either randomly or by applying a structured sampling methodology. Based on the results, you can determine the overall robustness of the design and tweak your design if necessary to produce more desirable results.

## Evaluation

---

In a HEEDS MDO Evaluation study, you specify the designs that you want to evaluate. With this type of study, you are using the process automation capabilities of HEEDS MDO to automate the creation and execution of, and results extraction for, different combinations of design variables that you specify.

# Anatomy of a HEEDS MDO Project

Now that you have a basic idea of what HEEDS MDO does, let's take a closer look at a HEEDS MDO project. A HEEDS MDO project is a collection of one or more search studies. There are four types of studies in a project: (1) parameter optimization studies, which search for an optimal design, or a Pareto set of optimal designs; (2) design of experiments (DOE) studies, which help you determine how variables influence the performance of your design; (3) robustness and reliability studies, which test your design against expected real-world deviations in your design and environmental variables; and (4) evaluation studies, which allow you to execute and post-process designs that you specify.

## Starting Out: The Baseline Design

Your first step, regardless of project or study type, is to create a baseline design in your analysis software. It does not have to be a good design or even a feasible one. It just has to be a workable one.

You create an input file just as you would if you were not using HEEDS MDO, assigning reasonable values that are representative of your design space to the quantities that will become the variables in your HEEDS MDO design study. Then, you run your analysis software to produce an output file and check the results to make sure the input file is error free.

HEEDS MDO uses the input and output file(s) from your design as a basis for creating and evaluating the design candidates. Except for the variables, the input file is used as is for each evaluation during a HEEDS MDO run. The output file is used only to determine the location of the values that will be used in determining the results of the study.

## Process Automation

The high-level setup for a project with HEEDS MDO begins with *Process Automation*. This level of setup automates your project evaluation process and is fundamentally the same for all types of projects. *Process Automation* includes three task levels: defining a *Process*, defining *Parameters*, and *Tagging* variables and responses. Tabs on the HEEDS interface reflect these task levels.

- *Process* is the level of your project where you define the analysis tools that will be used for evaluation of the design, identify the input and output files for each analysis, and select the command to execute the corresponding analysis tool.
- *Parameters* are the variables and responses that you define for your project.
- *Tagging* is how you link the values of variables and responses to locations in the input and output files.

In this chapter, we explain the requirements for each task level. In the next chapter, we give you an overview of HEEDS MDO. From this point you will be ready to try the sample projects in Part 2 of this guide.

## Processes and analyses

In HEEDS MDO, a simulation run of one specific design is called an *evaluation*. At the very least, one analysis will be executed for each design evaluated. In some cases, several separate analyses may be required to judge a single design. The set of analyses required to evaluate a design is called a *process*. Each study must have a process associated with it that contains the analyses that the study will perform. If all your studies perform the same analyses, they can share the same process. If your studies require a different set of analyses to be performed, you will need to create separate processes that contain the analyses specific to each study.

A minimal analysis definition consists of:

- An input file that contains the input values for the analysis.

- A command that executes the tool that is to be used for the analysis.
- An output file that contains the values from a successful analysis run.

Additional input and output files may also be referenced in a single analysis and/or more than one analysis may be required to complete the design evaluation. The command to execute is required for all analyses, except for some analyses with direct portals in which case the execution command is filled in automatically by HEEDS MDO. Some types of projects may not require input files or output files, although most do.

When you define an analysis in your HEEDS MDO project, you enter the command to run the analysis tool and identify the input and output files that are required for the analysis. The command must contain the exact text that would be entered at a command line prompt. The tool you use for a given analysis must be the one you used for the baseline design.

### **Input files**

HEEDS MDO uses the input file(s) from your baseline design as a template for creating the input file(s) for the designs to be evaluated. Each time it tries a new design, it writes out new input file(s), where it assigns new values to the variables. The input file(s) must contain values that represent the variables you intend to use in the evaluation. In projects where more than one tool is used in evaluating a design or more than one input file is used, not all of your defined variables will necessarily be included in a given input file. All files required by the tool for a successful analysis *must* be added as input files to the setup.

### **Output files**

HEEDS MDO uses the output files that were generated from your baseline design as a template for identifying the information used in determining the results of your study. In HEEDS MDO, these values are called responses. Your output file(s) must have been generated by your analysis tool and must contain values for the responses you want HEEDS MDO to use to compute design performance and/or post-processing results. Only the output files that contain the data needed for calculating the responses need to be added to the project.

## **Parameters: variables and responses**

During a HEEDS MDO run, designated quantities in the input file(s) are modified based on specifications you set. These quantities are represented by the project variables. To extract the results of a design, HEEDS MDO looks at values at the specified locations in the output file(s). These values are the project responses.

### **Types of project variables**

When you define a variable, you give it a name. HEEDS MDO Solver will use that name to identify the variable in the evaluation. You also assign it a type: *continuous*, *discrete*, *dependent*, or *parameter (numeric or text)*. The types of variables that are active in your project, and the way they are used, will depend on whether the project is a parameter optimization, DOE, robustness and reliability, or evaluation study.

#### **Continuous variables**

The values of *continuous variables* are chosen from a range of real numbers. Quantities like lengths, thicknesses, coordinates, etc., are typically specified as continuous variables.

When you define a continuous variable, you specify the *minimum value* and the *maximum value* of the range. For example, a minimum value of 2.0 and a maximum of 200.0 would specify real numbers between (and including) 2 and 200. During a HEEDS MDO run, the value is varied within the specified range.

#### **Discretization of continuous variables**

Most search methods employed by parameter optimization studies use a discretized form of the continuous variable to limit the possible choices. Instead of allowing HEEDS MDO Solver to select values totally at random from the range, continuous variables are discretized based on the level of refinement you specify when you define the study.

This level of refinement is the *resolution*. If you specified a range of 0 to 100 and set the resolution to 11, only values 0, 10, 20, ...100 would be used in an even distribution. If you specified a resolution of 101, any integer value from 0 to 100 could be used. More will be said about this when we discuss defining study variables during project assembly (see *Study variables*).

No resolution is specified for continuous variables in a DOE study. The values used are determined by the DOE method and are selected based on the maximum and minimum values of the range, unless you override them. Continuous variables are not varied in robustness and reliability studies. If they are present in the study, their baseline values are used.

Discretization of continuous variables reduces the size of the design space. This generally has a very beneficial effect on search efficiency whenever non-gradient-based search methods are used. There are also other advantages to variable discretization.

First, from a practical point of view, there is no reason to specify a value that cannot be achieved in practice. Second, discretization allows you to use more than one study to search the design space simultaneously at different resolutions. A study searching at a low resolution can quickly identify regions of the design space with high potential (and filter out those regions with low potential), while a second study, searching at a higher resolution and using information from the low-resolution search, can concentrate on the regions with high potential for a more efficient search. In the rare cases where continuous variations are desirable, the resolution can be set as high as necessary to mimic the behavior of a true continuous variable within the precision limits of the computer.

### Stochastic continuous variables

When a value is assigned to a variable, that value is often assumed to be exact (or *deterministic*). In other words, any possible variation of that value in the physical world is ignored. Often, however, the value of a variable may be known to be only within a certain tolerance, within which a distribution of values may exist. In this case, we say that the variable is *stochastic*, and a distribution about the mean of the variable is defined.

You can apply stochastic distributions to your continuous variables to account for known tolerances (e.g., variations in material thickness) and to test the robustness of a design against expected deviations. Stochasticity is defined at the project level when you define your variables (see Types of project variables on page 8). It is applied to your variables at the study level.

It is important to note that applying stochasticity may significantly increase the number of evaluations performed, depending on the percentage of variables that are stochastic.

It can often be beneficial to use two studies when you use stochastic variables, where one study, working with non-stochastic variables, feeds a second study, which then applies stochastic distributions to designs that have already been determined “good.” Your first study might, for example, determine that 2.25 is the optimal thickness and pass that design to the second study. The second study would then test the robustness of the design against the specified variations and modify the design as needed to increase its robustness. For more information about working with multiple studies, see the *HEEDS MDO online help*.

### Discrete variables

*Discrete variables* can only take values from a specified set of choices. This set may contain specific numeric values. It may also contain non-numeric choices (e.g., steel, aluminum, titanium). Discrete variables are used only when specific values are valid.

When you define a discrete variable, you create a discrete set, which contains the allowable values. If only three thicknesses are possible for a given material, for example, and the possible thickness values are not evenly distributed, you would specify the thickness as a discrete variable and reference the three thicknesses through a discrete set. There is no point in allowing HEEDS MDO Solver to select any number within a given range when only three would be considered valid input.

### Dependent variables

The value of a *dependent variable* is determined from other values supplied during the evaluation. A dependent variable can simply assume the value of another variable or response, or you may define a formula to calculate the value. When a value in a given variable must be based on the values in one or more other variables or responses, you write a formula to calculate that value.

For example, the area of a triangle must be calculated using its base and height. HEEDS MDO Solver can assign the base and height, based on your constraints, but it will not automatically calculate the area. So, to prevent HEEDS MDO Solver from assigning a non-logical area value, you would make the area variable dependent, and calculate it from the base and height variables.

Dependent variables can be used in all types of projects. Before you define a dependent variable, you must define the variables and/or responses it is dependent upon. During a HEEDS MDO run, these variables and responses must be assigned values before the value of the dependent variable can be determined.

Since the values for all variables in the design are written out before the analyses are run, your formulas can reference any variable that is used by the study, regardless of which analysis it is in. It does not matter if the analysis has been run.

If you have more than one analysis in a process, you can use dependent variables to pass values from one analysis to another. If, for example, you want the value in a response from one analysis to be used as the value for a variable in a subsequent analysis, you can assign that response to a dependent variable. You can also use the response in a calculation. Unlike variables, which can be used regardless of which analysis they are associated with, responses are only filled after an analysis is run. So the analysis that contains the response must be run before the analysis where you plan to use the value. Therefore, you must make sure that the analyses in that process run in the correct order.

When you have HEEDS PARALLEL enabled, HEEDS MDO will identify these cases and automatically reorder the analysis execution order by adding dependencies so that this condition is met. A warning is written out to the message file in this case.

### Constant variables

A *constant variable* is one whose nominal value does not change. You can also use a constant variable to vary a “known” value stochastically to test a design for robustness.

- *Absolute constants* are useful when you want to hold a value constant rather than allowing it to vary.
- *Stochastic constant* variables are used in robustness and reliability studies to test the effect of expected variations on design performance. They can also be used in parameter optimization studies to build robustness into a design. Stochasticity is not applied in DOE studies. So, if any *constant* variables are present in the study, only the baseline value is used.

*Absolute* numeric parameter variables can be used in all study types. Stochastic numeric parameter variables can be used only in parameter optimization and robustness and reliability studies.

### Text variables

Just like the *constant* variables, *text* variables can be used to assign a constant value (text string) to a variable instead of allowing it to be varied. No stochasticity can be defined with this variable type.

### Project responses

Responses are either designated values in the output file or values that are calculated from other variables and responses. The way a response is used depends on the type of project. For parameter optimization projects, responses are used either to evaluate a design (your *objectives* and your *constraints*) or to store values. The dependent variables and responses use such values, later in an evaluation of a design, or to monitor certain aspects of a design. The values could also be used in a formula. In HEEDS MDO, the latter are referred to as *prerequisites*.

In your parameter optimization studies, you will need to define one response for each objective and each constraint you want HEEDS MDO to use in determining the optimal design.

In DOE and robustness and reliability studies, responses do not have any designated specialized function in an evaluation. In DOE studies, the responses are used to determine how changes in variable values affect the designs. In robustness and reliability studies, HEEDS MDO determines the effect of expected deviations in designated variables on each response.

Responses can be extracted directly from an analysis output file or determined from a calculation based on project variables and/or other responses. Suppose, for example, two sides of a rectangle are project variables and the area of the rectangle is specified as a constraint. If the analysis code employed to evaluate the design does not provide the area of the rectangle as direct output, you can supply a formula to calculate the value.

## Tagging the files

To tell HEEDS MDO which values in the input and output files represent the variables and responses you have defined, you need to *tag* the files. Tagging an input file is merely a matter of linking the variables you have defined with the corresponding values you want to replace in the input file. When HEEDS MDO runs, it will replace these values for each design it evaluates.

The values you tag in the output file represent the locations from which the data will be read for calculations of the responses. Variables and responses are tagged on the *Tagging* page. There are three types of tagging:

- *Delimited marking* simply tags the values (based on row number, parsing delimiters, and position on the selected row) that represent the variables or responses.
- *Fixed width marking* simply tags the values (based on row number and column number) that represent the variables or responses. This is used when the adjacent fields are contiguous and not separated by any delimiters.
- *Scripting* uses a simple command-based language to determine the location of the values that represent variables or responses. Scripting provides a very flexible and dynamic tagging option that allows the tagging of locations in the file that may change from one design to the other, tagging of arrays, and everything that can be done with delimited or fixed marking.
- *Tagging with portals* allows users of certain supported software packages to tag variables and responses without requiring users to know exactly where the values are located in the input or output files.

After an evaluation is run, HEEDS MDO extracts the designated responses from the output file and uses them to guide an optimization search, to determine post-processing results, or to calculate values of other variables and responses.

## Exploration

Once you have completed the task levels of *Process Automation* (*Process*, *Parameters*, and *Tagging*) for your project, HEEDS can modify the files of your study, create new models, execute the files, and look at the results, all without user intervention. At this point you can move to the *Exploration* of your project, which includes the task levels *Study* and *Run*:

- *Study* includes tasks for you to define how HEEDS MDO will use your processes in a design exploration study. Here you identify the type and method of study you wish to conduct.
- *Run* includes tasks to run and monitor your study after the study definition for your project is complete.

## Study

Once you have the full information for *Process Automation*, you can move to the *Study* level tasks that include selecting the type of study to run and defining the study-specific data. The default study type is *Parameter Optimization*, and the default study method is *SHERPA – Weighted sum of all objectives*. From this point you can

review the variables for your study, add more variables if required, and evaluate how the variables are tagged. You can also review your responses and define which ones are objectives and which ones are constraints. When your study definition is complete, you are ready to *Run* your study.

Each project must have at least one study. A typical study is associated with one process. The variables and responses used by the study are those that are tagged in the input and output files used by the analyses in the process and any that are defined through a formula.

### **Study responses**

The responses used by a study are called the *study responses*. In parameter optimization studies, each study response that will be used as an objective or a constraint requires an additional definition to tell HEEDS MDO what to do with the response when HEEDS MDO evaluates the performance of the design. No further definition is required for DOE and robustness and reliability studies, as these studies are concerned only with the output value.

Study responses in parameter optimization studies can be of three types: *objective*, *constraint*, and *prerequisite*:

- *Objective responses* contain the values you are trying to maximize or minimize (your *objectives*, in other words). When you define an objective response, you specify whether you are trying to maximize or minimize the response. For example, if you were seeking to minimize the weight of a given object, “weight” would be defined as an objective response. You would be trying to minimize the objective because you are searching for the minimum value.
- *Constraint responses* contain the limiting factors for the search (your *constraints*). They allow you to codify the product requirements so designs that do not meet them can eventually be eliminated. When you define a constraint, you specify a limit and whether the response value must be less than or greater than that limit. A structural component, for example, might be constrained by size, stress, or displacement. In the response definition, you would specify the limiting value and indicate that the response must be greater than or less than that limit.
- *Prerequisite responses* are not directly used to evaluate how well a design performs. Instead, they are used in the formulas that are to populate dependent variables and/or calculated responses or just to monitor certain aspects of the designs without impacting the search. They do not have any attributes. No separate definition is needed for *prerequisites*. Any response not defined as an objective or constraint is treated as a prerequisite.

### **Study variables**

The variables used by a given study are called *study variables*. Parameter optimization studies and DOE studies require additional definitions. No further definition is required for robustness and reliability studies, as all necessary definition is done when the variable is created.

For continuous variables in parameter optimization studies, you can change the resolution. The resolution controls the “fineness” of the values that will be used within the specified range, and discretizes continuous variables for search methods that employ variable discretization. You may set this value as high or as low as you wish. If the range were between 0 and 10, a resolution of 11 would specify only integer values 0–10. A resolution of 101 would produce a finer set of values—0, 0.1, 0.2, 0.3, etc. The search algorithm selects its values from this set. Discrete variables do not require a resolution because they are specified as a set of explicit values, rather than a range.

When a continuous variable is defined as stochastic in a parameter optimization studies, it is assigned a stochastic distribution. However, it does not have to be used as a stochastic variable in your study. When you define your study variables, you can decide whether you want to treat the variable as stochastic in the analyses performed by that study. This choice is not available at the study level for other types of variables that are assigned stochastic distributions.

For DOE studies, you must select your factors from the variables defined for your study. Only continuous and discrete variables may be used as factors. The default values HEEDS MDO uses for the factors during a run are

determined by the DOE method in use for the study and the maximum and minimum values you define for the variable. You may optionally override these defaults to have more control over the values used.

## Methods

Each study uses a defined method to perform its design exploration functions. For parameter optimization studies, a proprietary search method called SHERPA (Systematic Hybrid Exploration that is Robust, Progressive and Adaptive) combines several different search methods simultaneously, adapting and refining them as the search progresses. By default, SHERPA optimizes the weighted sum of all objectives. Alternatively, you may choose the Multiple objective tradeoff study (Pareto front) option. The multiple-objective option performs a search for each objective in studies with multiple objectives (e.g., solving for both minimum weight and maximum stress). It can be used instead of the weighted sum when you want each objective evaluated independently to determine tradeoffs instead of finding the best possible design based on the weighted combination of all objectives. You can alternatively select one of several other methods if desired.

For DOE studies, the method determines how the design matrix is created. You may select the specific method you want to use or allow HEEDS MDO to select it for you.

For robustness and reliability studies, you simply select whether you want the values generated at random within your stochastic distributions or whether you want to use a “structured” sampling method.

For evaluation studies, you specify the designs that need to be evaluated. No search is performed in this study.

## Run

Whenever you save your project, HEEDS MDO creates a set of files that are needed for the execution of the run. You do not need to do anything with them to run your study, but you may be interested in their contents. The following files are created for all projects:

File	Description
<b>Definitions.in</b>	This file contains information about all the project variables defined in the project. There will be only one <b>Definitions.in</b> file for a study.
<b>Representation<math>n</math>.in</b>	This file contains the study variables that are being used by a particular representation. The $n$ represents the ID of the representation file. This number must be unique and can be any integer.
<b>Performanc<math>n</math>.in</b>	This file contains the information associated with the responses being used by a particular study. The $n$ represents the ID of the performance file and can be any integer. This number must be unique.
<b>Assembly.in</b>	This file contains information that connects the process data to the study data. There will be only one <b>Assembly.in</b> file for a project.
<b>Process<math>n</math>.in</b>	This file contains the information about the process and all the analyses defined in that process. The $n$ represents the ID of the process and can be any integer.
<b>StudyName.in</b>	This file controls the execution of a HEEDS MDO Solver run for a study. The name of the file is the same as the name of your study.
<b>M_inputfilename</b>	This file contains tagging information related to marking for the specified file.
<b>P_inputfilename</b>	This file contains tagging information related to the portal definition for the specified file.
<b>response.hs</b>	This is a script file created for reading a response tagged using scripting, where <i>response</i> is the name of the response.

For each study, HEEDS MDO creates a folder inside the project folder (this is the directory where you saved your HEEDS MDO project) named using the following convention: ProjectName\_StudyName. These files are placed in your study directory.



**Note:** For multi-processor runs, some additional run files are available. For more information, see the *HEEDS MDO online help*.

## Running the study

You can run a study either from HEEDS MDO or from a command line prompt. In the examples in this manual, we will be running from within HEEDS MDO. Instructions for running from the command line can be found in the *HEEDS MDO online help*.

During the run, HEEDS MDO displays runtime plots that let you monitor the progress of the run. Post-processing plots show the results of the run. Post-processing plots for parameter optimization and robustness and reliability studies can be viewed while the study is running, as well as at the end of the run, so you can always see what the end result would be if you were to stop the run. DOE-specific post-processing plots are not available until the entire run is completed.

HEEDS MDO displays many types of post-processing plots, some of which are described below. For a full description of the plot types available for all study types, please refer to the *HEEDS POST online help*.

The types of plots available depend upon the type of study in the project.

While HEEDS MDO runs, it creates a number of working files. The ones of interest are as follows:

File	Description
<b>HEEDSn.plot</b>	Contains the results for the “best” designs found during the optimization study. HEEDS MDO Solver starts by writing the first design to the file. Then each time it finds a design that is better than the last, it adds that design to the file. At any time during the run, the last entry will be the best design found so far. In addition to the best designs, HEEDS MDO Solver also writes out any unsuccessful designs (designs for which the Solver was unable to extract the results) to this file.
<b>HEEDSn.gph</b>	Contains the performance rating, the values of the project variables, and the values of the responses for all “best designs.”
<b>HEEDSn.res</b>	Contains the results for every evaluation that is performed during the run. It includes the performance rating, the values of the project variables, and the values of the responses.



**Important Note:** While the run is in progress, these files should only be opened in a text editor that does not lock the file from being written (e.g., Notepad, WordPad, Vi, Gvim, Textpad). Do NOT use Excel or other tools with similar behavior, or this will stop the run.

# The HEEDS MDO Environment

Your project is defined in HEEDS MDO. HEEDS MDO contains a set of tabs, one for each step in the project setup.

## Starting HEEDS MDO

### Windows

1. Click **Start** at the lower left corner of your screen.
2. Select **All Programs** from the *Start* menu.
3. Select **HEEDSMDO** from the **Programs** menu. A submenu will appear.

The start screen appears with the **File** tab selected:



**Hint:** If you are running in Windows, you can place a shortcut to HEEDS MDO on your desktop. Then you won't need to start it from the **Start** menu each time. The default path to HEEDS MDO is: **C:\HEEDS\MDO\version\Win64\HEEDSMDO.exe**, where **version** is the version of HEEDS MDO that is installed.

**Note:** If you are running a single-user system and you did not set up the license manager to start the server automatically at power up, you will need to start the license server first. Refer to the installation guide for instructions.

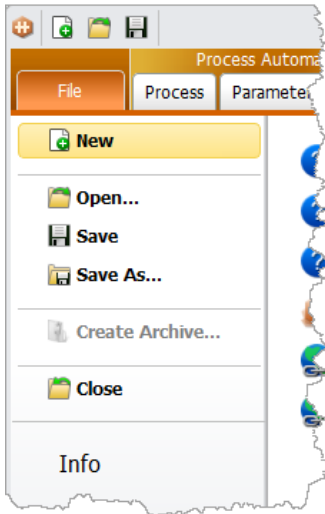
### All other platforms

1. Open a terminal.
2. Type **heedsmdo.sh**.

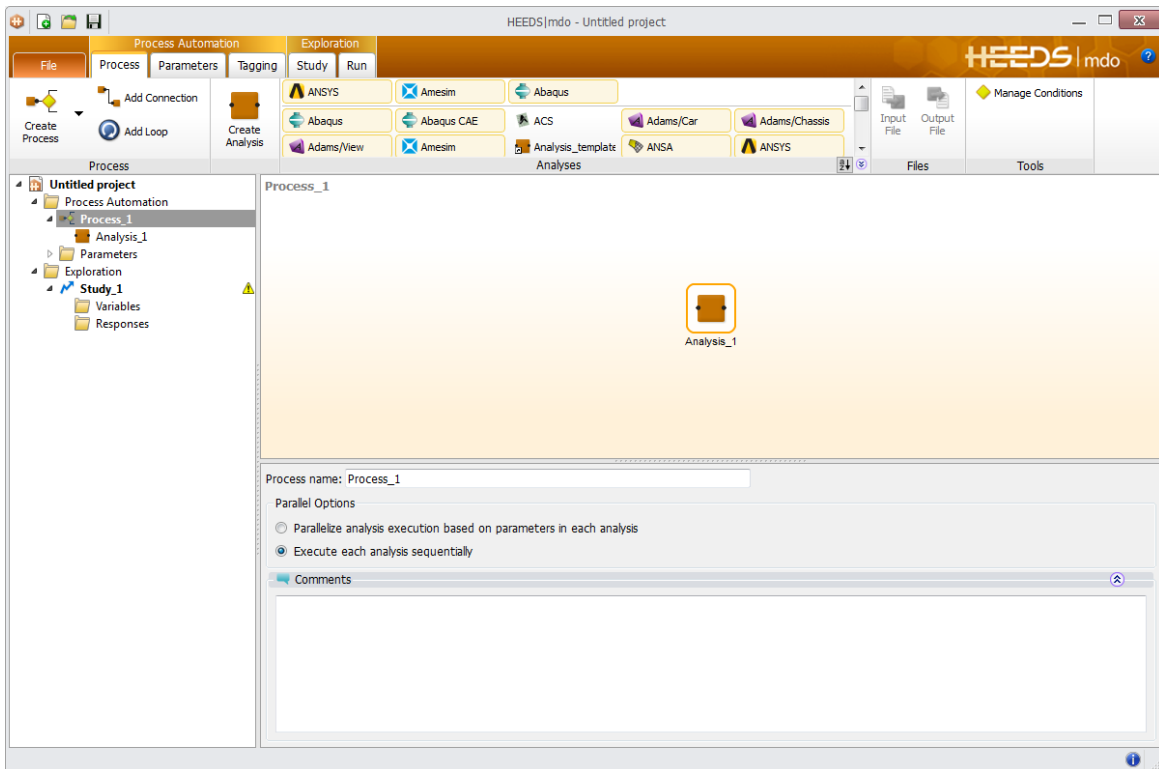
## Starting a New Project

In order to examine the screen, we need to have a project loaded. Because we don't have any existing projects yet, we will start a new one.

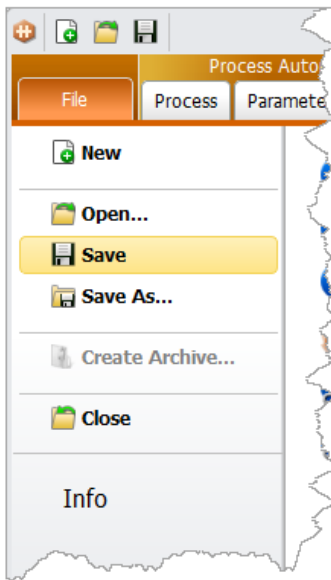
1. On the **File** tab, select **New**:



A new screen appears:



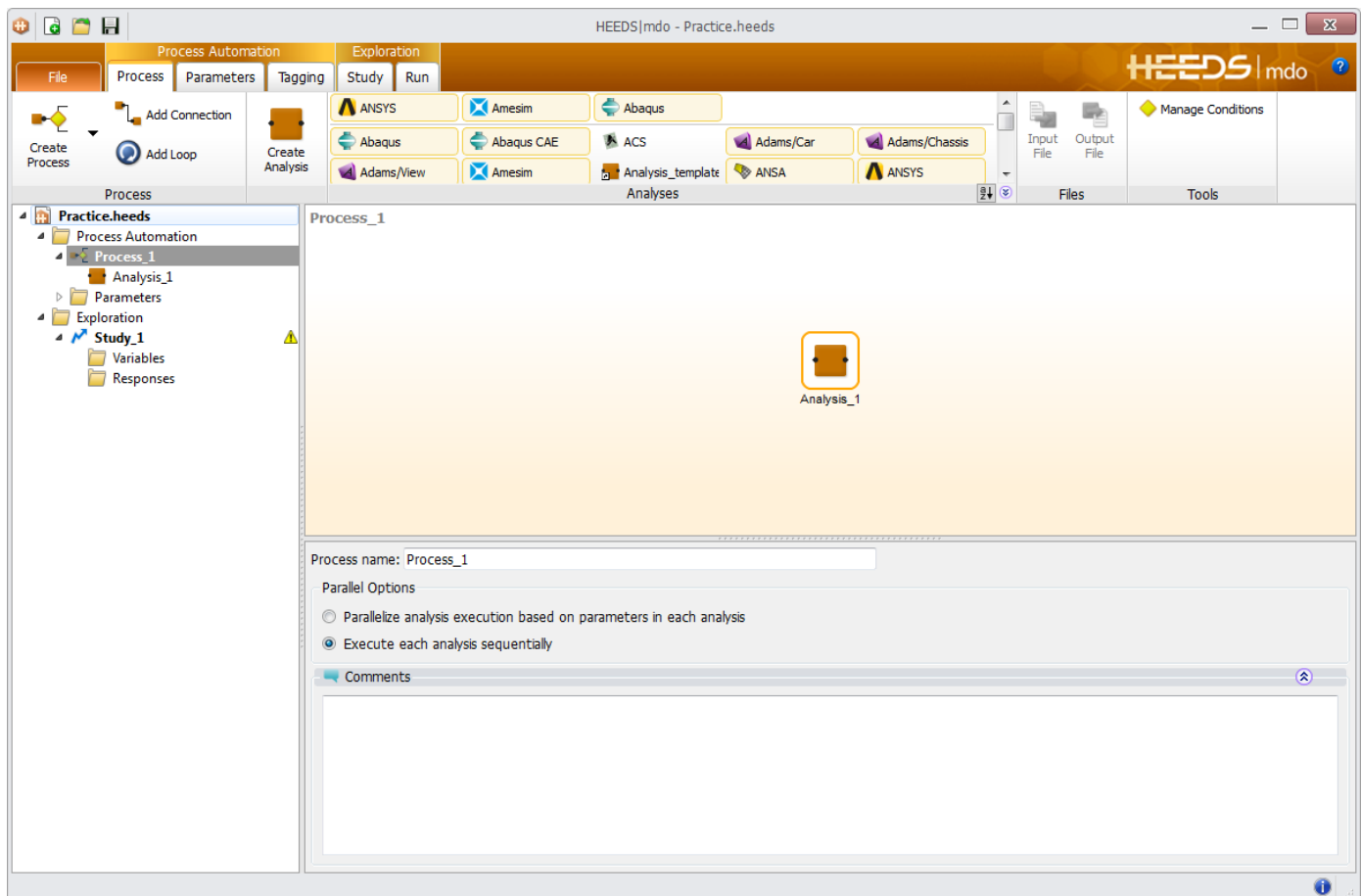
2. On the **File** tab, select **Save**:



The **Save Project As** dialog box appears.

3. Navigate to the directory where you'd like all of your project files to be, and save your new project as **Practice**. The screen returns with your current project in the title bar.

## The HEEDS MDO Layout



The screen is laid out on a set of tabs organized inside of two high-level tasks: *Process Automation* and *Exploration*. Click a tab and the page for that tabbed section will appear. Each tab contains all the features you need to perform one of the main steps in setting up and running a HEEDS MDO project.

Tab	Description
<b>File</b>	Used to create a new process or open a current process.
<b>Process</b>	Used to define the processes and analyses for the project.
<b>Parameters</b>	Used to define the project variables and responses and their properties.
<b>Tagging</b>	Used to tag variables and responses in the input and output files.
<b>Study</b>	Used to define the study type and method of the project.
<b>Run</b>	Used to run and monitor the project. Gives option to open HEEDS POST.

To get a quick overview, click on each tab and take a look at the page contents. You will probably see some correlation between the screens and the parts of a HEEDS MDO project that we described in the previous chapter.

We will give you specific instructions for setting up a project in our tutorial examples. In this chapter, we will concentrate on the way the screens function in general.

## Entering Information

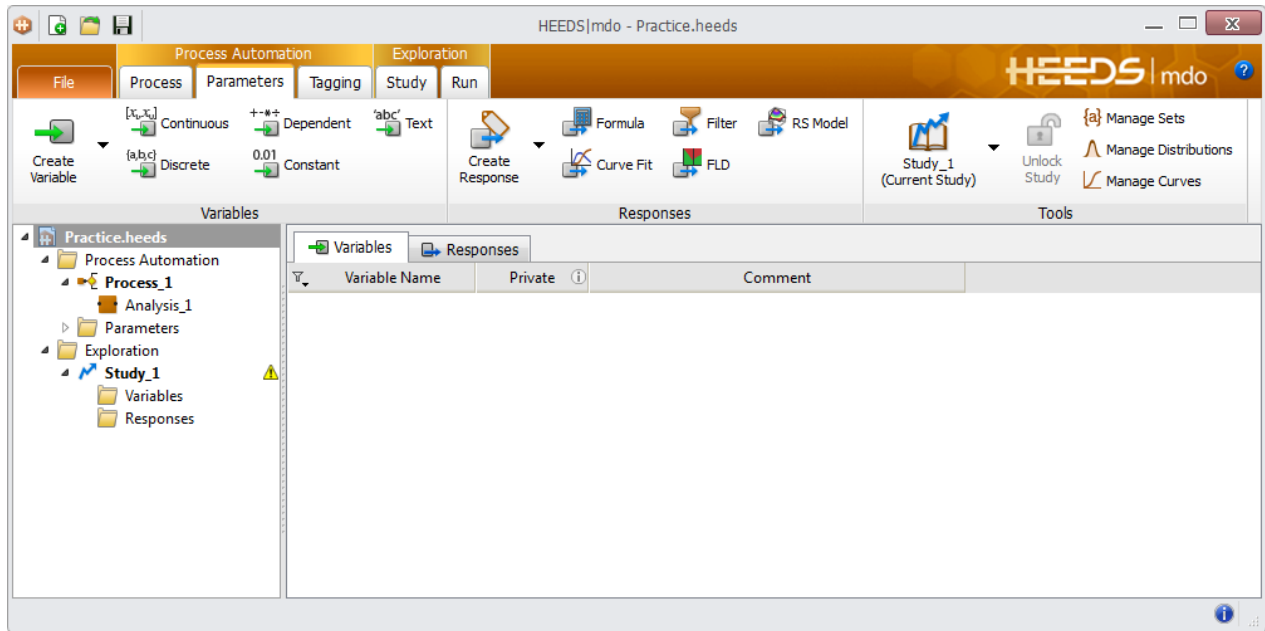
---

Each tab has its own specific set of instructions, so we won't explain them all here. We will save that for the examples in Part 2. However, there are some things that are common across-the-board that may be helpful to know before you start trying to define a project.

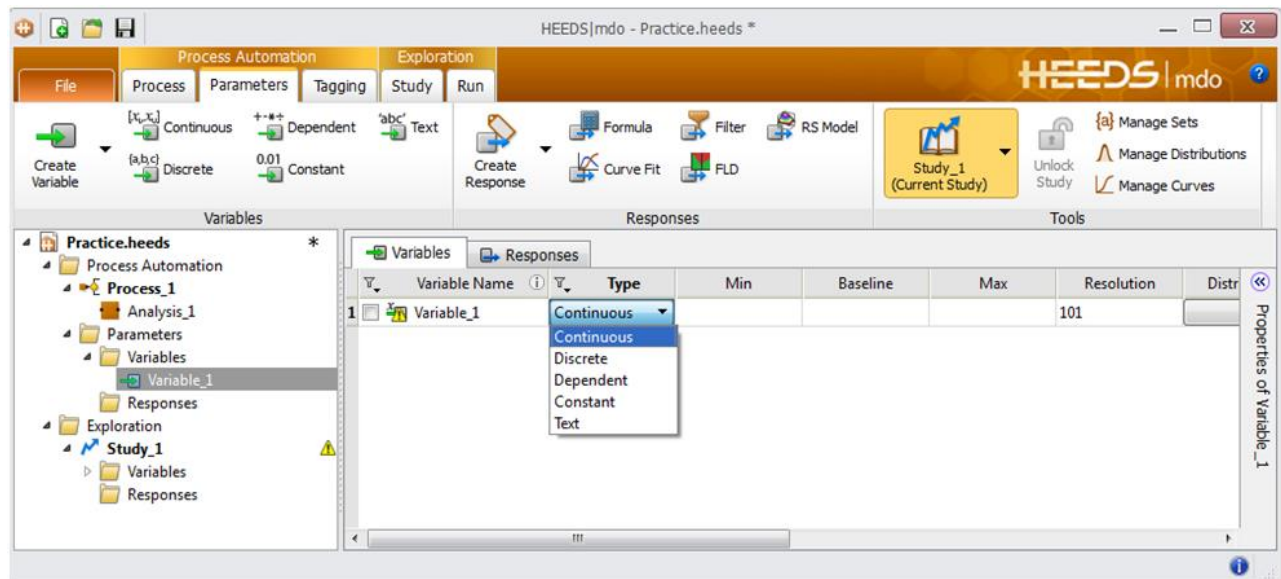
On the various pages beneath the tabs, you will see some of the input controls (tools, text boxes, trees, etc.) that you are already familiar with. Because you know how to use them, there is no point in belaboring them in this chapter. However, there are two distinct features that are used in HEEDS MDO that may not be familiar. We will examine them on the *Parameters* page, so click the **Parameters** tab now.

## Adding Variables


1. You can add items to your study using the tools on the *Ribbon*. To add variables to your study, you can select from these types: **Continuous**, **Dependent**, **Discrete**, **Constant** or **Text**. After you select a variable type, you will see it added to the *Variables Manager* on your screen. The screen below shows that you added a *Continuous* variable:

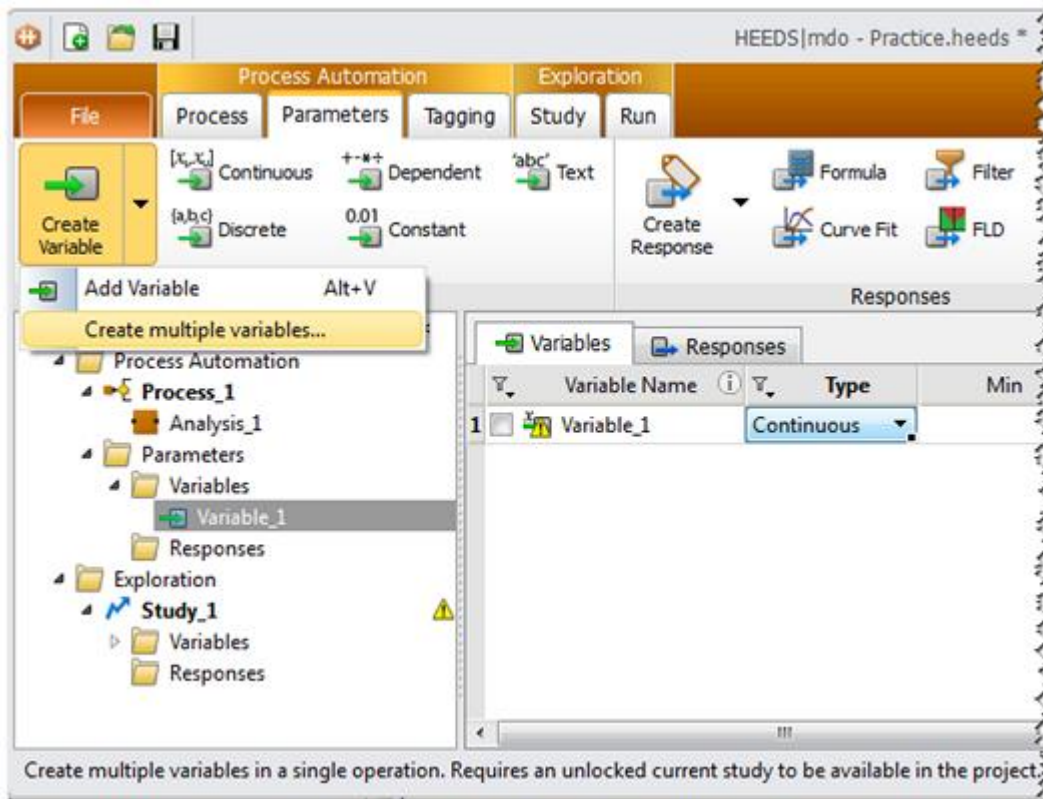


2. Alternatively, you can add a new variable by clicking **Create Variable** on the *Ribbon*. A variable will appear in your *Variables Manager*. When you add a variable in this way, you will need to define the type by selecting the **Variables** tab and then choosing a type from the list:




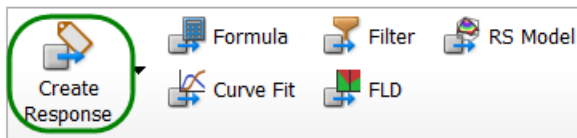
3. More actions can be initiated by clicking and right-clicking in the table. You can add, delete, copy, etc., directly from the table.

- To add multiple variables at once, you can open the dropdown on the  **Create Variable** tool, and select **Create Multiple Variables**:




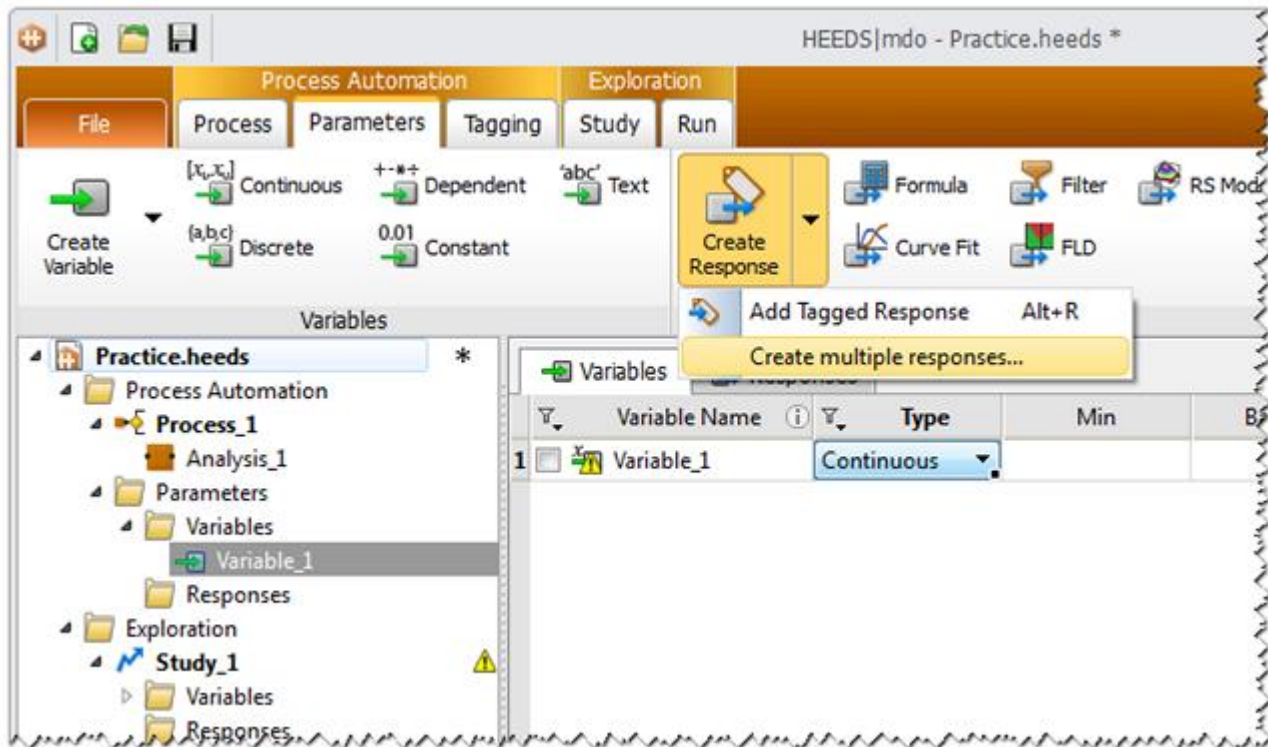
## Adding Responses

- To add responses to your study, select from these types on the *Ribbon*: **Formula**, **Filter**, **RS Model**, **Curve Fit**, or **FLD**. After you select a response type, you will see it added to the *Responses Manager* on your screen.
- Alternatively, you can add a new response by clicking  **Create Response**:



A response will appear in your *Responses Manager*. When you add a response in this way, you can define the response type by clicking the arrow inside of the **Source** box and then selecting a response type from the list. The default type is *tag*.

3. To add multiple responses at once, you can open the dropdown on the  **Create Response** tool, and select **Create Multiple Responses**:



4. You can define *Variable and Response attributes* using these tools on the right end of the *Ribbon*: **Manage Sets**, **Manage Distributions**, and **Manage Curves**.




**Note:** In this chapter, we focus on how information is entered and how the screen features work. We are not concerned with the actual values that are being entered or how they relate to a real HEEDS MDO project. We will use actual values in the examples in Part 2.

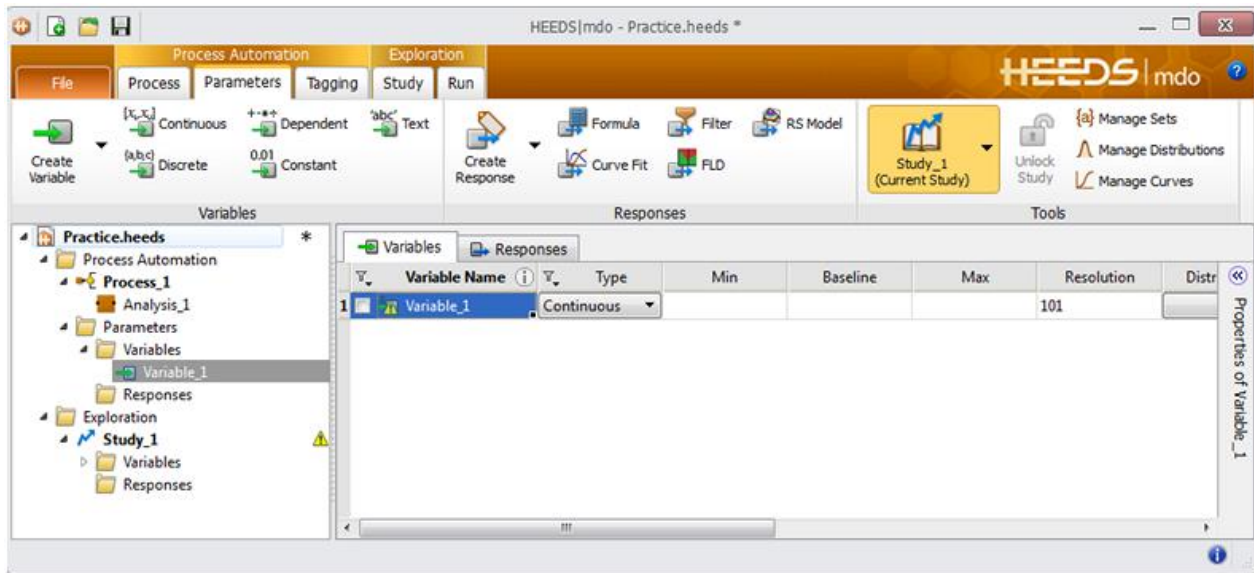
## Adding and removing items


You can add items by clicking tools on the *Ribbon* or by right-clicking and selecting through the menu options in the tables. You can delete by selecting items and pressing **Delete** on your keyboard or by right-clicking and selecting **Delete** through the menu options in the tables.

### Adding items

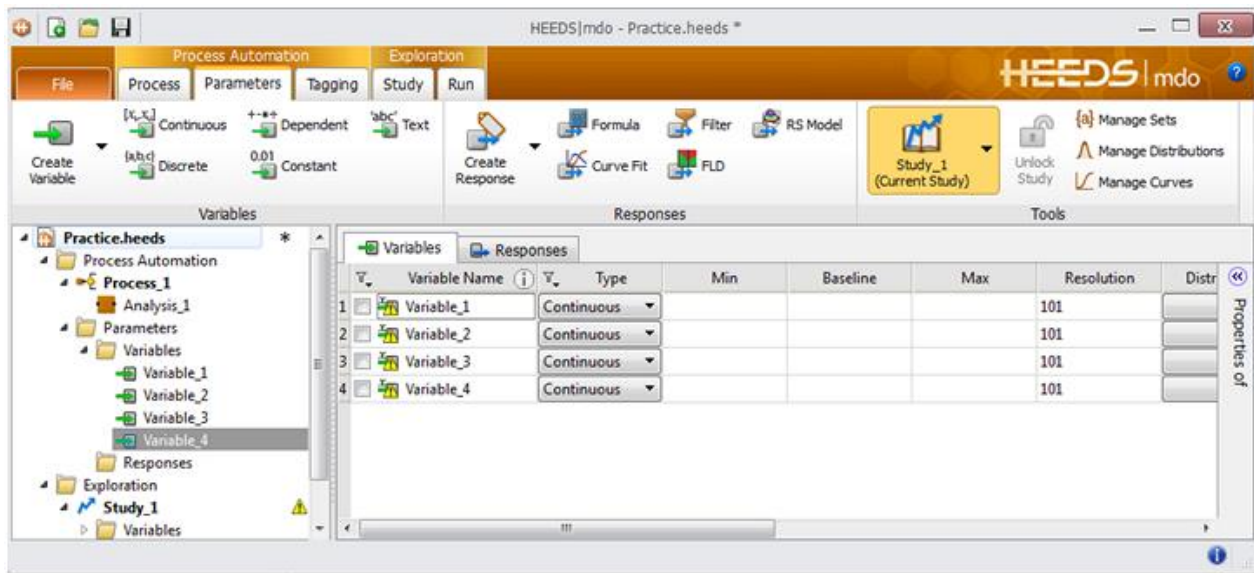
To add an item, click the tab associated with where you want to add your item. Locate the item on the *Ribbon*, and then click the item. Try the following:

1. Click the **Parameters** tab to display the *Parameters* page, and then click the  **Continuous** variable. The *Variables Manager* appears with a continuous variable added:



2. Add three more continuous variables to the table. You can add them in a variety of ways:
  - Click  **Continuous** on the *Ribbon* three more times or
  - Right-click on the table and select **Add Variable** from the context menu or
  - Open the dropdown on the **Create Variable** tool. Select **Create Multiple Variables**, and then follow the prompts on the *Wizard*.

When you are finished, your variables should look like this:



## Removing items

To remove an item from a table, click the item (or multiple items) to select it. Press **Delete** on your keyboard or right-click and select the menu option to **Delete**.

1. Select the box **variable\_2**, then right-click on the box and select **Delete**.
2. Click **Yes** in the prompt and the selected variable is removed from the table.

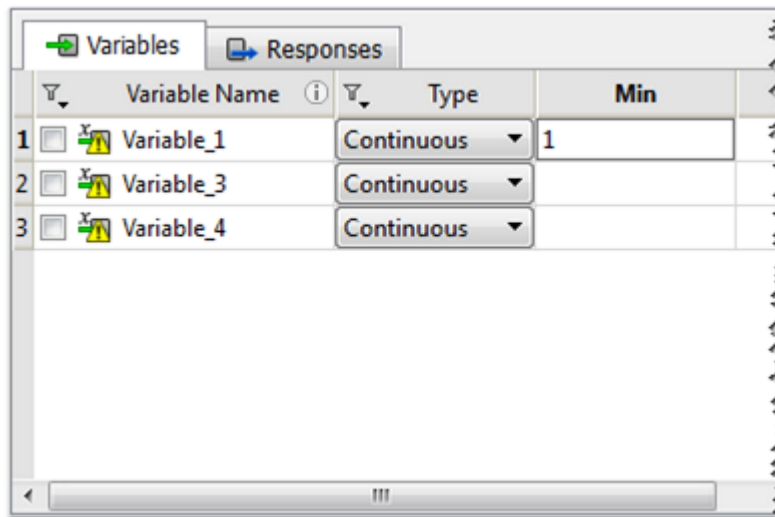
## Entering values


Each cell in a table row contains a field for entering a value. These fields will either be text boxes or list boxes.

### Text box cells

You can click cells to select them and double-click to enter values in them:

1. Click the **Min** cell for **variable\_1**. The cell darkens to show that it is selected. Click in the cell, and an insertion point appears in the box.
2. Enter **1** in the cell:

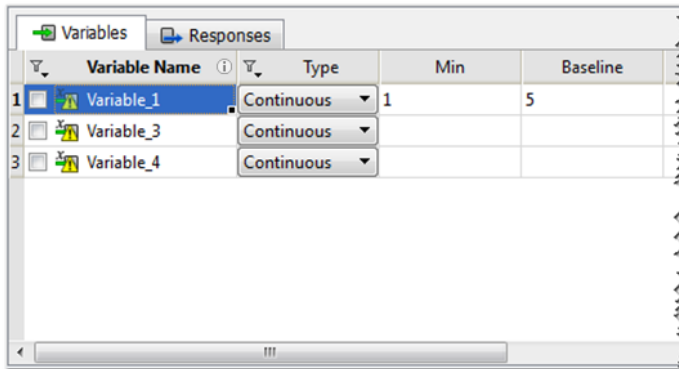


3. You can use the **Tab** key to move from cell to cell. Press **Tab** and the selection box moves to the **Baseline** cell. Enter a **5** in the cell.
4. Click the cell to the left of the **Baseline** value **5** that you just entered. The cell with **Min** value **1** one should be selected.
  - When you select a cell, the other cells in that row are also selected by association. The information related to these cells is available on the panel to the right of the table. In this case, the **Min** value is selected and appears in the panel along with the other values associated with it.
  - Click  to expand or minimize the panel. You can enter values on the panel the same as in the table.
5. To deselect a cell, click in the blank area beneath the table. The cell deselects and the information in the panel clears.

## Replacing a value

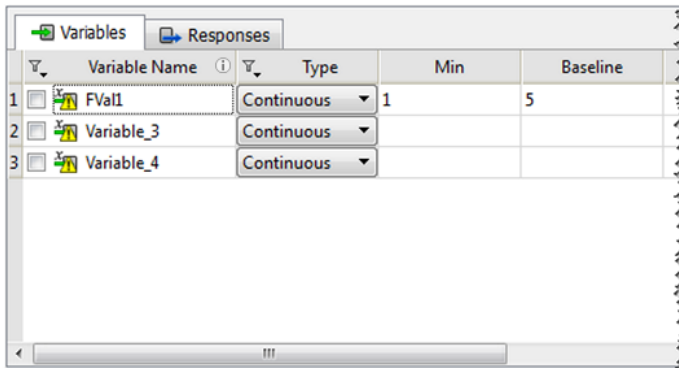
To replace an existing value in a cell, click the cell and enter the new value:

1. Click on **Variable\_1**. The current value will be highlighted:



	Variable Name	Type	Min	Baseline
1	Variable_1	Continuous	1	5
2	Variable_3	Continuous		
3	Variable_4	Continuous		




2. Change the name of the variable to **FVal1**, then click below the table to deselect the line. The new name will appear in the line:




	Variable Name	Type	Min	Baseline
1	FVal1	Continuous	1	5
2	Variable_3	Continuous		
3	Variable_4	Continuous		


## Copying and pasting


You can copy and paste values in a table with standard Windows keyboard keys. This is handy when you are entering values that are similar or identical:

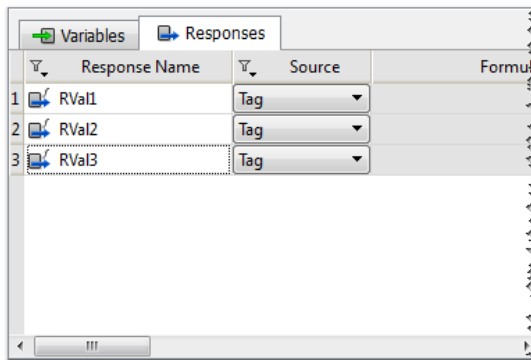
1. Select **FVal1**.
2. Press **Ctrl + C** to copy the highlighted value.
3. Press  to move to **Variable\_3**, and then press **Ctrl + V** to replace the highlighted value with the copied text.
4. Press  to position the insertion point at the end of **FVal1**. Then back space and enter **2** to change the value to **FVal2**.
5. Press  to move to **Variable\_4**. Press **Ctrl + V** to paste your copied value again, and then change the value to **FVal3**.

## Adding and naming responses

You can add responses with the  **Create Response** tool on the *Ribbon* of the *Parameters* page. You can also add responses by right-clicking in the table and selecting **Add Response** from the context menu:

1. On the *Ribbon*, click  **Create Response**. A project response appears in the *Responses* table on your screen.
2. Add two more responses:



- Click  **Create Response** again. A second project response appears in the table.
  - Move your cursor to the table and right-click. Select **Add Response** from the context menu. A third project response appears in the table.
- Double-click on the first response in the **Name** column and change the name from **Response\_1** to **RVal1**. Click below the table to deselect the cell. The name of the response appears as **RVal1** in the table.
  - Select **RVal1** and then right-click and select **Copy** from the context menu to copy the highlighted value.
  - Click **Response\_2**, then right-click and select **Paste** from the context menu to replace the highlighted value with **RVal1**.
  - Click at the end of the field, and then back space and type **2**.
  - Now that you know how, paste the copied value into **Response\_3** and change the **1** to a **3**. Your responses should look like this:

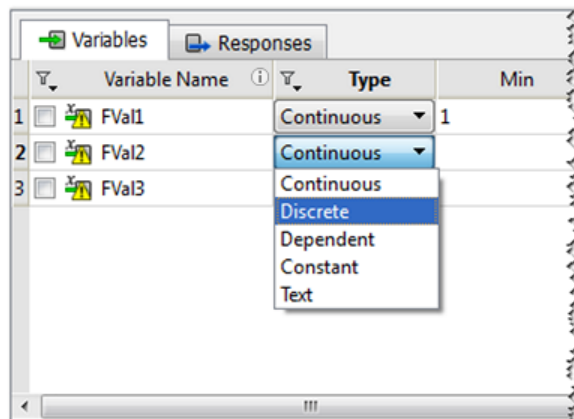


	Response Name	Source	Formula
1	RVal1	Tag	
2	RVal2	Tag	
3	RVal3	Tag	

## List boxes

List boxes in tables work just like drop-down lists in dialog boxes:

- Click the **Variables** tab.
- Click the **Type** cell for **FVal2**. You will see an arrow  which indicates a list box.
- Click the  and select **Discrete** as the **Type** value:



	Variable Name	Type	Min
1	FVal1	Continuous	1
2	FVal2	Continuous	
3	FVal3	Continuous	

## Selective activation

HEEDS MDO activates cells and screen features on an “as-needed” basis. If a cell or feature is not applicable, it will not be active.

## Table cells

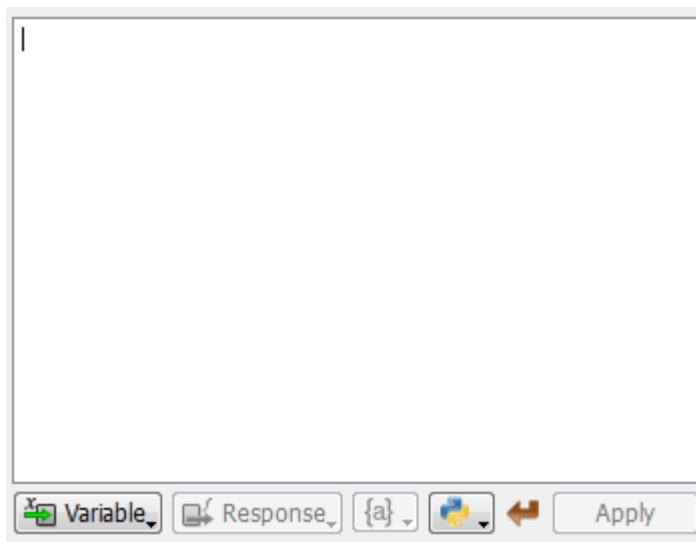
To examine this feature, let's start with table cells:

1. Click the **Min** cell for **FVal2**. Notice that you can select the cell, but that you cannot enter a value in the cell because **Min** is not a valid value for *Discrete* variables. The same is true for **Max**.
2. Click **Set** for **FVal1**. Note that you cannot enter a value because **Set** is not applicable to *Continuous* variables.
3. Click **Set** for **FVal2**. A list box appears because **Set** is a valid attribute for this variable. The list is empty, however, because no discrete sets have been created.

## Screen features

Like table cells, screen features that are not applicable remain inactive. The *Formula Definition* editor has been absent from the screen up to this point, because it has not been applicable to any of our variables:

1. Select **Dependent** as the **Type** for **FVal3**. The editor immediately appears in the lower right corner because dependent variables require a formula:



2. Select the **FVal1** table row. The editor is once again hidden.
3. Reselect **FVal3**, and the editor is redisplayed.

## Dependent lists

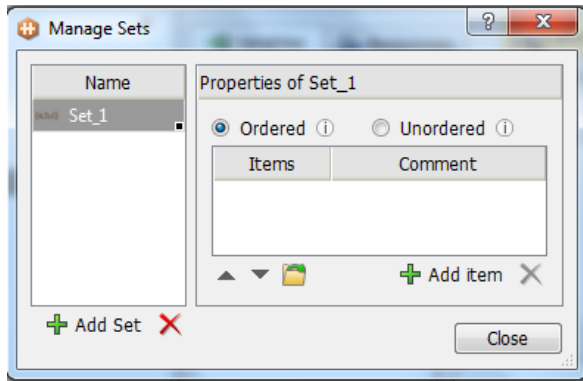
Some lists are dependent on data entered in other areas. Such lists will remain inactive until the required data are entered.


Click the **Set** list for **FVal2**. Notice the message displayed reads: **No items available**. This message appears because the list is associated with a designated discrete set that has not yet been defined. You can add and define a discrete set using the *Sets Manager*.

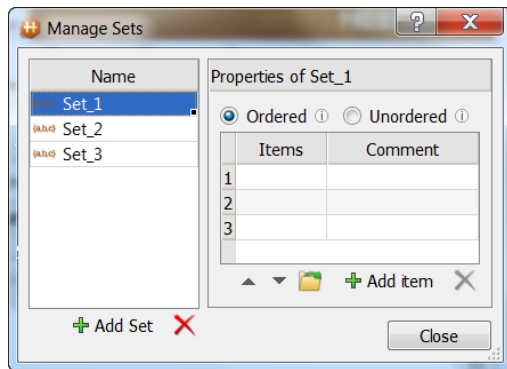
## Using the sets manager

1. To add a discrete set to your variable attributes, click **{ } Manage Sets** on the *Ribbon*. The *Sets Manager* appears.

2. Click **+ Add Set**. **Set\_1** appears:



3. Click **+ Add Set** two more times to create two more sets.
4. Close the *Sets Manager* and move your cursor to the **Set** column for FVal2 in the table.
5. Click the  and select **Set\_1** from the list box. **Set\_1** appears in your table.
6. Open the *Sets Manager* as you did above and click **Add item**. A table, **Properties of Set\_1**, appears with **Item\_1** included.
7. Add two more items. Three items now appear in the table:



8. Deselect **Set\_1**. Notice that the items disappear. Reselect **Set\_1** and the items reappear.
9. Click **Set\_2**. There are no values for *Items* because we have not defined this set yet.
10. Add one item to **Set\_2**. The item appears in the table as a value for **Set\_2**.
11. Select **Set\_1**. The list for **Set\_1** appears. Reselect **Set\_2** and you will see the list for **Set\_2**.
12. Close the *Sets Manager*.




**Note:** If you click in the **Set** field for FVal2, you will now see a list because we have created three discrete sets.

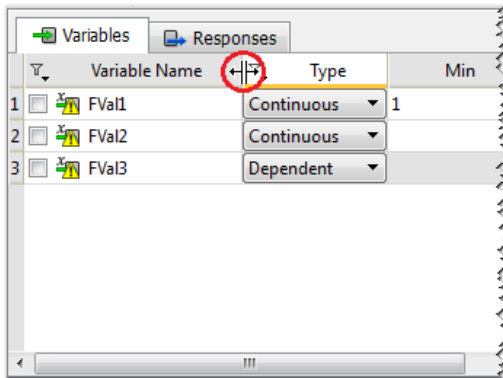
## Viewing Table Data

Not all tables can fit on the screen in their entirety. If you look at the *Project Variables* window, you may notice that the table “hangs off” the edge. To view the rest of the table, you need to use the scroll bar at the bottom of the screen.

## Changing the column width

If you want to see more of your table at once, you can change the width of the table columns:

1. Click and hold the line that separates the columns in the table header. An arrow icon  appears:



2. Drag the mouse pointer to the left or right to change the column width.


## Changing the window height

When two or more tables appear on your screen, you can change the window space between them with the horizontal *splitter* (the bar that separates two adjacent windows):



- Drag the pointer up or down to change the height of the window.



**Note:** Not all windows have moveable splitters between them. If the  does not appear, the splitter is not moveable.

## Saving and Retrieving HEEDS MDO Projects

To save and retrieve your projects, you will need to use the menu under the **File** tab.

### Saving projects

You can save your project at any stage of development. We can save our current project, even though it is not really a project:

1. Click the **File** tab, and select **Save** from the menu options.
2. The file will be saved with the name you gave it: **Practice.heeds**.

## Exiting from HEEDS MDO

After you load a project, it remains open until you either load another project or exit from HEEDS MDO.

Open the **File** tab again, and select **Exit**. HEED MDO will prompt you to save changes with this message: **The following projects contain modified data.** You are then prompted to save the project (**Save**), exit without saving (**Don't Save**) or cancel the save operation and return to HEEDS MDO (**Cancel**).

This message reminds you to save if necessary before you exit.

## Retrieving projects

You can retrieve existing projects using the *File* menu. When you use the **Open** option to retrieve a project, the current project you are working remains open and can be seen on the tree menu. We recommend that you always save your current project first before opening a new or existing project:

1. Open HEEDS MDO again. Click **File** from the tab bar, and then select **Open**.
2. Navigate to the *Examples* folder and open **Practice.heeds**.
3. Select **Open** on the *File* menu once again. Navigate to the directory where you have your HEEDS projects and select a project. After you select a project, HEEDS MDO returns with the project on your screen. The tree menu on the left of your screen lists the name of this project below any other projects that are opened.

## Your Next Step

---

Now that you have a general idea of how HEEDS MDO works and what goes into a HEEDS MDO project, you are ready to create a project. The next part of this guide contains tutorial examples that will guide you through some of the basics of defining and running a project. The tutorials progress from easy to more advanced, and each assumes that you have completed the previous tutorial, so be sure to move through them in order.



# Part 2

## Example Problems



# Example 1: A Simple Function Problem

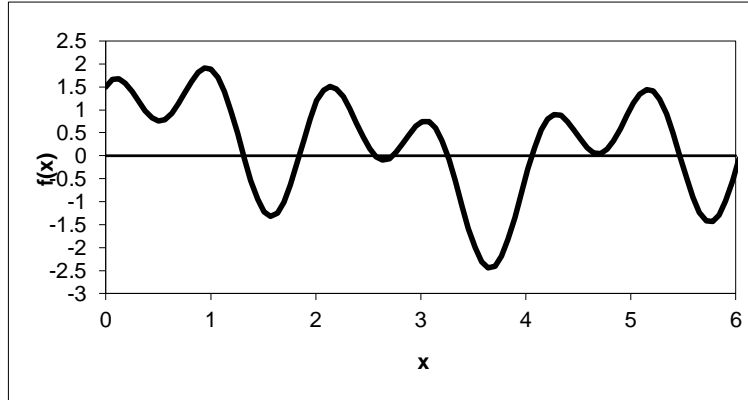


Fig. 1. Function representing the design space of an optimization problem

In our first example, we will illustrate the basic steps involved in setting up and running a HEEDS MDO parameter optimization project. Fig. 1 represents the multi-modal design space of a non-trivial physical problem with a single variable. Our purpose is to find the global minimum of the function  $f(x)$ .

## The problem

The current optimization problem is defined as:

**Minimize:**  $f(x)$   
**By varying:**  $0 \leq x \leq 6$

The function  $f(x)$  is represented as follows:

$$f(x) = \cos(x) + \sin(3x) + \cos(6x) + \begin{cases} 1.5/2 * x - 0.5 & \text{for } x < 2 \\ -1.5/(2\pi - 2) * x + 1.7 & \text{for } x \geq 2 \end{cases}$$

The location of the global minimum is  $x = 3.661925$  at which  $f(x) = -2.44985$



**Note:** In the present example, the function  $f(x)$  is defined analytically, but it represents the unknown design space for every problem.

## The baseline design files

The files for our example are found in the *Function* folder in the *Examples* directory:

- The default path for Windows is: **C:\HEEDSMDO\Version\Examples\Function**
- The default path for Linux and UNIX is: **/opt/HEEDS/MDO/Version/Examples/Function**

If you installed HEEDS MDO in a different directory, look for the files in that directory instead. There are three files:

File Name	Description
<b>function.in</b>	The input file.
<b>function.out</b>	The output file.
<b>function.exe</b> (Windows) <b>function</b> (Linux)	The tool that will be used to execute the analysis in Windows.

## Step 1: Starting a New Project

Before you create a project, we recommend that you create a directory on your system specifically for your HEEDS project. Do this for each HEEDS MDO project you create. Be sure to choose a location for which you have write permissions.

1. Click the **File** tab to display the *File* page. Select **New**. A new project appears on your screen. You can also start a new project by clicking the **Process** tab after you launch HEEDS MDO.
2. Click the **File** tab, and then select **Save**. The *Save Project As* dialog appears and prompts you to browse to a location to save your project.
3. Follow the prompt and navigate to the directory that you have set up to save this project.
4. Enter **FunctionEx** as the project name, and then click **Save**. The file will be saved with a **.heeds** extension.

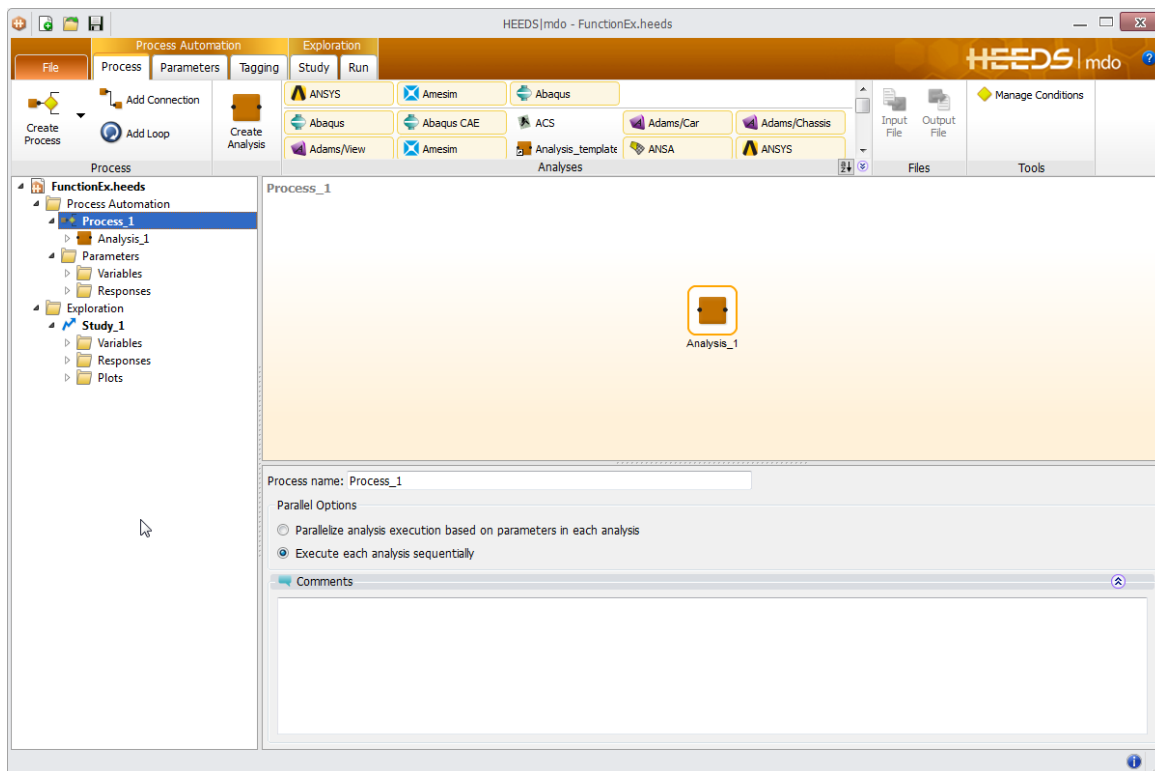
## Step 2: Defining the Process

As we have discussed in the introduction to this guide, creating a HEEDS project involves two high-level tasks: *Process Automation* and *Exploration*. Steps one through four will take you through the *Process Automation* tasks of your project: *Process*, *Parameters*, and *Tagging*. Steps five through eight will take you through the *Exploration* tasks: *Study* and *Run*.

### Identifying the analyses

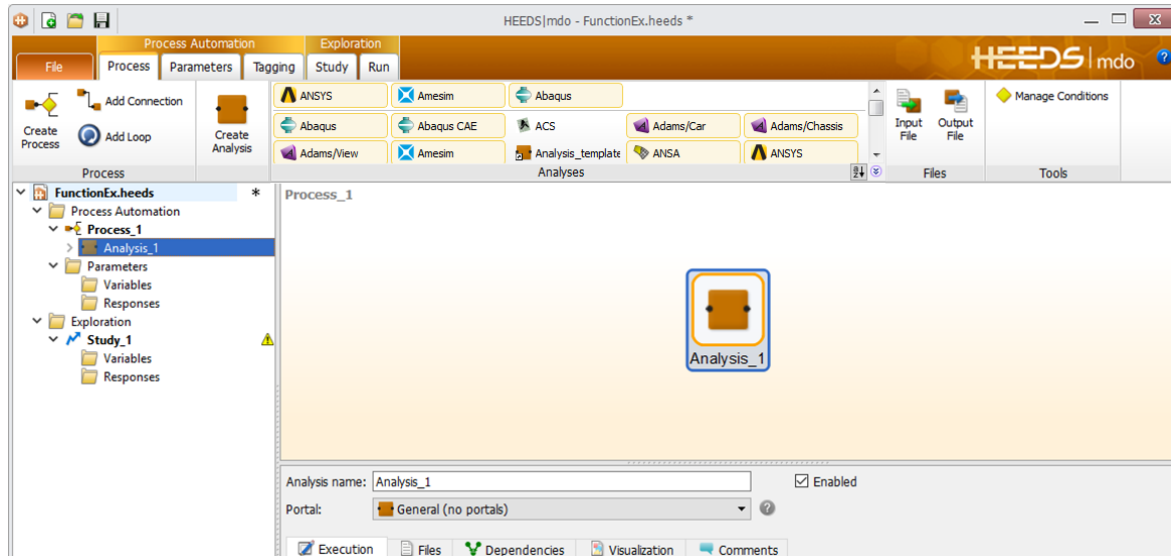
Our first step is to identify the analyses of our *Process*. Our example is a simple one. It has a single analysis with one input file and one output file. Identifying the analysis is simply a matter of linking the input file, **function.in**, and the output file, **function.out**, with the tool that was used in the original analysis, our executable, **function.exe**.

1. Click the **Process** tab and get familiar with the screen that you see.

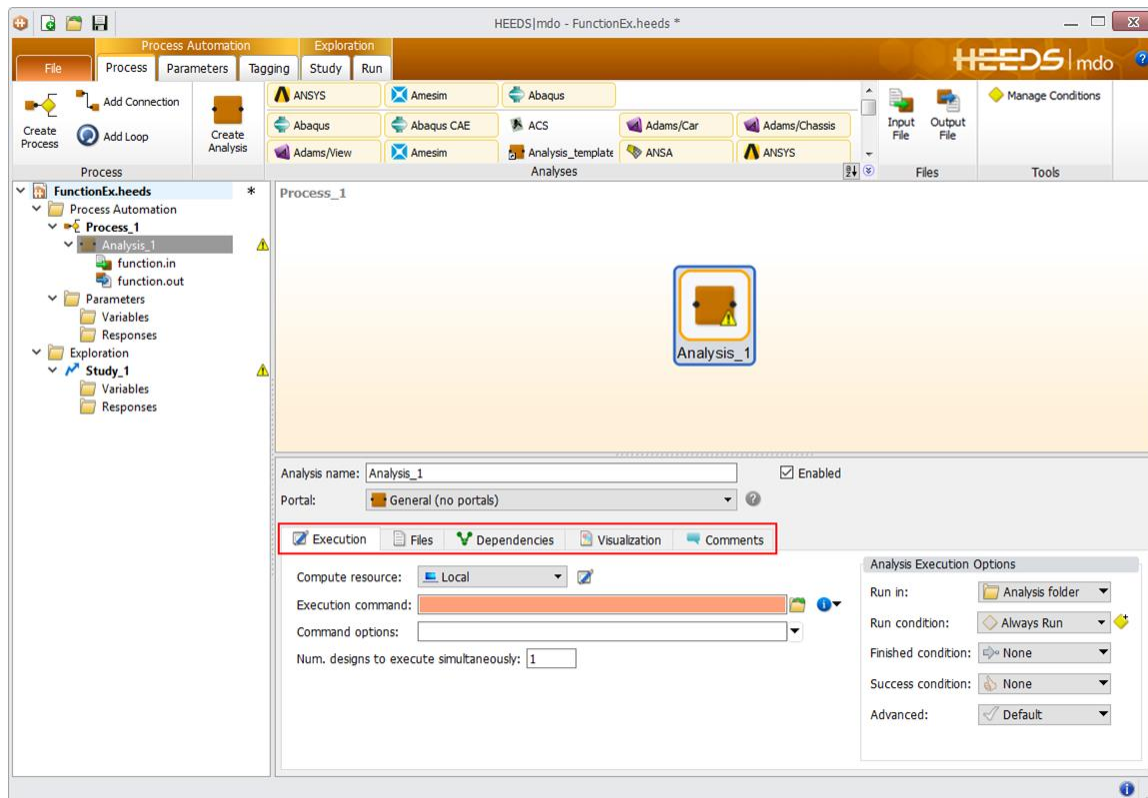


2. Click **Process\_1** in the tree menu. The *Process View* appears on your screen and **Process\_1** darkens on the tree to show that you have selected it.

- The default project setup consists of a single analysis (**Analysis\_1**) in a single process (**Process\_1**). Our function example will use the default setup.
  - If your project required more than one analysis, you could add the analyses with the **Create Analysis** tool on the *Ribbon*. Alternatively, you could right-click on **Process\_1** in the tree menu, open a context menu, and select the option to add an analysis from the menu.
3. On the *Process View*, click the **Analysis\_1** tool. The *Analysis Manager* appears and **Analysis\_1** darkens on the tree to show that you have selected it:




The *Analysis Manager* opens a set of tabs where you set the properties for the analysis.



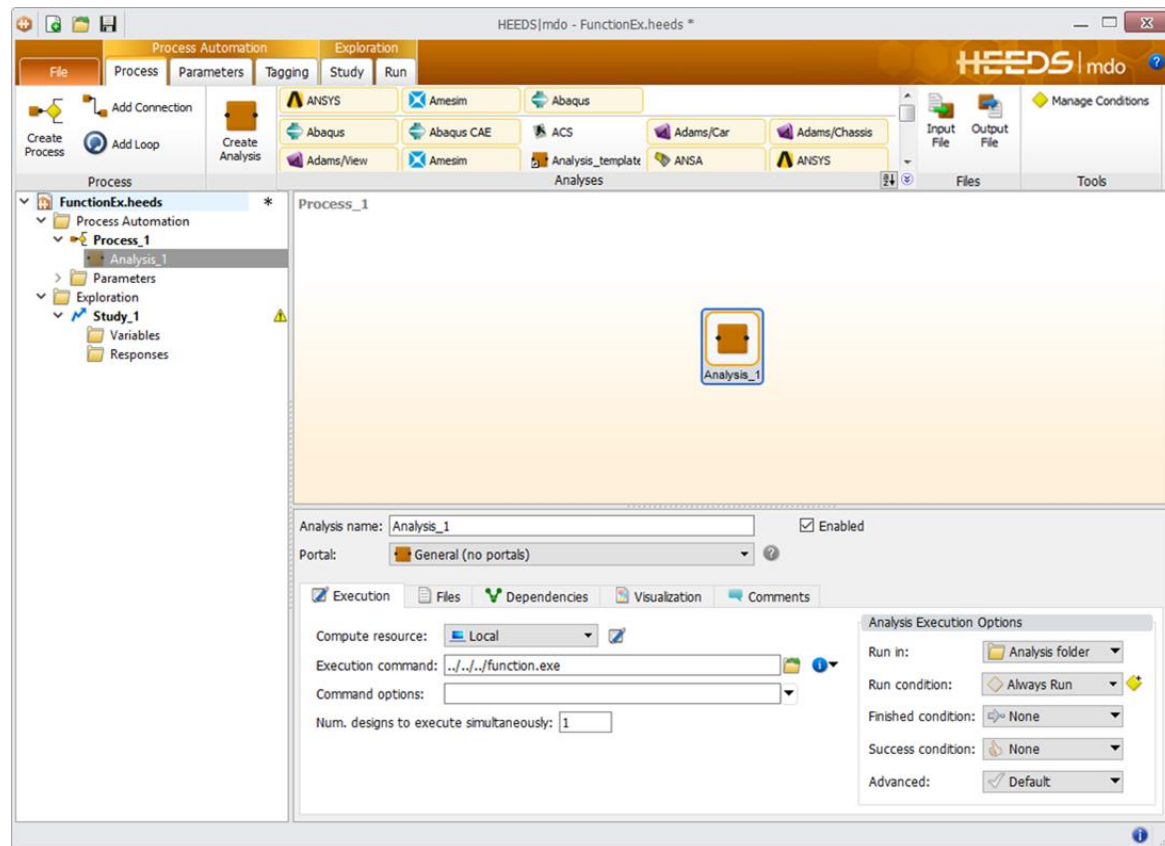
4. Click in the blank area on your screen to deselect **Analysis\_1**. The *Analysis Manager* closes.

## Defining the analysis tool

Our *Function* example has one analysis. To define it, we need to tell HEEDS MDO what analysis tool to use.

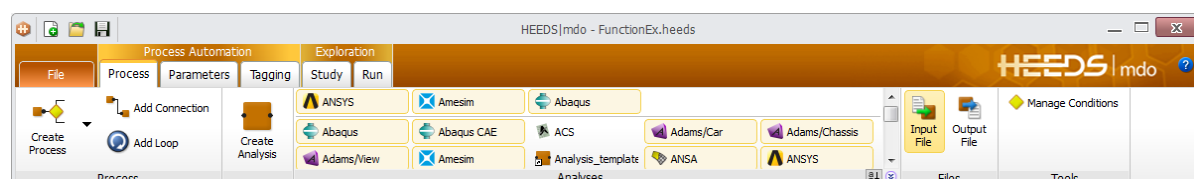
1. Click the **Analysis\_1** tool again to display the *Analysis Manager*. The *Analysis Manager* appears with fields to define your analysis: **Analysis name** and **Portal**.
2. Because there is only one analysis, keep the default name: **Analysis\_1**.
3. Keep the default value for portal: **General (No portals)**.
4. In the **Execution** tab, click  next to the **Execution command** field and browse for a file. The **Open File** dialog box appears.
5. Browse to the **Function** example directory and open **Function.exe**. **Function.exe** appears in the **Execution command** field. This executable does not require any command line options. If the executable did require command line options, you would enter them in the **Command options** field.

Your screen should look similar to this:



## Identifying the input and output files

1. Click the **Analysis\_1** tool. Notice when **Analysis\_1** is selected, the **Input File** and **Output File** options on the *Ribbon* appear in color. They change from black and white to let you know that they are now active. Next, we'll add an input file to the selected analysis.
2. On the *Ribbon*, click **Input File**:

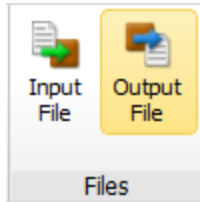


The *Open* dialog box appears.



**Note:** The **Input File** and **Output File** tools can also be found in the **Files** tab of the **Analysis Manager**.

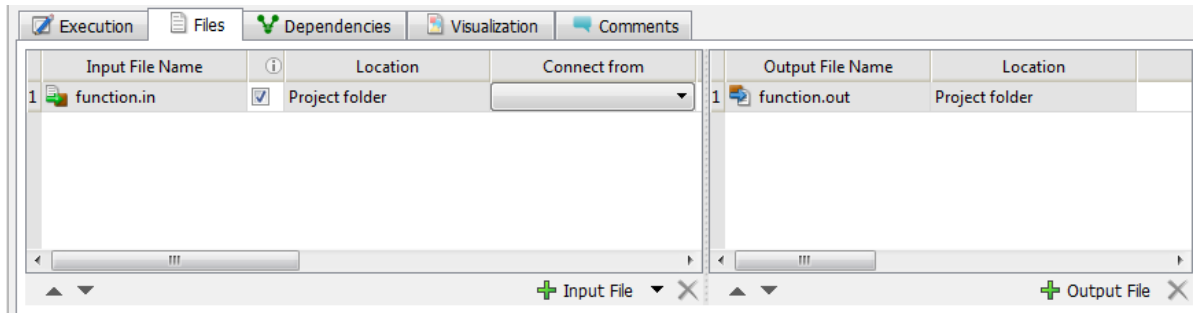
3. Navigate to the *Function* example directory, and double-click **function.in** (or highlight the file name, and click **Open**). The file is included in the *Files Manager* for this analysis.
4. On the *Ribbon*, click **Output File**:



The *Open* dialog box appears.

5. Navigate to the *Function* example directory, and double-click **function.out** (or highlight the file name, and click **Open**). The file *function.out* appears in the *Files Manager* for this analysis.
6. In the *Analysis Manager*, click the **Files** tab.

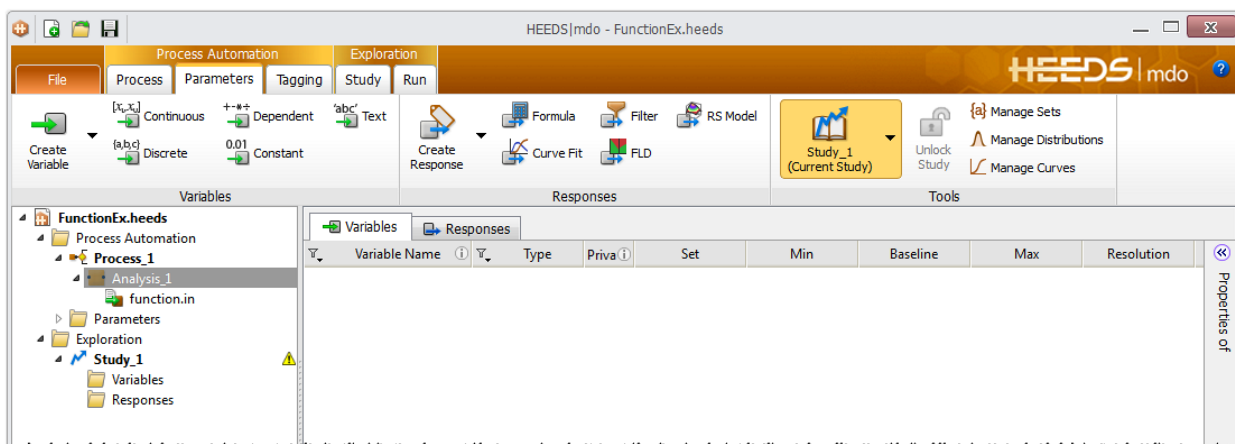
The *Files Manager* should contain the input and output files you just added. Your screen should look something like this:



7. Save your project.


## Step 3: Defining the Parameters

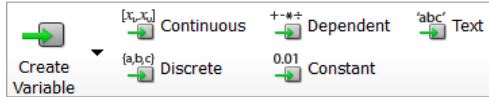
The next task of *Process Automation* is to define the project *Parameters*. This task includes identifying and defining the variables and responses that are required for your project. Click the **Parameters** tab. The *Parameters* page opens to the *Variables Manager*. You can also define the parameters on the **Study** tab.




## Identifying variables and responses

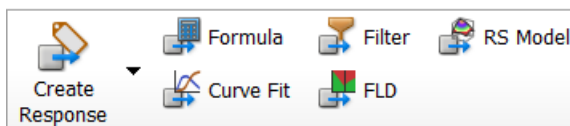
This section describes some of the functionality on the *Parameters* page for identifying and defining project variables and responses.

1. On the *Ribbon*, you can click  **Create Variable** to add a variable to your project, or you can select one of the tools that adds a specific variable type:



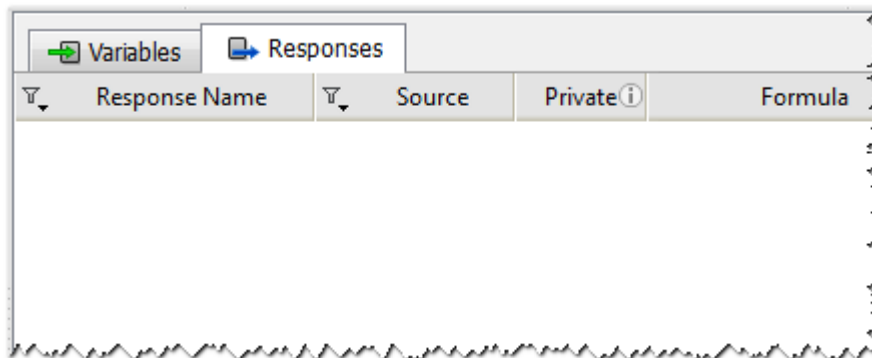
The variables you add will be listed in the *Variables Manager*.




2. On the *Ribbon*, you can click  **Create Response** to add a response to your project, or you can select one of the tools that adds a specific response type:



The responses you add will be listed in the *Responses Manager*.

3. From the *Parameters* page, you can click the  **Responses** tab. The *Responses Manager* appears where you can add and define the responses that are to be extracted from each design of your project:

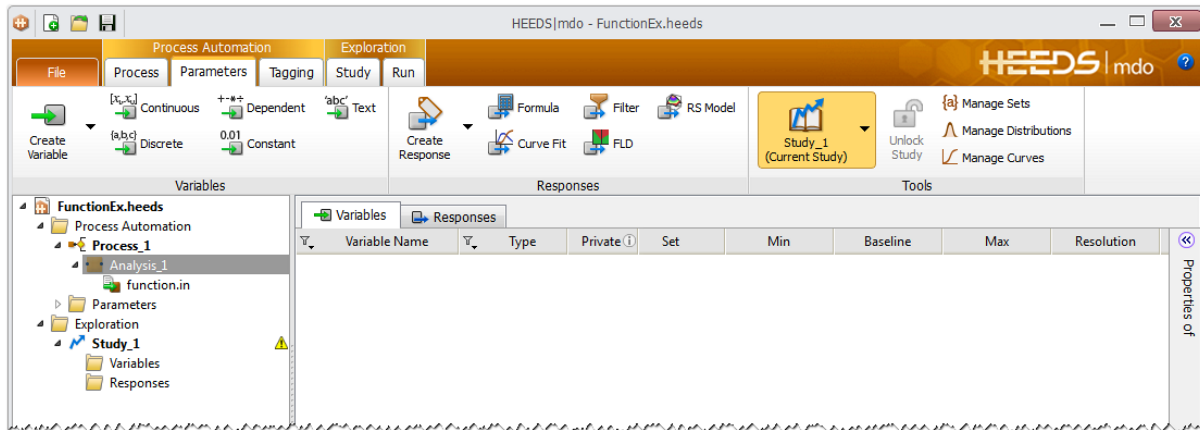



4. Although not needed for this example, you can use these tools on the *Ribbon* to further define variables and responses:
  -  **Manage Sets** to define values for discrete variables.
  -  **Manage Distributions** to define distributions for stochastic variables.
  -  **Manage Curves** to define responses determined by a curve fit.

## Defining the project variables

The project variables represent the values that will be varied for each evaluation. Our function example has only variable to define.

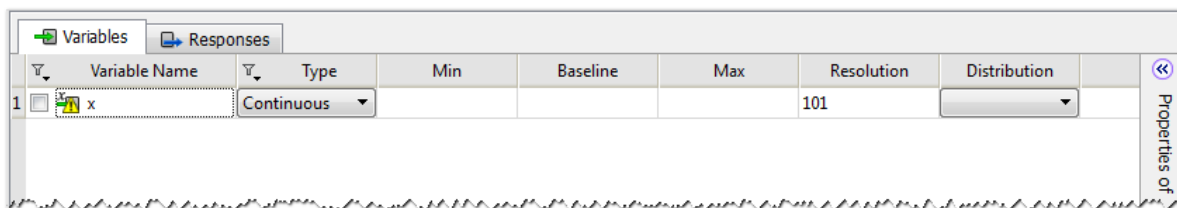
1. On the *Parameters* page, click the  **Variables** tab. The *Variables Manager* appears:



2. On the *Ribbon*, click  **Create Variable**. **Variable\_1** is added to the *Variables Manager* as a continuous type variable. *Continuous* is the default.

All of the values for your project variables appear in the table. **Variable\_1** for this project is shown as a continuous type variable.

3. Click the cell, **Variable Name**, and rename **Variable\_1** as **x**:



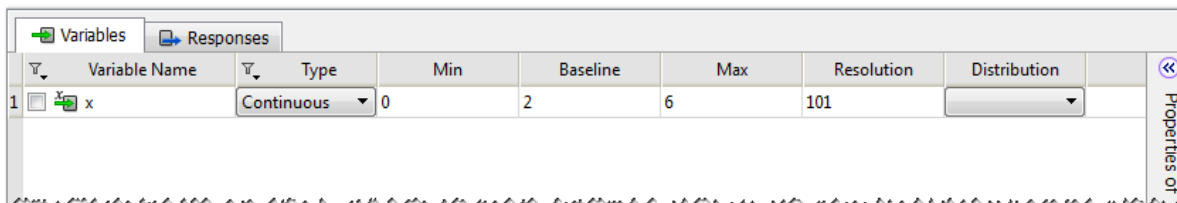
4. Because the specific value for **x** is chosen from a range of values, leave the default setting for **Type** as **Continuous**.
5. Enter the following values in the **Min**, **Baseline**, and **Max** fields:

Min	Baseline	Max
0	2.0	6

The **Baseline** value is the value from the baseline input file (although it does not necessarily need to be).

6. For **Resolution**, use the default value of **101**. The **Resolution** value determines the level of refinement of the variable. The other fields do not apply at this time.

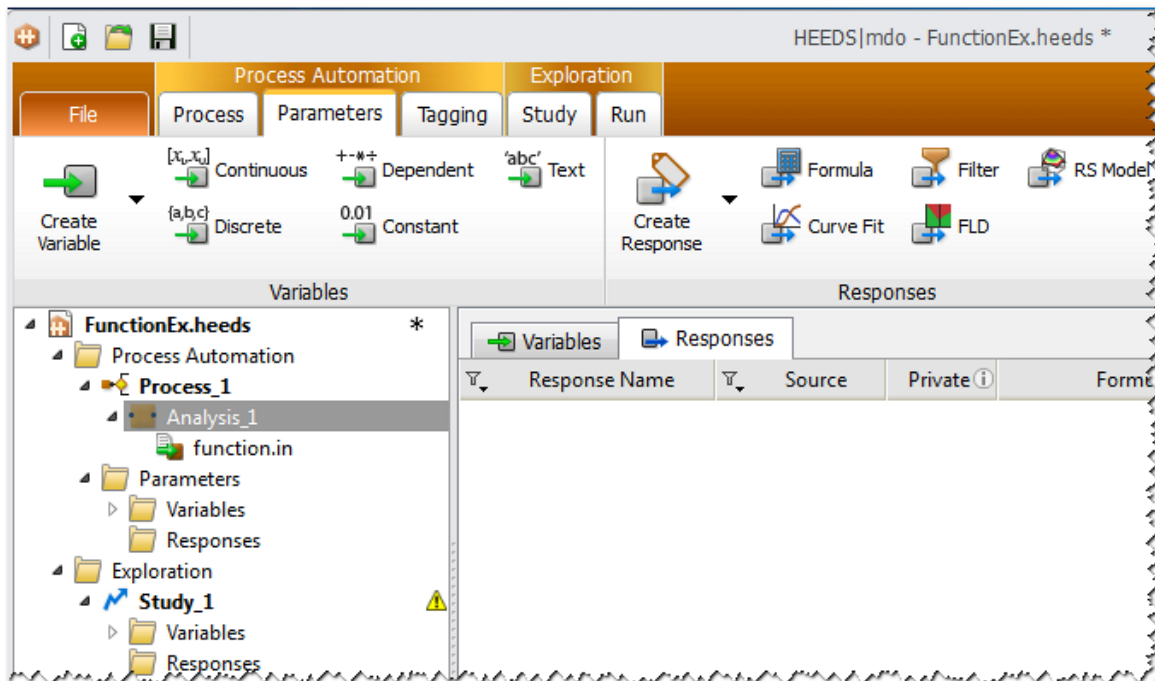
Your *Variables Manager* should look like this:



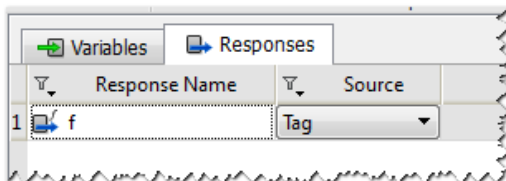
## Defining the project responses

The project responses are the values we want HEEDS MDO to test. The function example has only one response.

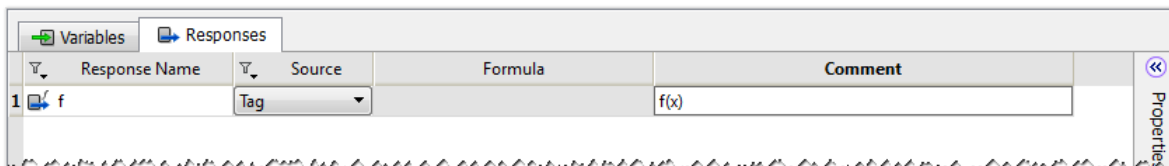
1. Click the **Responses** tab. The project *Responses Manager* appears:



2. On the *Ribbon*, click **Create Response**. **Response\_1** is added to the *Responses* table.
3. Click the **Name** cell, and rename **Response\_1** as **f**:



4. Leave the default setting for **Source** as **Tag**. The other source settings (*Formula*, *Curve Fit*, *Filter*, *FLD*, and *RSM*) are not applicable for this example.
5. The **Comment** field is useful for adding information that will help you remember the purpose of an object. This field is particularly valuable for problems with numerous definitions. Enter **f(x)** in the **Comment** field. The completed project *Responses* definition should look like this:

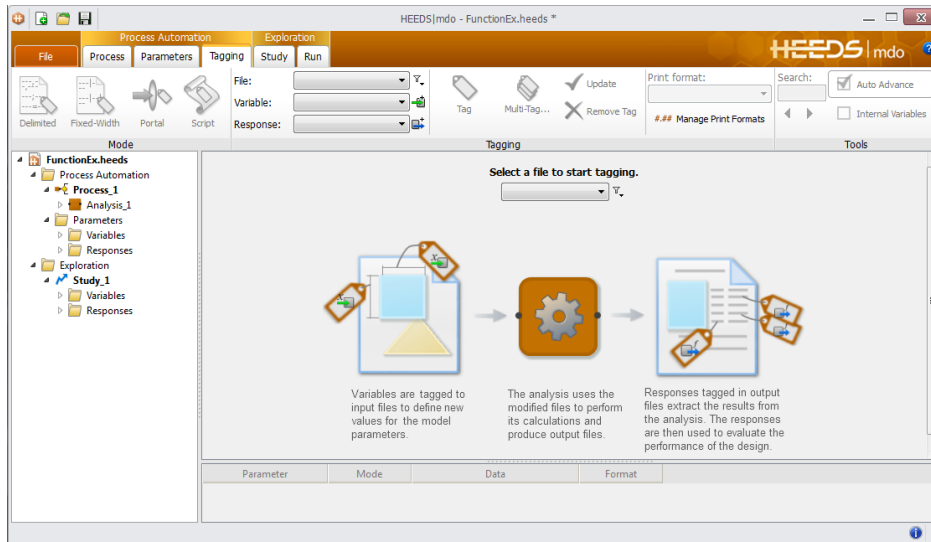


6. Save your project.

## Step 4: Tagging the Input and Output Files

You have identified your input and output files and defined your project variables and responses. Now you need to tag the values in your input and output files that represent the variables and responses.

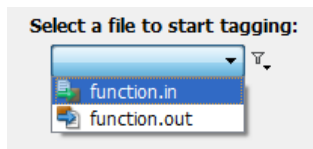
Select the **Tagging** tab. The *Tagging* page appears. The graphic images and text in the center of the screen summarize the tagging process:



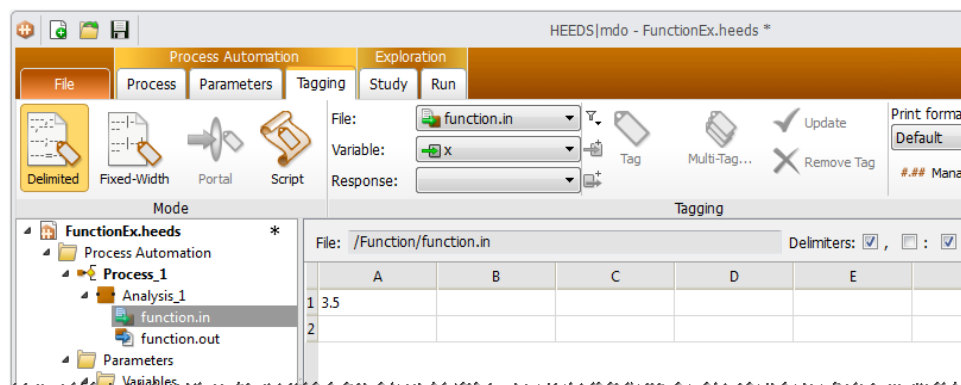
**Note:** You can also open the Tagging page from the tree menu on the left of your screen. On the tree, select **Process\_1**, **Analysis\_1**, and then **function.in**. The Tagging page is activated and you can see on the Ribbon a summary of the file in use, **function.in**, and the associated project variable, **X**. If you activate tagging in this way, you won't get the prompt as described next. *Tagging the input file*

When we tag the input file, we tell HEEDS MDO which values in the file correspond to the project variables we have defined. In this example, we have only one variable to identify:

1. On the *Tagging* page, locate the prompt: **Select a file to start tagging**.
2. Under the prompt, open the dropdown and select **function.in**:



A table with values for the parsed file, **function.in**, appears:



- Next we'll tag the value in the input file that corresponds to variable **x**. Select the cell in column **A**, row **1**, that contains the value, **3.5** (the only cell in the file for this example).

	A	B
1	3.5	
2		

- Now move to the *Ribbon* and click **Tag**. Variable **x** should appear in the table in place of **3.5**:

	A	B
1	x	
2		

### Tagging the output file

Next, we have to identify the project response value in the output file. The tagging procedure for responses is the same as above.

- On the *Ribbon*, locate **File** and open the dropdown.
- Select **function.out**. The table for the parsed response file appears.  
Alternatively, you can select **function.out** from the tree instead of from the *Ribbon*.
- Select the cell in column **A**, row **1**, that contains the value, **-1.88961** (the only cell in the file for this example):

	A	B
1	-1.88961	
2		

- Move to the *Ribbon* and click **Tag**. **Response f** should appear in place of **-1.88961**:

	A	B
1	f	
2		

- Save your project.

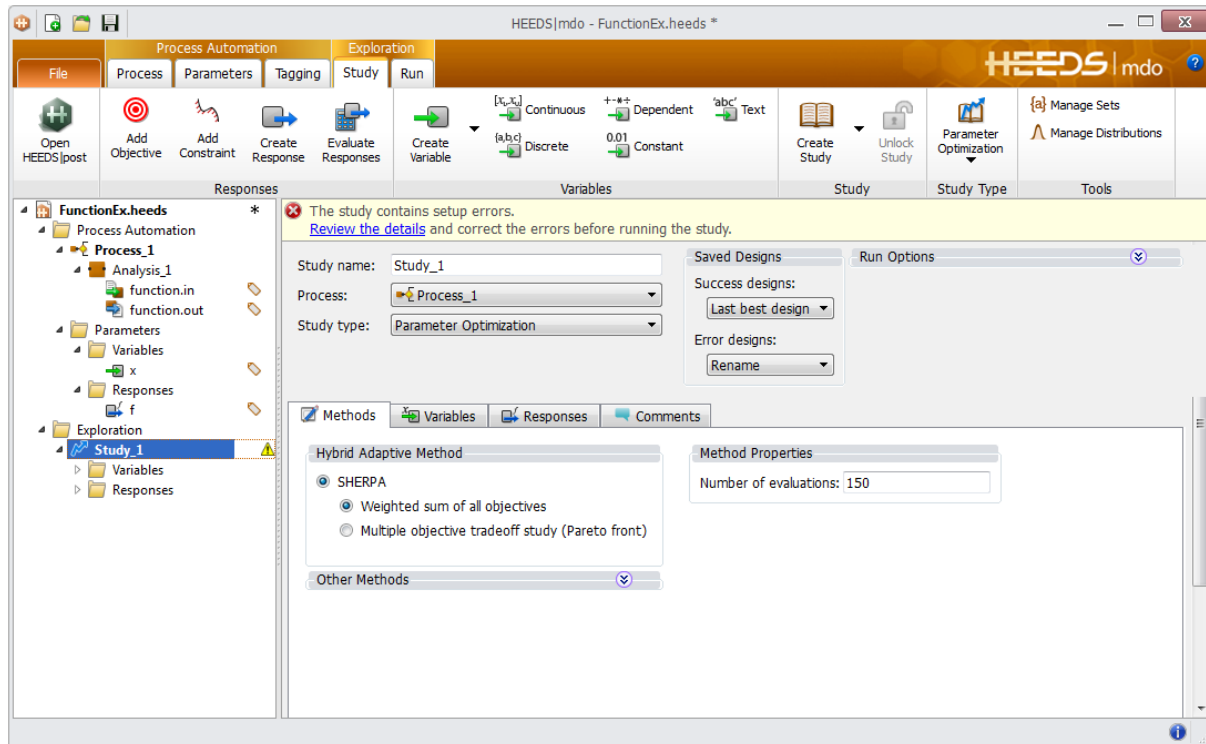
---

## Step 5: Defining the Study

Steps in sections one through four above have taken you through the *Process Automation* tasks of your project. This section helps you move through tasks at the *Exploration* level: *Study* and *Run*.

At the *Study* level, you create the problem definition of your study. This definition includes four parts: *Methods*, *Variables*, *Responses* and *Comments*. Since our example is a simple one, we need to define only *Methods* and *Responses*.

1. Select the **Study** tab on the tab bar. The *Study* page opens to the *Study Manager*:

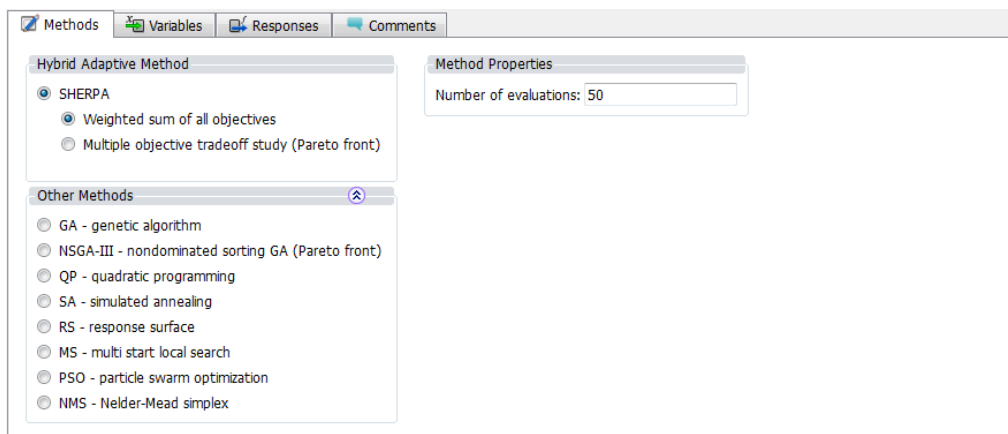


2. Notice the four tabs on the *Study Manager*: **Methods**, **Variables**, **Responses** and **Comments**. These tabs include all of the tasks required to complete a study's problem definition. In general, we work through the tabs in the order that they appear on the screen.

### Selecting the search method

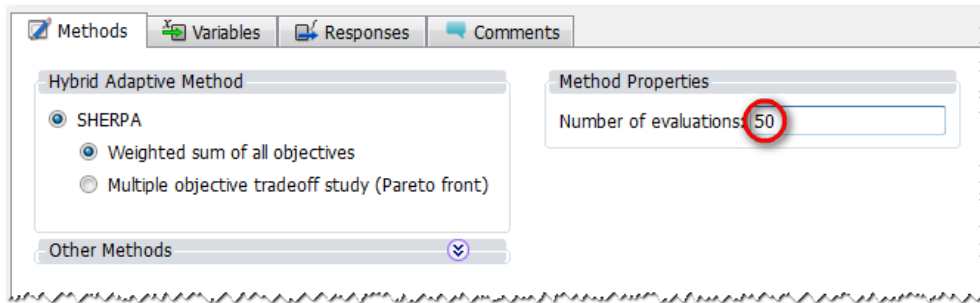
Here we'll select the search method for our study and we'll specify the number of evaluations we want HEEDS to run.

1. Click the **Methods** tab on the *Study Manager*.
2. Select the search method that you want to use for your study. For this example, choose the default, **SHERPA – weighted sum of all objectives**:



This method handles everything automatically, using whatever search methods that produce the best results. Weighted sum of all objectives is the best method to use for most studies. If we had a study with multiple objectives, we might want to use Multiple objective tradeoff study (Pareto front) to examine tradeoffs between objectives. The other methods are provided primarily for academic purposes.

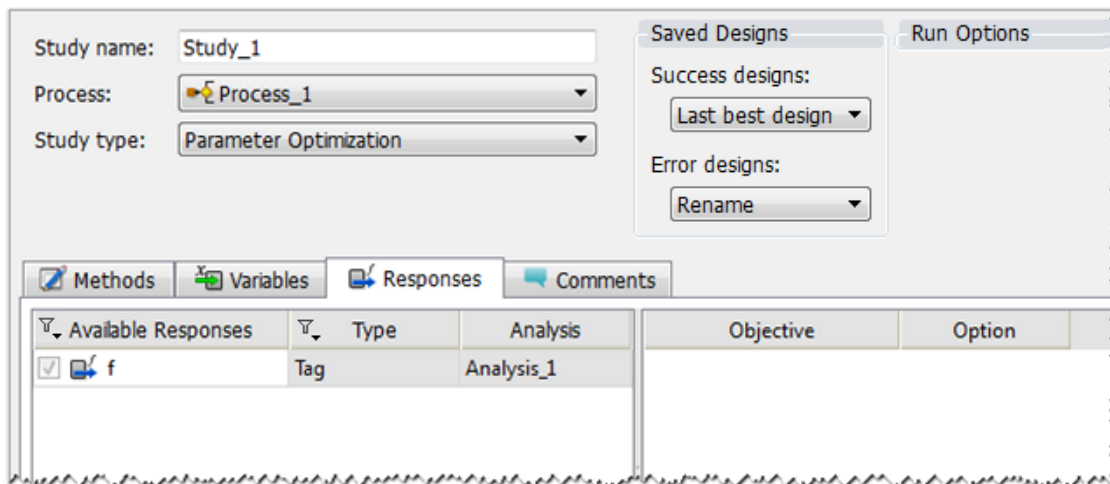
- Under **Method Properties**, locate the field: **Number of evaluations**. Enter **50** as the value:



### Completing the variable and response definitions

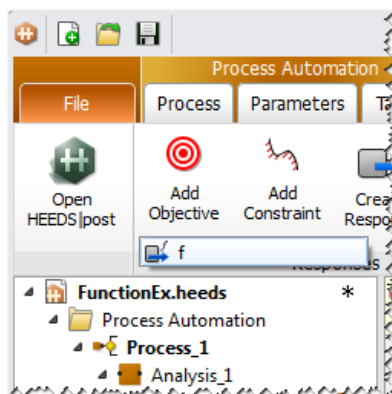
Our variable definition is complete for our problem definition. We will skip the **Variables** tab and move to the **Responses** tab where we will add an objective to our problem definition.

- On the *Study Manager*, click the **Responses** tab:



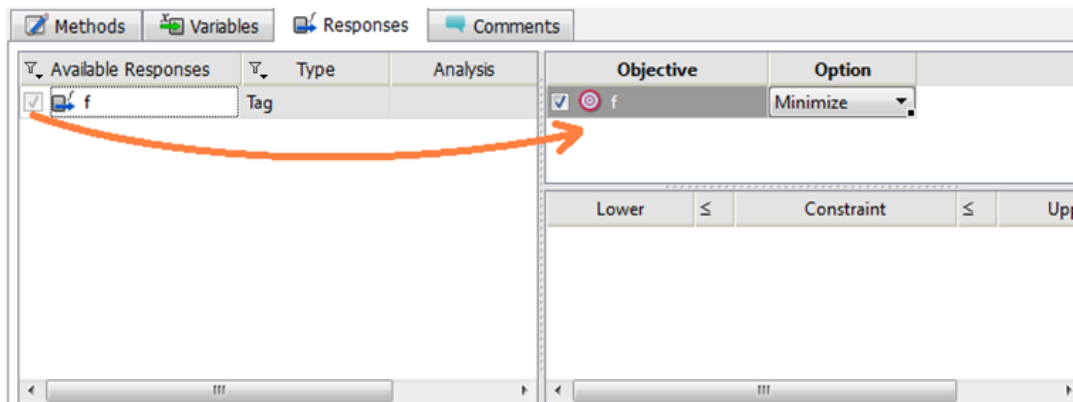
All available responses for our analysis are listed here. In our project, we have only one response.

- There are two ways you can add an objective. You can use the tool on the *Ribbon* or drag and drop:
  - From the *Ribbon*, click **Add Objective**. A list of available objectives appears. Select **f**:

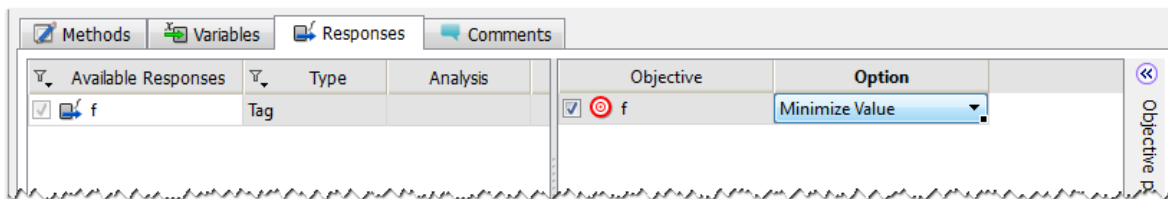


Response **f** is added as an objective for this problem definition.

- Alternatively, you can simply select the response from the *Responses* list, and then drag and drop it into the *Objective View*:

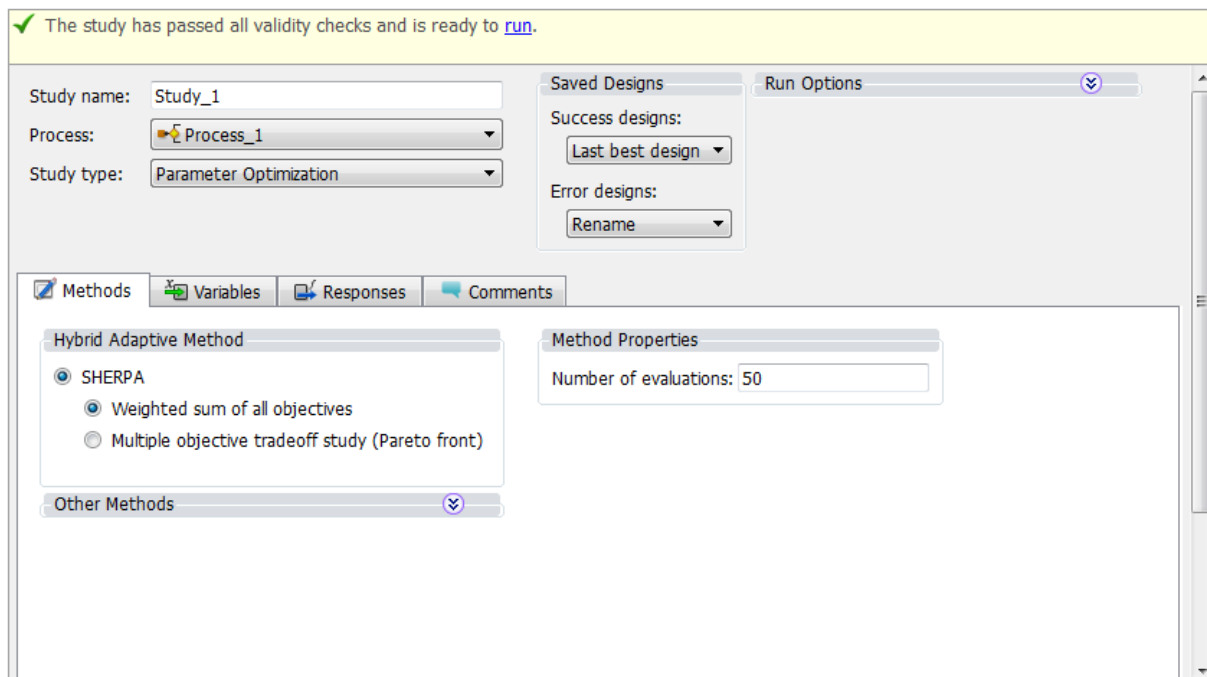


- On the *Study Manager*, locate the **Option** field and select **Minimize Value** from the list box. This option indicates that we are looking for the minimum value for the response:



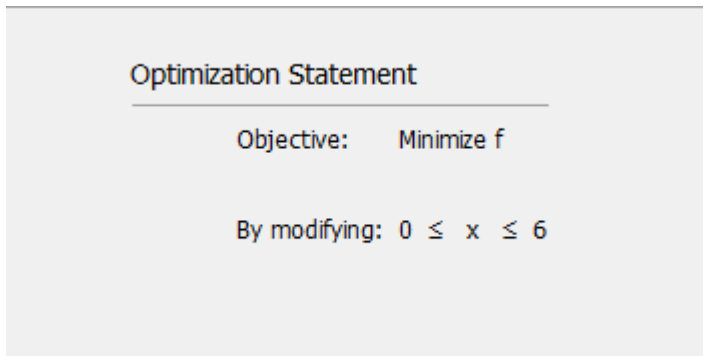
- Save your project.

Your completed *Study* definition should look like the one below:



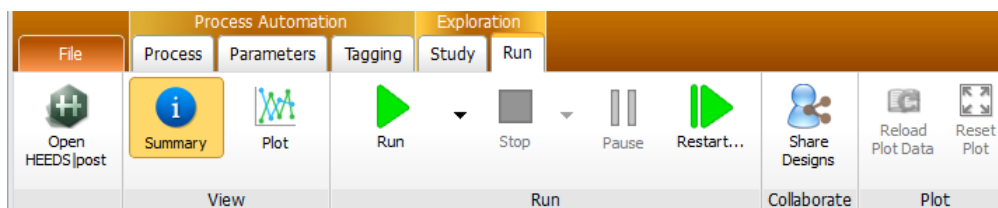
## Step 6: Running the Study

You execute your study from the *Run* page, where you can monitor it while it is running. Select **Run** from the tab bar and view the screen. A summary of your *Study Details* and *Optimization Statement* appears.

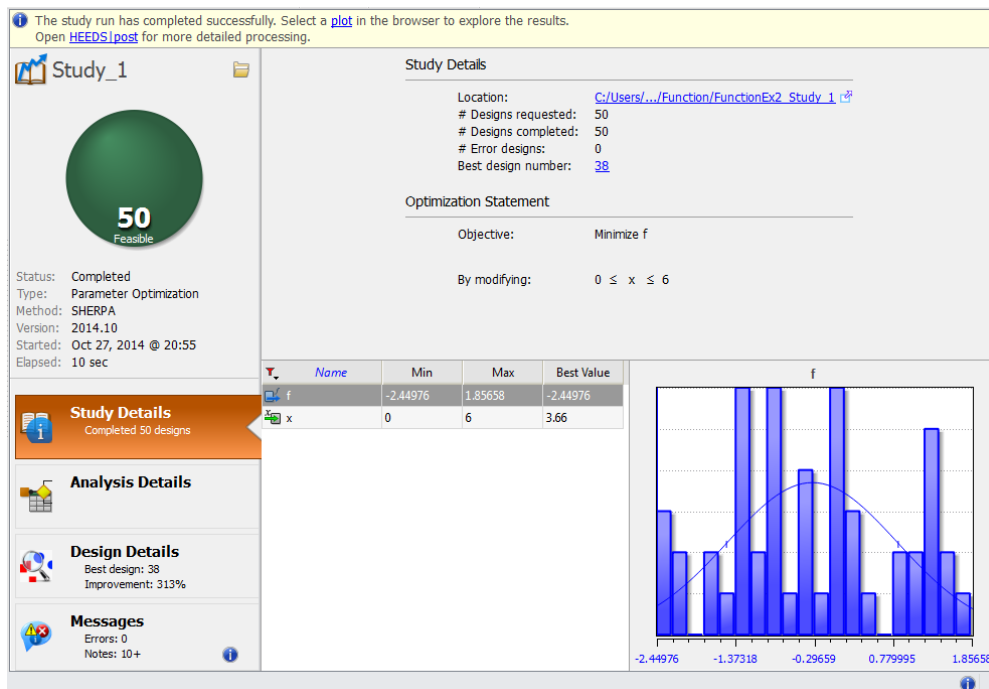


### To run the study

On the *Ribbon*, click **Run** to start the study:



As the study is running, you'll see the number of designs completed on the screen. Because this is a simple study, it will run very quickly. When the study is finished running, you should get a summary on your screen that looks like this:



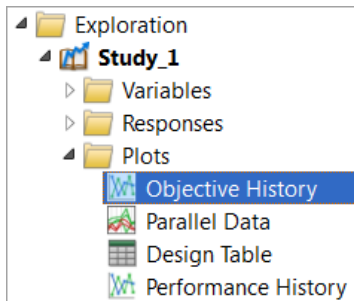
The *Study Summary* allows you to view details on the study and each analysis in the project. You can also investigate the details of a specific design, compared to the reference design, and view any messages that resulted from the run.

### Plotting data

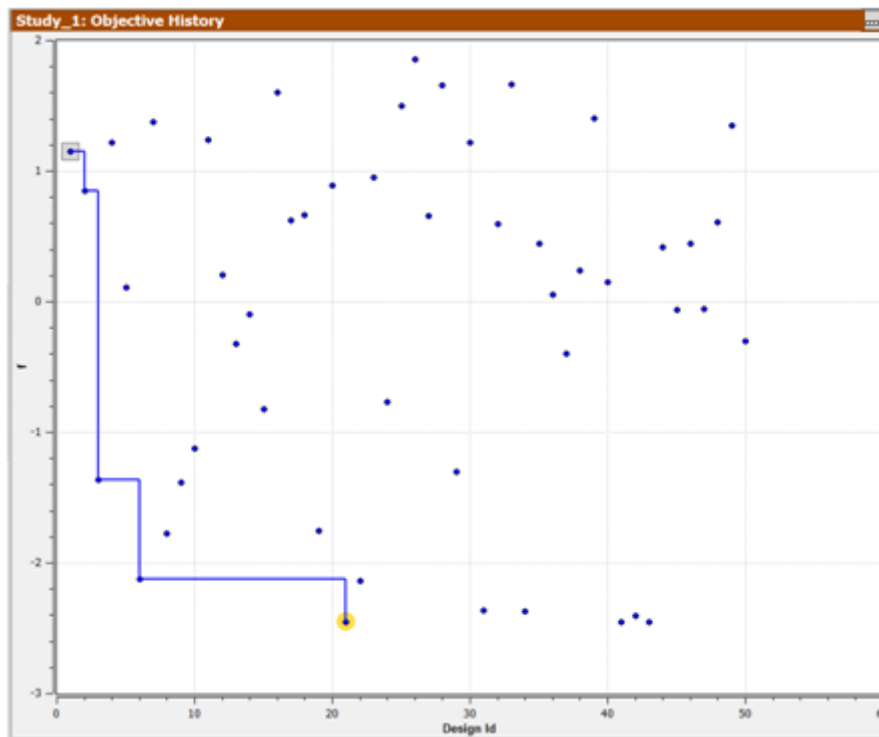
HEEDS gives you a number of plots to monitor your study while it is running: **Objective History**, **Parallel Data**, **Correlation**, and **Performance History**. You can also open a **Design Table** that provides a summary of each run. Let's take a look at these options beginning with the **Objective History** plot.

#### ► To see a plot of the objective

From the **Plots** node on the tree, choose **Objective History**:



A plot with the history of the objective appears:



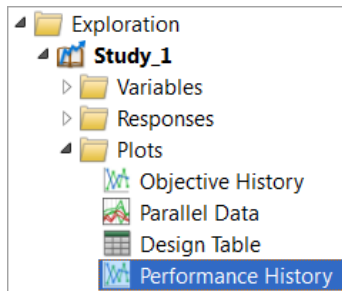
For each evaluation, the point plotted is the value for the objective,  $f$ . The *blue* line indicates the best design found at that time in the run. The point highlighted in *yellow* is the best design found, while the point highlighted in *gray* is the baseline design.



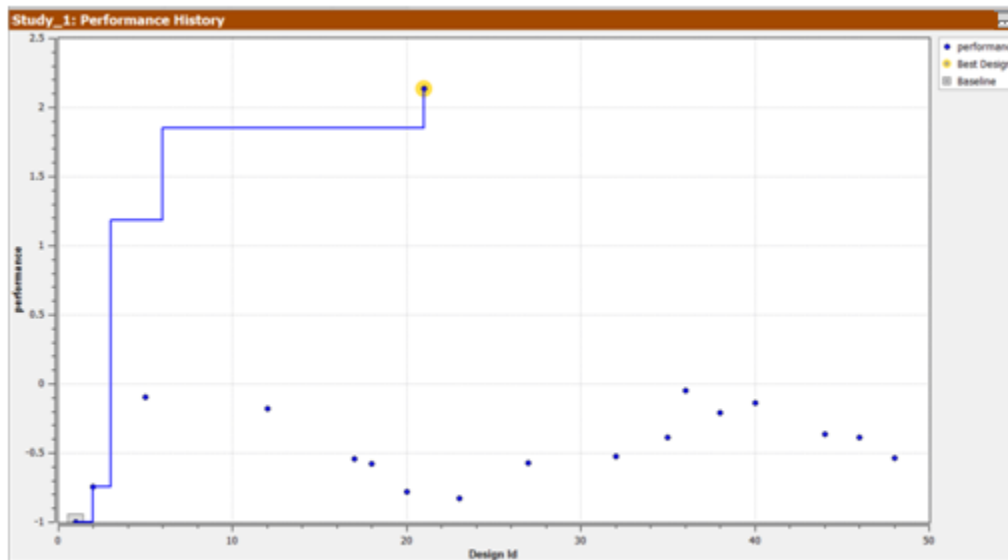
**Note:** If the best design and baseline design are not highlighted, click the **Best Design** and **Baseline** buttons in the **Plot Data** group of the **Plot** ribbon.

### ► To see a plot of the performance

From the **Plots** node on the tree, select **Performance History**:



A plot of your study's performance history appears:



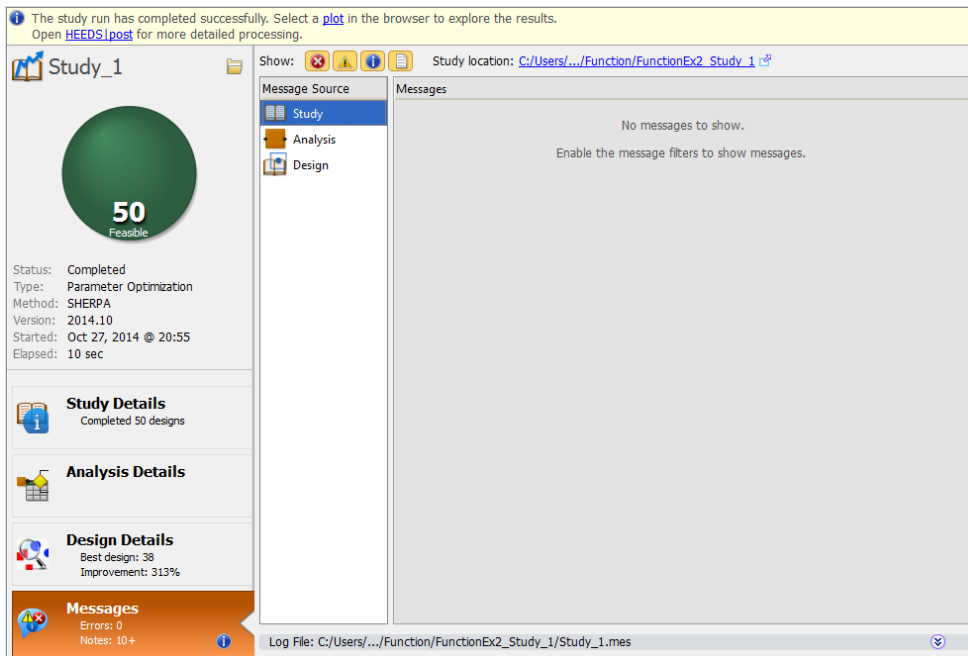
The performance plot line shows how well the objective is being met. We are always trying to maximize performance. A higher value means a better design.



**Note:** Your plots may look different than the ones in our illustrations, but your end result should be similar.

► **To view the contents of the study log file**

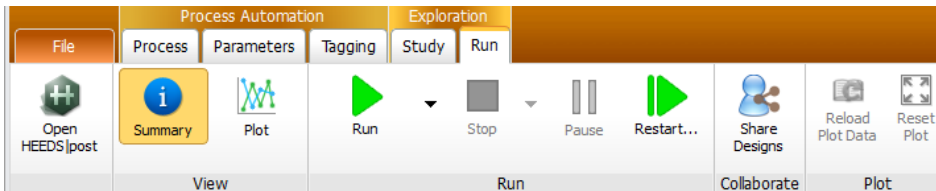
On the *Study Summary*, click the **Messages** tab.



If your study had messages, they would appear in the **Study** messages pane. You can also click **Analysis** and **Design** to view messages for specific analyses and designs.

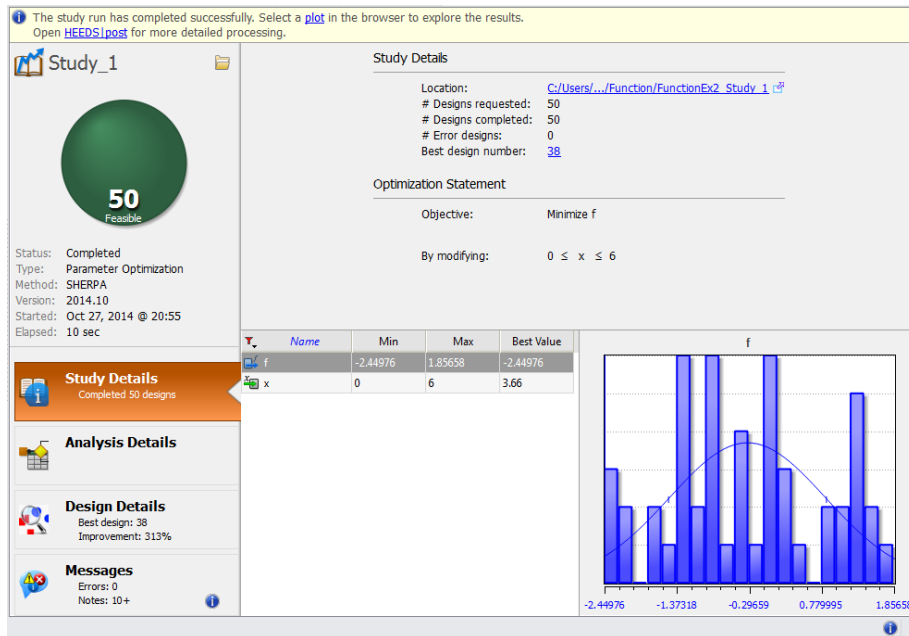
► **To view a summary of the study information:**

On the *Ribbon*, click **Summary**:



A summary appears of your study's *Study Details*, *Design Details*, *Analysis Details* and *Messages*. This page appears as the default when your study finishes running.

Use the tabs to move through the information.

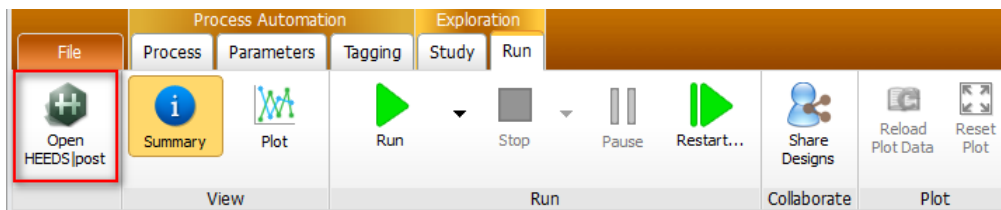


## Step 7: Viewing Results Using HEEDS POST

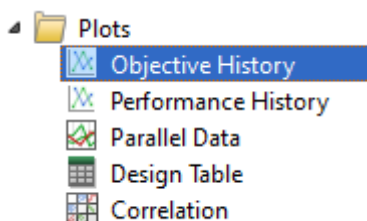
HEEDS MDO provides several plots for viewing the results of a parameter optimization run. These plots are displayed as read-only in HEEDS MDO. In HEEDS POST, you can access all of the same plots, modify them, and create new ones. When you switch back to HEEDS MDO, the new plots and modification are reflected.

### Viewing the objective history

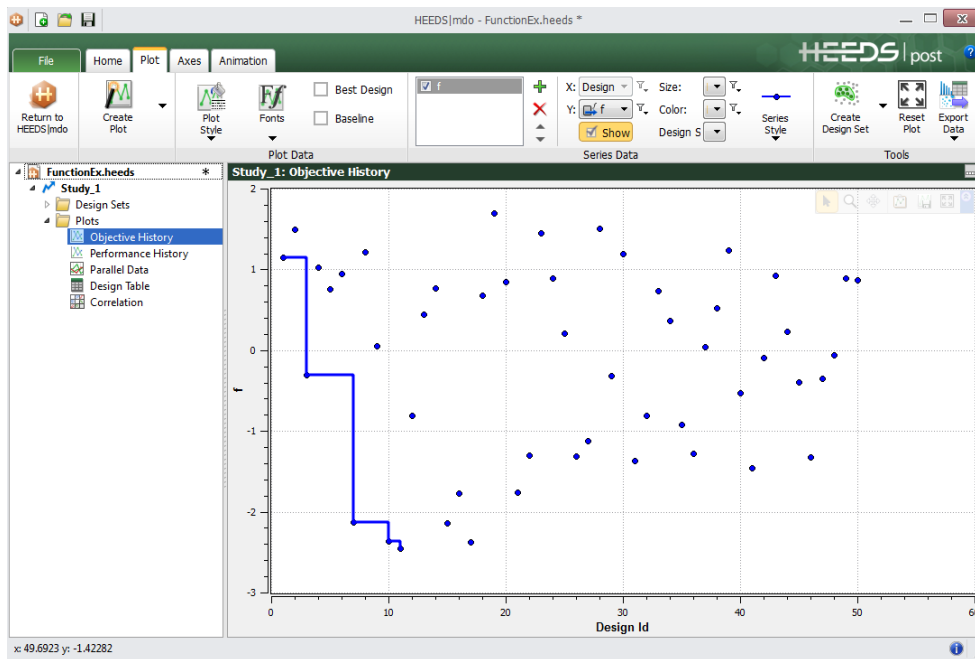
1. On the *Ribbon*, click **Open HEEDS|post**:



2. From the **Plots** node on the tree, select: **Objective History**:



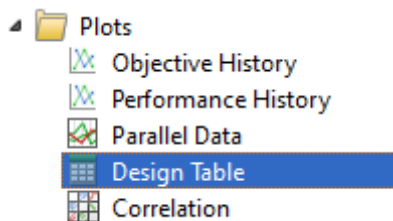
3. The **Objective History** plot appears. Notice that the tool ribbon contains the controls to make changes in the plots. Any changes you make here will be reflected when you view the plots in HEEDS MDO.



### Viewing the design table

Use the *Design Table* to review evaluations of the designs.

From the **Plots** node on the tree, click **Design Table**:



The **Study\_1: Design Table** appears with the performance ratings and the values of **x** and **f** for each design:

Study_1: Design Table			
Design Id	f	performance	x
1	1.14788	-1	2
2	1.5	-1.30676	0
3	-0.30168	0.262815	4.02
4	1.02418	-0.892236	1.98
5	0.756576	-0.659107	3.06
6	0.949931	-0.827553	4.98
7	-2.12262	1.84917	3.54
8	1.22019	-1.06299	2.28
9	0.0575224	-0.0501119	2.76
10	-2.36291	2.0585	3.6
11	-2.44976	2.13416	3.66
12	-0.807176	0.703189	3.96
13	0.444665	-0.387379	1.26
14	0.766337	-0.667611	0.54
15	-2.14007	1.86437	3.78
16	-1.77484	1.54619	3.84
17	-2.37303	2.06732	3.72
18	0.675901	-0.588826	4.92

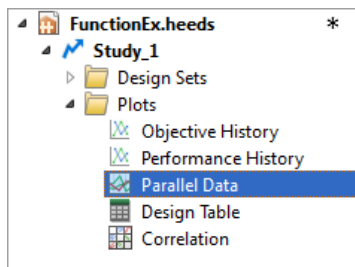
In the **Plot Data** group of the **Plot** ribbon, from the **Design options** list, select **Rank**. A new column with the design rankings is displayed. Click the column header, **Rank**, to sort by rank.

Study_1: Design Table				
Design Id	Rank	f	performance	x
11	1	-2.44976	2.13416	3.66
17	2	-2.37303	2.06732	3.72
10	3	-2.36291	2.0585	3.6
15	4	-2.14007	1.86437	3.78
7	5	-2.12262	1.84917	3.54
16	6	-1.77484	1.54619	3.84
21	7	-1.75524	1.52911	3.48
41	8	-1.45096	1.26403	5.76
31	9	-1.36221	1.18672	5.82
46	10	-1.31658	1.14697	1.56
26	11	-1.31496	1.14556	3.9
22	12	-1.30013	1.13264	3.42
36	13	-1.28005	1.11514	1.62
27	14	-1.12503	0.980094	5.88
35	15	-0.92112	0.802453	1.44
12	16	-0.807176	0.703189	3.96
32	17	-0.804853	0.701165	3.36
40	18	-0.529217	0.461039	1.38
45	19	-0.396011	0.344993	5.52
47	20	-0.344297	0.299942	1.8

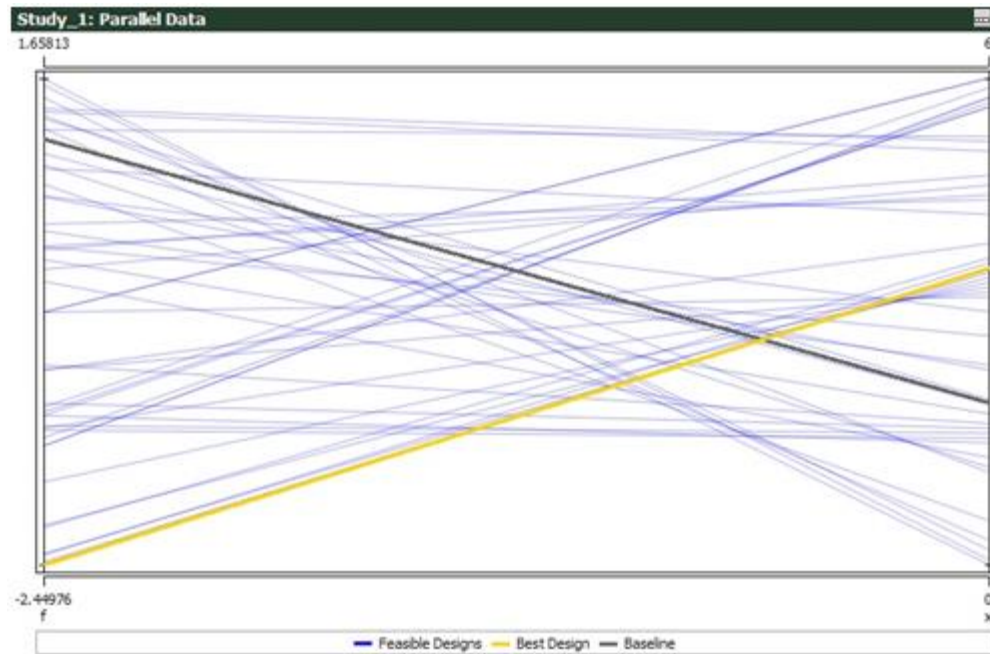
Now it is obvious that the design in the first row is the best (top-ranking) design, as it had the lowest value of **f** found. Notice that your best design should be similar in value, but the design may not have the same evaluation number.

### Viewing the parallel data

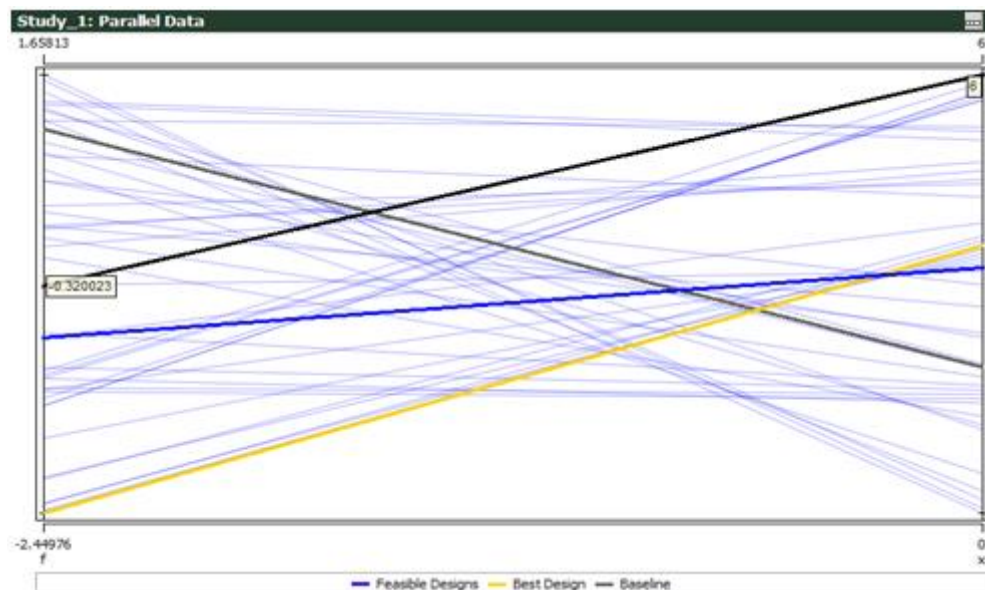
1. From the **Plots** node on the tree, click **Parallel Data**:



The **Study\_1: Parallel Data** plot displays values for  $f$  and  $x$  from all evaluated designs:



- To see how a single design compares with the other designs, click one of the lines:

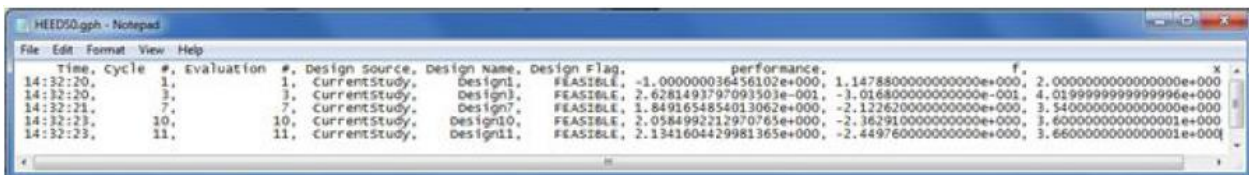


The value chosen for **x** and the response returned for **f** are displayed for that design. We will explain more about the parallel plot in our other examples when we have more variables and responses to work with.

## Step 8: Looking at the Results Files

HEEDS MDO saves a number of results files that contain information about the run. To determine the actual values returned for your design, you can look at the results files:

1. In your browser, navigate to the working directory for this study (this is the study folder located in the directory where you saved the project at the beginning of this example). There are a number of files with the prefix HEEDS. These are the results files that HEEDS produced during the run.
2. Open **HEEDS0.gph** in any text editor. You should see a display similar to the following:



Time	Cycle #	Evaluation #	Design Source	Design Name	Design Flag	performance	f	x
14:32:20	1	1	CurrentStudy	Design1	FEASIBLE	-1.00000036456102e+000	1.14788000000000e+000	2.00000000000000e+000
14:32:20	3	3	CurrentStudy	Design3	FEASIBLE	2.6281493797091503e-001	-3.01680000000000e-001	4.01999999999999e+000
14:32:21	7	7	CurrentStudy	Design7	FEASIBLE	1.8491654854013062e+000	-2.12262000000000e+000	3.54000000000000e+000
14:32:23	10	10	CurrentStudy	Design10	FEASIBLE	2.0584992712970765e+000	-2.36291000000000e+000	3.60000000000000e+000
14:32:23	11	11	CurrentStudy	Design11	FEASIBLE	2.1341604429981365e+000	-2.44976000000000e+000	3.66000000000000e+000

- This file contains the same information that was displayed in the *Designs* table: All **best** designs are listed. Every time HEEDS finds a design that is better than the others, HEEDS adds it to the list.
  - The **Design Flag** field tells you whether the design results were feasible, infeasible, or unsuccessful (error). A design is flagged as **ERROR** when results could not be extracted or any success conditions defined for the analyses were not met.
  - The performance rating and values for the response **f** and the variable **x** are listed for each best design.
3. Close the file.
  4. Open **HEEDS0.res** in your text editor. This file contains the same information as **HEEDS0.gph** for all the evaluations that were performed.
  5. Close the file and your browser.

## Example 2: Design of a Cantilevered Beam

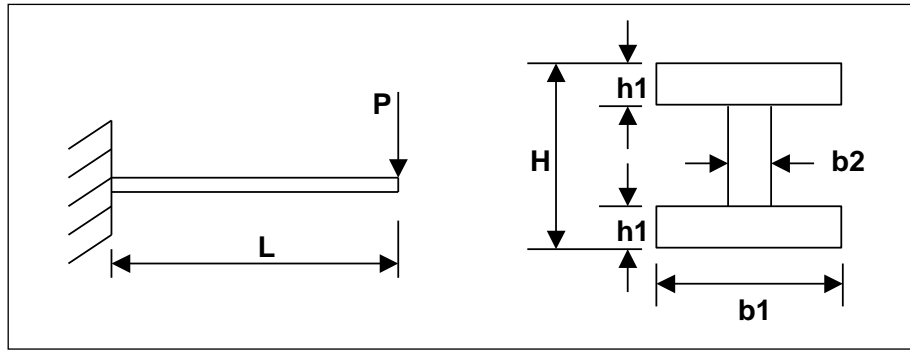


Fig. 1. Cantilevered beam

In this parameter optimization problem, the cross-sectional shape of a cantilevered beam is to be designed so as to minimize the mass of the beam for the load case shown in fig. 1. The beam is made of aluminum. This example shows you how to set up a project with several variables and several responses and illustrates the use of discrete variables in HEEDS MDO.

### The problem

The current optimization problem is defined as:

- Minimize:** volume of the beam
- Subject to:** maximum bending stress,  $\sigma \leq 5,000$  psi  
tip deflection,  $\delta \leq 0.1$  inches
- By varying:**  $h1$  from the set  $\{0.1, 0.25, 0.35, 0.5, 0.65, 0.75, 0.9, 1\}$   
 $2.0 \text{ in.} \leq b1 \leq 12.0 \text{ in.}$   
 $0.1 \text{ in.} \leq b2 \leq 2.0 \text{ in.}$   
 $3.0 \text{ in.} \leq H \leq 7.0 \text{ in.}$

### The baseline design

For this example, an executable named *cbeam.exe* was created to solve for the volume, max stress ( $\sigma$ ), and tip deflection ( $\delta$ ) using the following equations:

$$\begin{aligned} \text{Volume} &= [2 \cdot h1 \cdot b1 + (H - 2 \cdot h1) \cdot b2] \cdot L \\ \sigma &= P \cdot L \cdot H / (2 \cdot I) \\ \delta &= P \cdot L^3 / (3 \cdot E \cdot I) \\ \text{where: } I &= 1/12 \cdot b2 \cdot (H - 2 \cdot h1)^3 + 2 \cdot [1/12 \cdot b1 \cdot h1^3 + b1 \cdot h1 \cdot (H - h1)^2 / 4] \end{aligned}$$

The input file used by the executable contains values for  $P$ ,  $E$ ,  $L$ ,  $H$ ,  $b1$ ,  $h1$ , and  $b2$ .

The files for our example are found in the **CBeam** folder in the **Examples** directory:

- The default path for Windows is: **C:\HEEDS\Version\Examples\CBeam**
- The default path for Linux and UNIX is: **/opt/HEEDS/Version/Examples/CBeam**

If you installed HEEDS MDO in a different directory, look for the files in that directory instead. This example includes three files:

Option	Description
<b>cbeam.inp</b>	The input file.
<b>cbeam.out</b>	The output file.
<b>cbeam.exe</b> (Windows) <b>cbeam</b> (Linux)	The file that contains the analysis code.

## Step 1: Starting a New Project

---

If you have the function project from *Example 1* open, you can start a new project without closing HEEDS MDO. The new project will be added to the tree.

1. Create a directory for your new HEEDS project. Choose a location where you have write permissions.
2. From the **File** tab, select **New**. A new project appears on your screen.
3. From the **File** tab, click **Save**. The *Save Project As* dialog appears.
4. Browse to the directory that you have set up to save this project, and enter **CBeam\_Opt** as your project name. The file is saved with a **.heeds** extension.



**Note:** Prior to starting each new HEEDS MDO project, we recommend creating a directory specifically for that project. Be sure to choose a location where you have write permissions.

## Step 2: Defining the Process

---

As we have seen in *Example 1*, we begin our project with the tasks of *Process Automation*. Our first task is to identify the analyses of our *Process*.

### Identifying the analyses

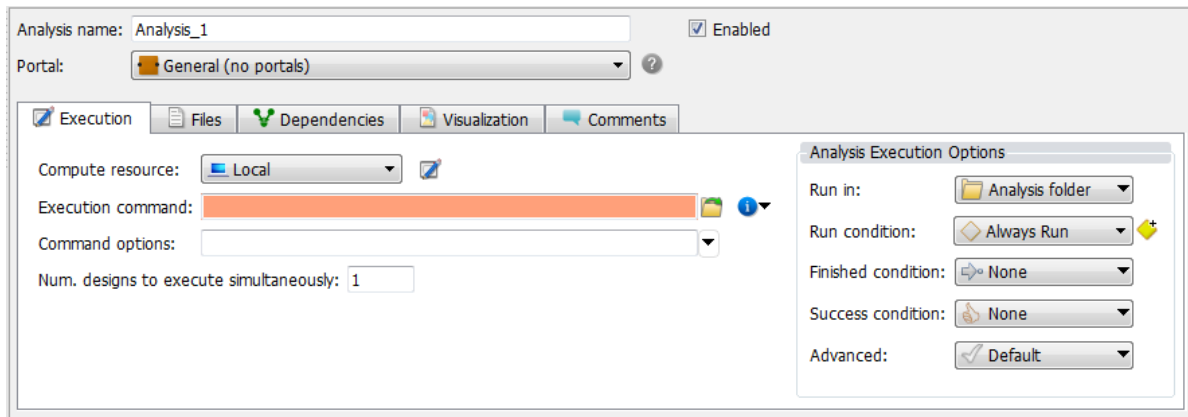
Like our function example, the beam example has only one analysis with one analysis tool and one input and one output file:


#### **Defining the analysis tool**

1. Select the **Process** tab. The *Process* page appears.
2. Click the **Analysis\_1** tool on your screen:

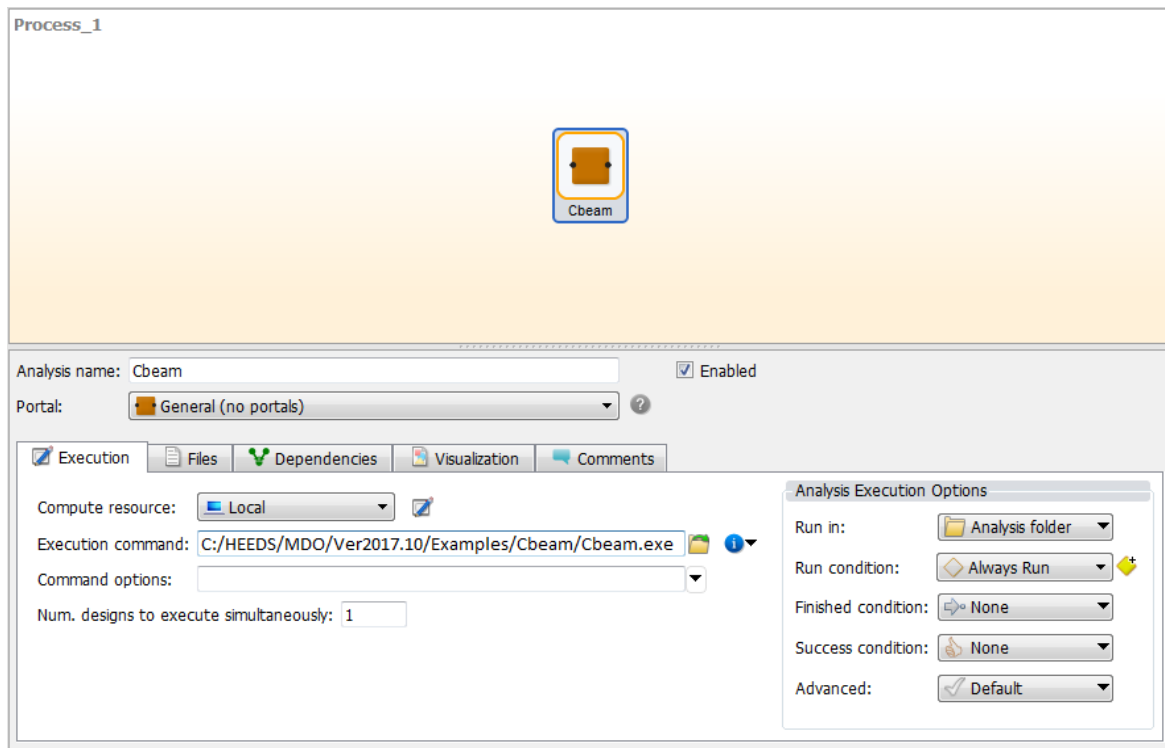


The *Analysis Manager* opens the **Execution** tab:

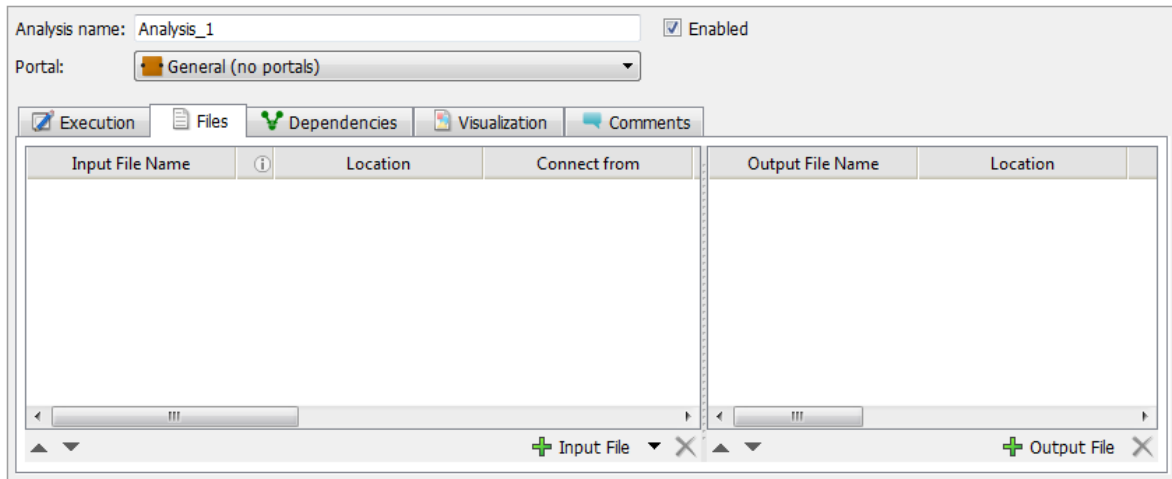


3. Find the **Execution command** field on the *Analysis Manager*, and click  to browse for a file. The **Open File** dialog box appears.
4. Browse to the **CBeam** example directory and open **CBeam.exe**. **CBeam.exe** appears in the **Execution command** field.

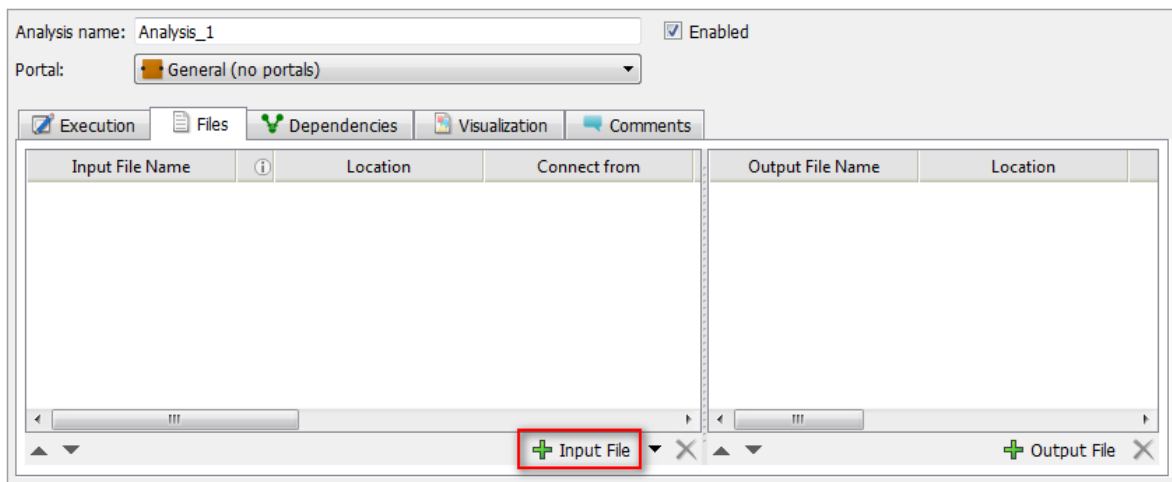
Your *Analysis Manager* will look similar to the following, although your path may be different:



5. Next, click the **Files** tab. The *Files Manager* appears:



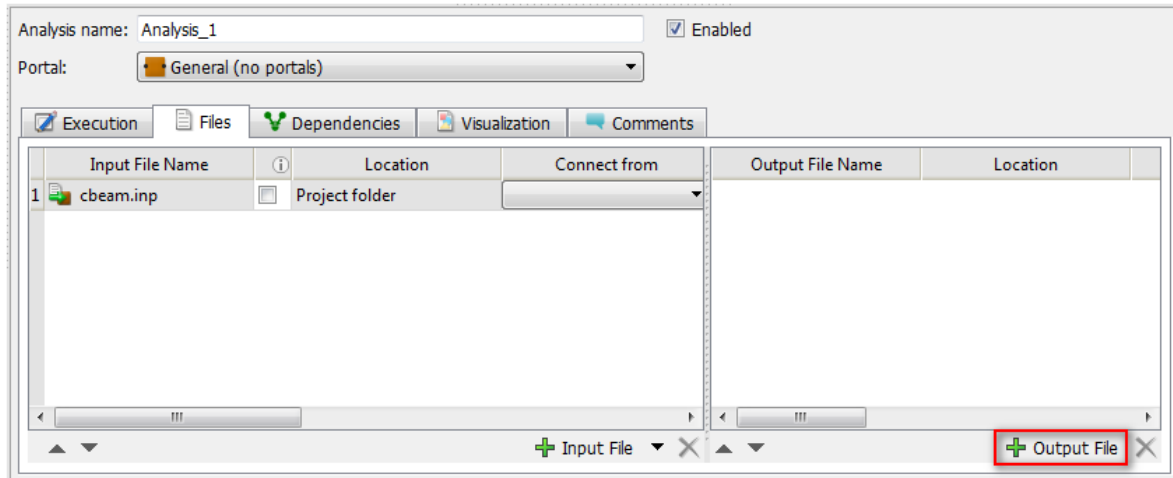
6. On the bottom of the *Files Manager*, click **+ Input File**:



The *Open* dialog box appears.

7. Navigate to the **CBeam** folder, select **cbeam.inp**, and click **Open**. The *Files Manager* includes **cbeam.inp** as an input file.

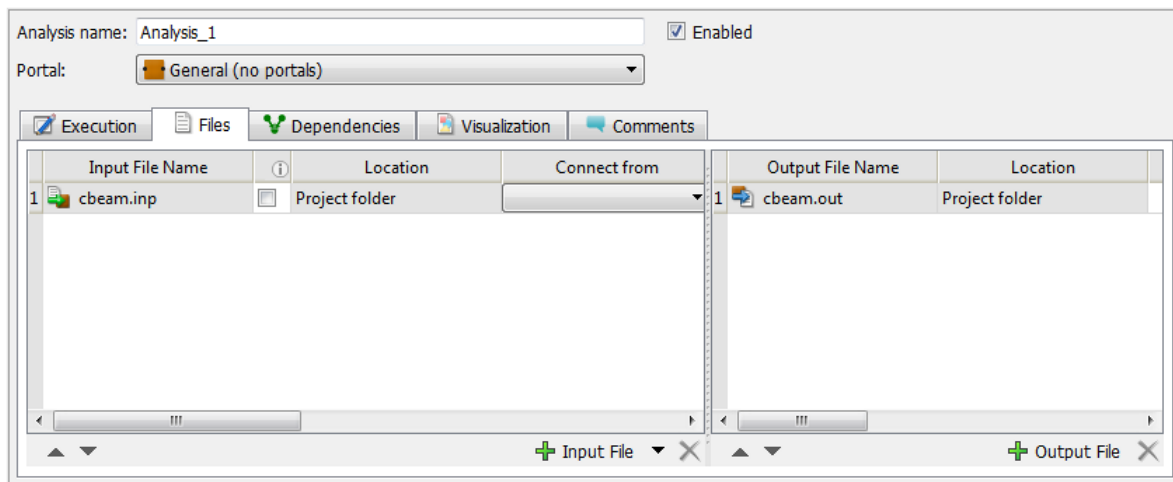
8. On the bottom of the *Files Manager*, click **+ Output File**:



The *Open* dialog box appears.

9. Navigate to the **CBeam** folder, select **cbeam.out**, and click **Open**. The *Files Manager* includes **cbeam.out** as an output file.

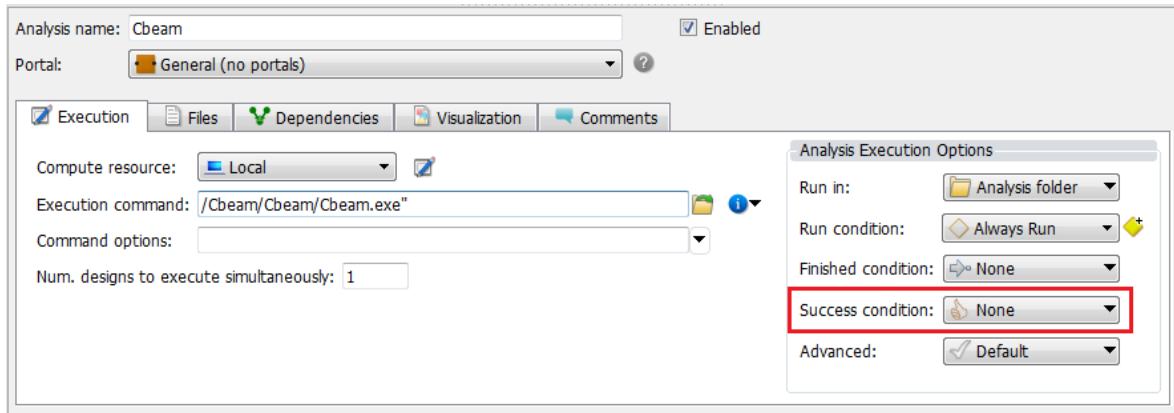
Your completed *Analysis* definition should look something like this:



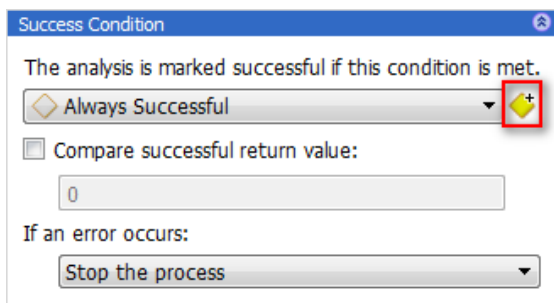
10. Click the **Execution** tab to add a success check condition. (This step was postponed until files had been added to the analysis.)

**A success check** lets you tell HEEDS what criterion to use to determine whether an evaluation is successful or not successful.

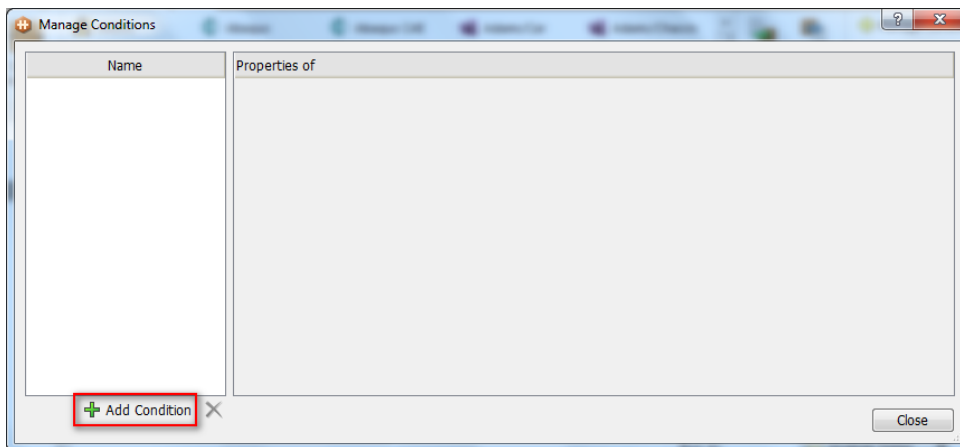
11. In the Analysis Execution Options area, click the **Success condition** field:



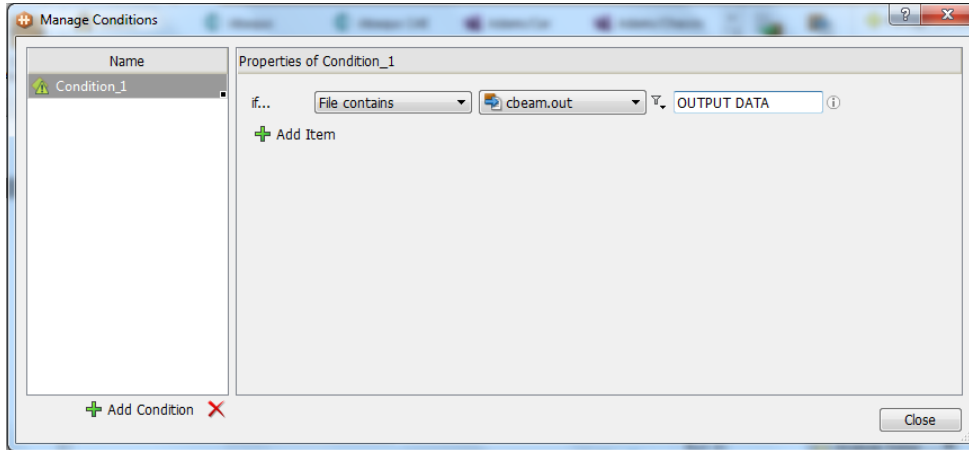
12. In the **Success Condition** dialog box, click the **Manage conditions** button to create a new condition:



13. In the **Manage Conditions** dialog box, click **Add Condition**:



14. Create a new condition using the following parameters:



This condition tells HEEDS that the analysis is considered successful if the **cbeam.out** file contains the phrase **OUTPUT DATA**.

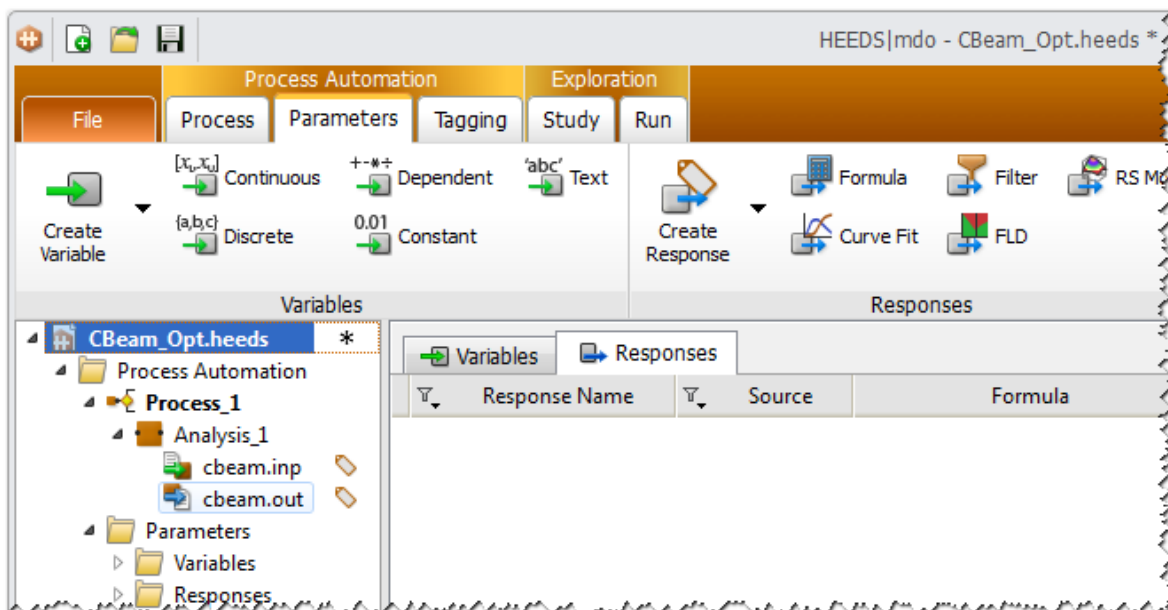
15. Close the **Manage Conditions** dialog box.
16. Save your project.

### Step 3: Defining the Parameters

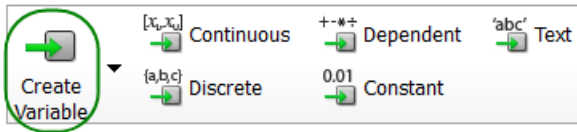
The next task at the *Process Automation* level of our project is to define *Parameters*. This task includes identifying the variables and responses that we want to include in our project. Our *C-Beam* example has a larger set of variables and responses than the *Function* example. The *C-Beam* project has four variables and three responses. One of the variables is discrete and requires that we create a discrete set for it.

#### Identifying the project variables

1. Click the **Parameters** tab. The *Parameters* page opens to the *Variables Manager*: You can also define the parameters on the **Study** tab.



- On the ribbon, locate the **Variables** group. Click **Create Variable** four times to add four variables to the project.



- Click the **Variables** tab. All of the values for your project variables appear in the table. The **Create Variable** tool uses the **Continuous** type variable as the default:

Variables		Responses						
	Variable Name	Type	Min	Baseline	Max	Resolution	Distribution	
1	Variable_1	Continuous				101		
2	Variable_2	Continuous				101		
3	Variable_3	Continuous				101		
4	Variable_4	Continuous				101		

- Rename the variables and define them as follows:

Variable Name	Type	Min	Baseline	Max	Comment
b1	Continuous	2	4	12	Flange width
h1	Continuous				
b2	Continuous	0.1	0.25	2.0	Web width
H	Continuous	3	3.5	7	Total beam depth



**Note:** Depending on your screen settings, you may need to use the scroll bar to display the Comment column. You can also tab to the Comment column if you prefer.

- Next change the variable type of **h1** from **Continuous** to **Discrete**: Click the list box for **Type** of **h1**, and select **Discrete**:

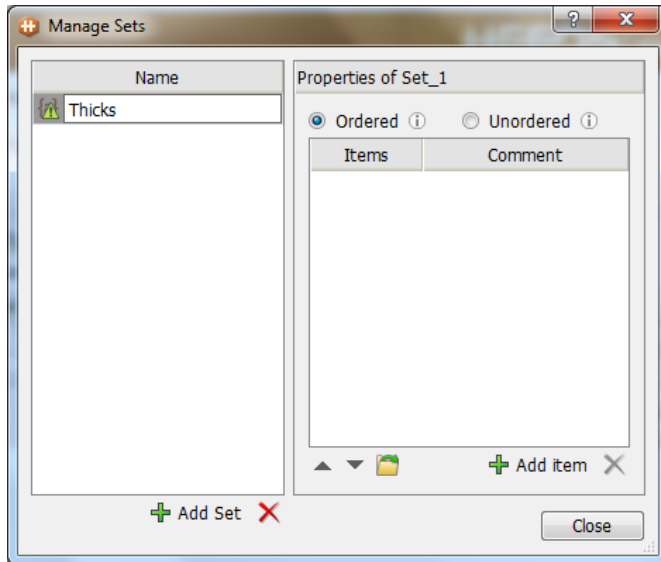
Variables		Responses			
	Variable Name	Type	Set	Min	
1	b1	Continuous		2	4
2	h1	Discrete			
3	b2	Continuous		0.1	0.25
4	H	Continuous		3	3.5

- Add a comment, **Flange thickness**, to **h1**.
- Create a *discrete variable set* to define the baseline value. The following steps explain how to do this.

#### ► Defining a discrete set

- On the *Ribbon*, Click **Manage Sets**. The *Sets Manager* appears.
- Click **Add Set** in the *Sets Manager*. **Set\_1** appears.

3. Click **Set\_1** and rename it **Thicks**. The new set name appears in the *Sets Manager*:

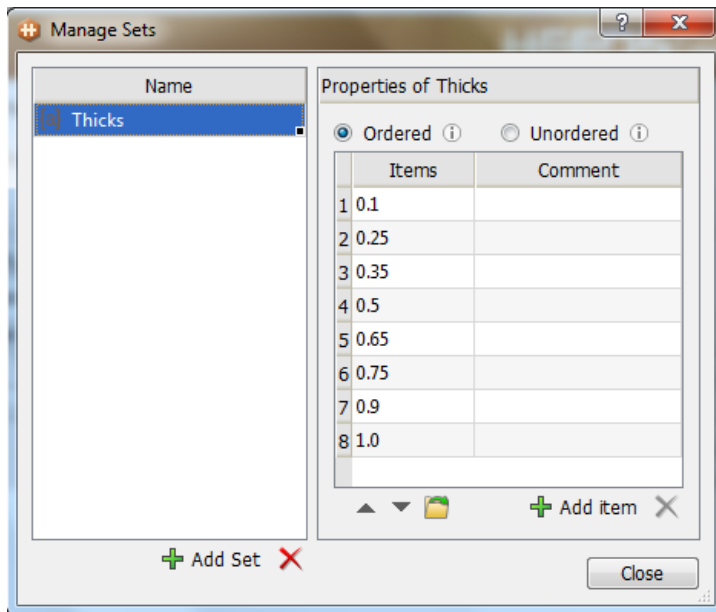


4. Next we'll add items to the set. In the **Properties of Thicks** box, click **+ Add item**.
5. Add 7 more items. You'll have a total of eight items.
6. To add a value to an item, click the item's cell, and then type in the value. In this manner, enter the following values to the items:

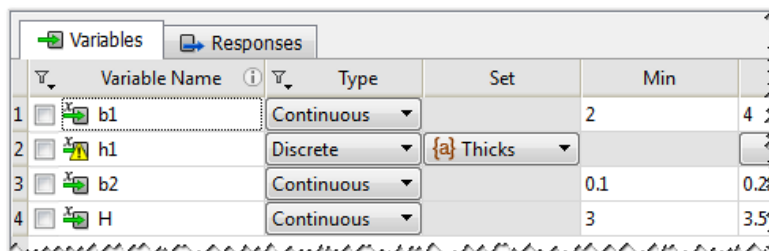
Item	Value
1	0.1
2	0.25
3	0.35
4	0.5
5	0.65
6	0.75
7	0.9
8	1.0

- Since our items are in numerical order, keep the **Ordered** option clicked on in the *Sets Manager*. The **Ordered** option tells HEEDS MDO that our set items have a specific order as opposed to a random order.

The completed discrete set definition should look like this:

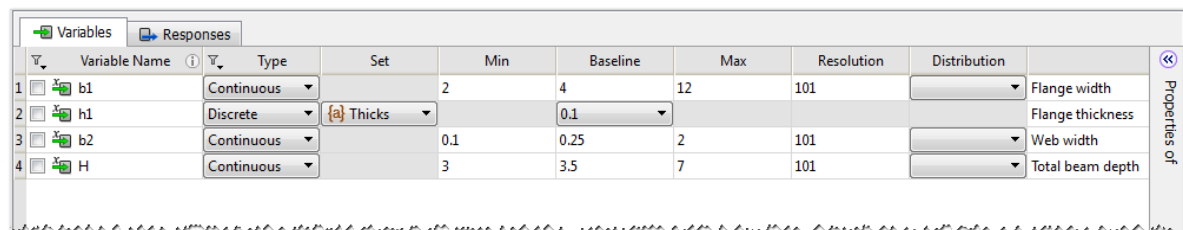


- Click **Close**. The *Sets Manager* closes and you return to the *Variables Manager*.
- In the **Set** field for variable **h1**, open the list box, and select **Thicks**:



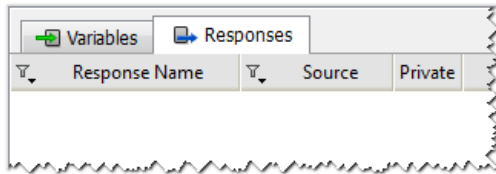
**Thicks** is defined as the set for discrete variable **h1**.

- Now that you have defined a discrete set for variable **h1**, HEEDS will let you enter values for **h1**. Locate the **Comment** field for **h1**, and enter: **Flange thickness**. Your completed project *Variables* definition should look like this:



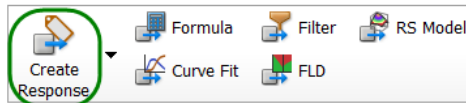
## Defining the project responses

- Click the **Responses** tab to open the *Project Responses Manager*.



- On the *Ribbon*, click **Create Response** three times to add three responses to your project.

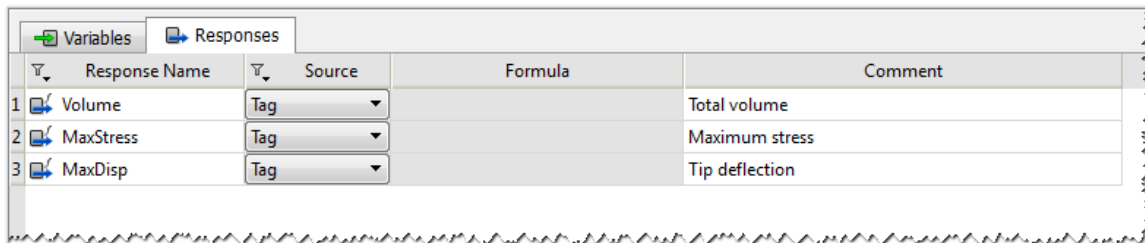
Alternatively, you can add the responses by opening the list box under **Create Response**, selecting **Create Multiple Responses**, and following the prompts in the *Wizard*.



- Rename and define the responses as follows:

Response Name	Source	Comment
<b>Volume</b>	Tag	Total volume
<b>MaxStress</b>	Tag	Maximum stress
<b>MaxDisp</b>	Tag	Tip deflection

Your completed *Responses* definition should look like this:



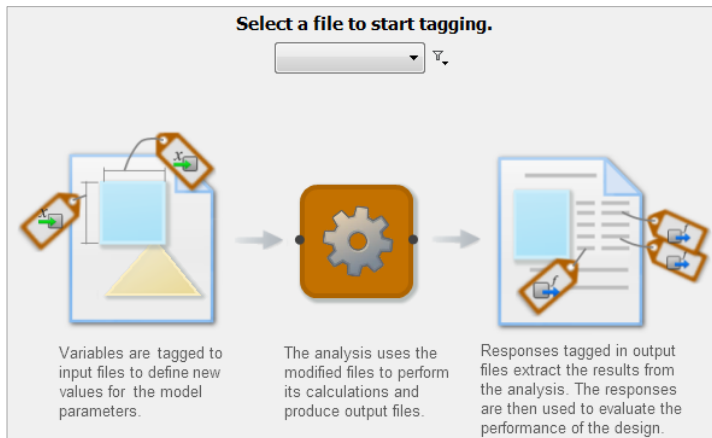
- Save your project.

## Step 4: Tagging the Input and Output Files

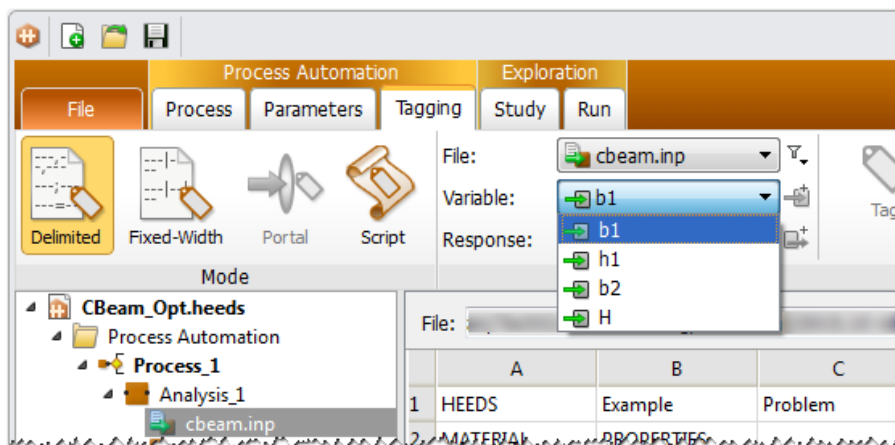
Next, we need to tag the values that correspond to our project variables and responses. We will begin with tagging the variables for our input file *Cbeam.inp*.

### Tagging the input file

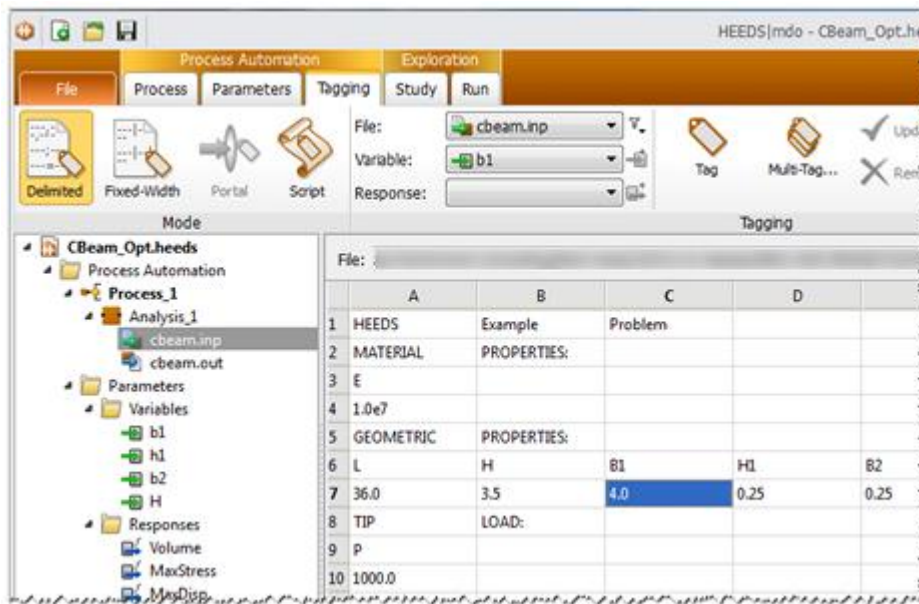
1. Click the **Tagging** tab. The *Tagging* page appears with the prompt: **Select a file to start tagging:**



2. Open the dropdown and select **Cbeam.inp**. The parsed input file is displayed.
3. On the *Ribbon*, open the dropdown for **Variable**, and select **b1**:



- Move to the *Variables* table and select the cell that contains the value of **B1**, which is **4.0**. The cell is located at **C7** on the table:



- Move back to the *Ribbon* and click **Tag**. Variable **b1** should appear in place of the value, **4.0**:

	A	B	C	D	E
1	HEEDS	Example	Problem		
2	MATERIAL	PROPERTIES:			
3	E				
4	1.0e7				
5	GEOMETRIC	PROPERTIES:			
6	L	H	B1	H1	B2
7	36.0	3.5	b1	0.25	0.25
8	TIP	LOAD:			
9	P				
10	1000.0				

- Tag the remaining variables that are on the **File** dropdown: **H**, **h1**, and **b2**. Tag the values as indicated:

	A	B	C	D	E
1	HEEDS	Example	Problem		
2	MATERIAL	PROPERTIES:			
3	E				
4	1.0e7				
5	GEOMETRIC	PROPERTIES:			
6	L	H	B1	H1	B2
7	36.0	3.5	b1	0.25	0.25
8	TIP	LOAD:			
9	P				
10	1000.0				

The variables you tagged will remain shaded. Your table should look like the one below:

	A	B	C	D	E
1	HEEDS	Example	Problem		
2	MATERIAL	PROPERTIES:			
3	E				
4	1.0e7				
5	GEOMETRIC	PROPERTIES:			
6	L	H	B1	H1	B2
7	36.0	H	b1	h1	b2
8	TIP	LOAD:			
9	P				

### Tagging the output file

1. On the *Ribbon*, open the **File** dropdown, and select **cbeam.out** to display the parsed output file.
2. Open the **Response** dropdown, and select **Volume**.
3. Move to the table on your screen. Scroll down on the table and click the cell at location **C30** that contains the value for **Volume**, which is **.9900E+02**:

	A	B	C
21	0.1000E+04		
22			
23			
24	OUTPUT	DATA:	
25			
26	TIP	DEFLECTION:	0.2657E+00
27			
28	MAX	STRESS:	0.1076E+05
29			
30	TOTAL	VOLUME:	0.9900E+02
31			

4. Move to the *Ribbon* and click **Tag**.

5. Tag the remaining responses, **MaxDisp** and **MaxStress**, in the same fashion. Use the **Response** dropdown on the *Ribbon* to select each response, and then tag the corresponding value as indicated:

	A	B	C
21	0.1000E+04		
22			
23			
24	OUTPUT	DATA:	
25			
26	TIP	DEFLECTION:	0.2657E+00
27			
28	MAX	STRESS:	0.1076E+05
29			
30	TOTAL	VOLUME:	0.9900E+02

Your tagged output file should look like the one below. The tagged values remain shaded:

	A	B	C
21	0.1000E+04		
22			
23			
24	OUTPUT	DATA:	
25			
26	TIP	DEFLECTION:	MaxDisp
27			
28	MAX	STRESS:	MaxStress
29			
30	TOTAL	VOLUME:	Volume

6. Save your project.

## Step 5: Defining the Study

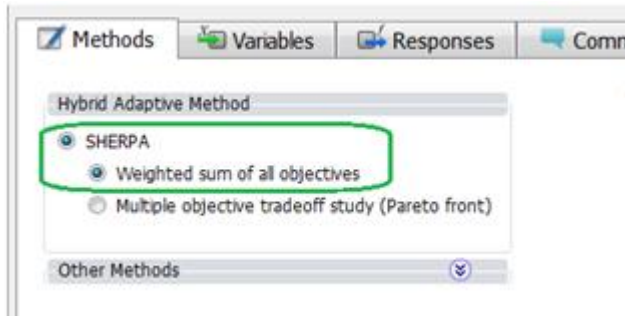
Sections one through four above have taken us through the *Process Automation* tasks of our project. The following section includes tasks at the *Exploration* level: *Study* and *Run*. We will begin with *Study*, and assemble the problem definition.

### Selecting the search method

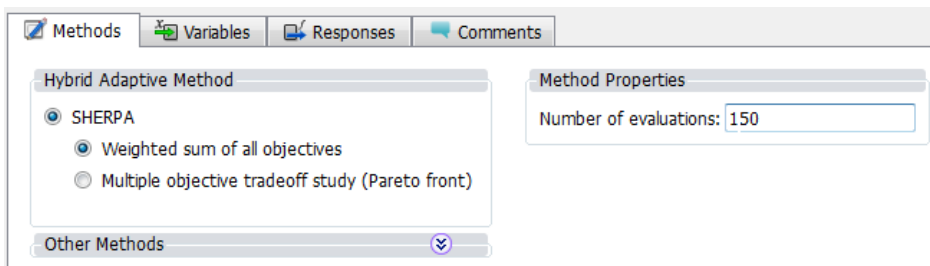
Here we'll select the search method for our study and we'll specify the number of evaluations we want HEEDS to run:

1. Select the **Study** tab on the tab bar. The *Study* page opens to the *Study Manager*:
2. Click the **Methods** tab on the *Study Manager*.

3. Select the search method that you want to use for your study. For this example, choose the default, **SHERPA - weighted sum of all objectives**:



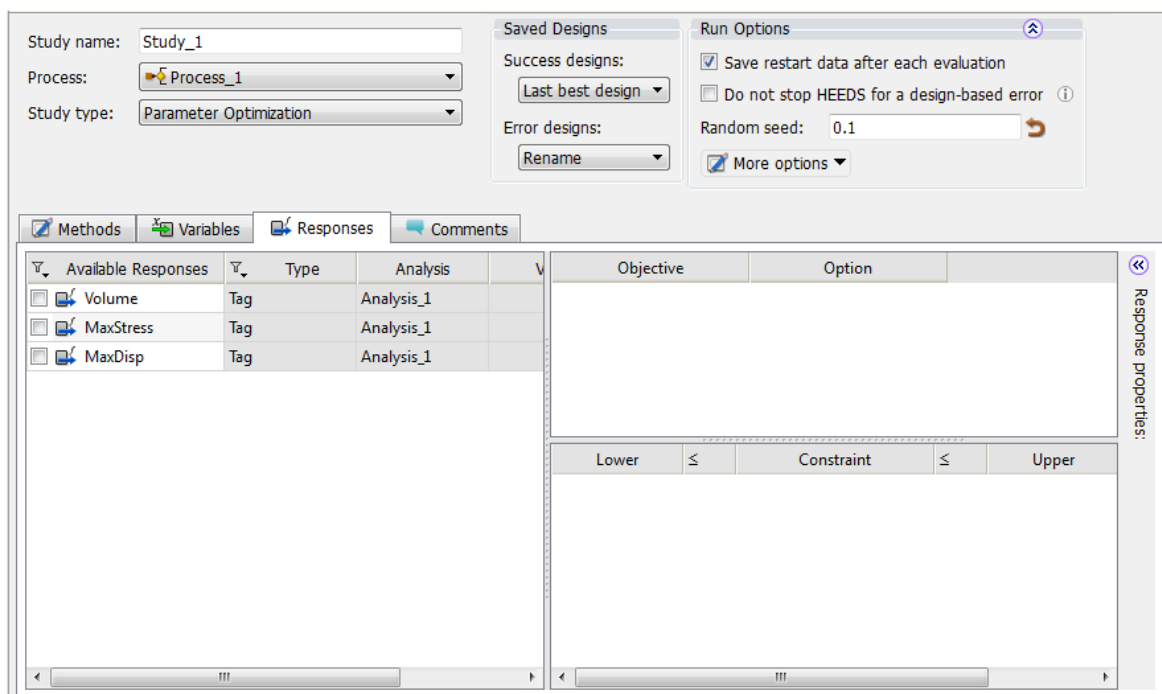
4. Locate the field: **Number of evaluations**. For this example, use the default value of **150**. When we run the study, HEEDS will complete 150 evaluations. The completed *Method definition* for your study should look like this:



### Completing the response definition

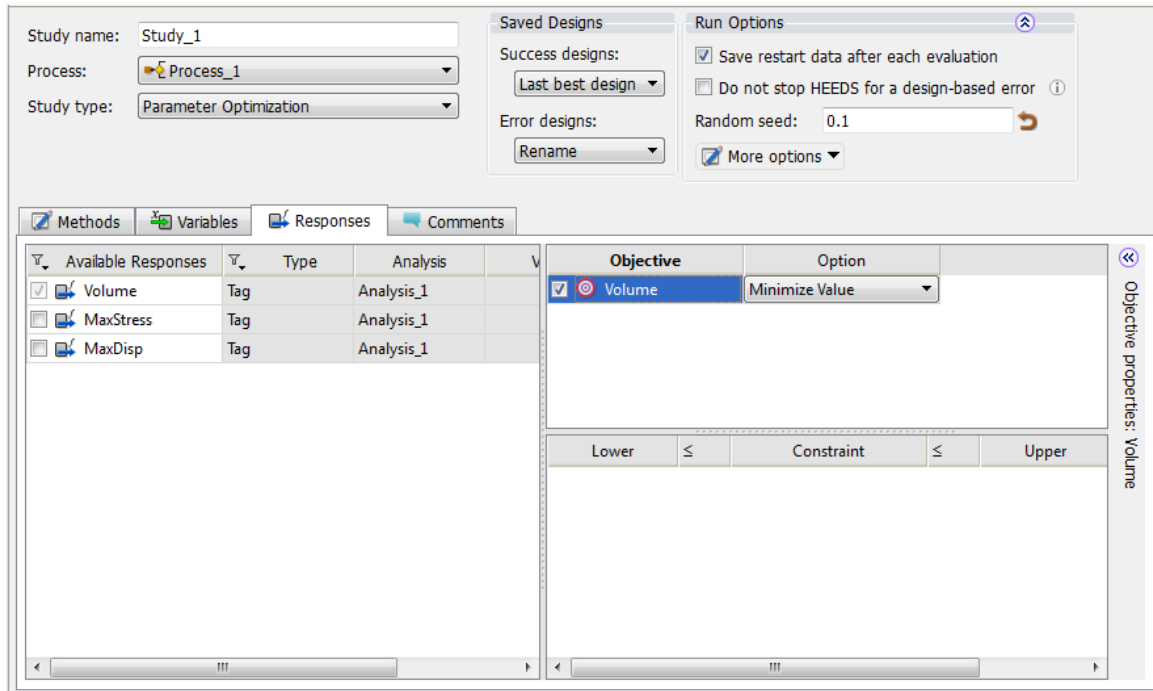
In the function example, you had only one response, which was the objective. In this example, you also have two constraint responses:

1. On the *Study Manager*, click the **Responses** tab. All available responses for the analysis are listed here. For this project, we have three:

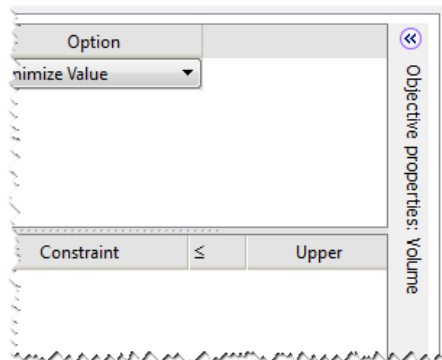


- On the *Ribbon*, click **Add Objective**, and then select **Volume** from the dropdown. **Volume** appears in the *Study Manager* as an Objective.

Alternatively, you can add the objectives on the *Study Manager* by dragging and dropping them from the *Responses* list into the *Objective View*.



- In the *Objective View*, open the **Option** list, and select **Minimize Value**.
- To the right of **Option**, click to open the panel: **Objective properties: Volume**:



The *Objective properties* panel opens.

- Locate the **Normalization factor** field, and enter **100.00**. (The volume for this example ranges between 50 and 500.)

- Leave the remaining fields set to default and close the panel.
- On the *Ribbon*, click **Add Constraint**, and select **MaxStress** from the dropdown. **MaxStress** appears in the *Study Manager* as a constraint.

Alternatively, you can add the constraints on the *Study Manager* by dragging and dropping them from the *Responses* list into the *Constraint View*.

- On the *Study Manager*, select the cell  $\leq$  (*upper*) for the *Constraint Max Stress*, and enter **5000** for the value.



**Note:** *MaxStress* and *MaxDisp* are *Constraint responses*. This means that we need to set a limit for them. The *Option  $\leq$*  indicates that the response must be less than or equal to the amount we specify.

- Add a second constraint. On the *Ribbon*, click **Add Constraint**, then select **MaxDisp** from the dropdown:

The constraint **MaxDisp** appears in the *Study Manager*.

- From the *Study Manager*, select the cell  $\leq$  (*upper*) for the *Constraint MaxDisp* and enter **0.1** as the value.

Your completed *Responses* definition should look like this:

Available Responses	Type	Analysis
<input checked="" type="checkbox"/> Volume	Tag	Analysis_1
<input checked="" type="checkbox"/> MaxStress	Tag	Analysis_1
<input checked="" type="checkbox"/> MaxDisp	Tag	Analysis_1

Objective	Option
<input checked="" type="checkbox"/> Volume	Minimize Value

Lower		Constraint		Upper
	≤	<input checked="" type="checkbox"/> MaxStress	≤	5000
	≤	<input checked="" type="checkbox"/> MaxDisp	≤	0.1

**Note:** The Objective, Volume, and the Constraints, MaxDisp and MaxStress have darkened checkmarks next to them to indicate that you have included them in the study. Unchecking a constraint or objective would exclude it from the study, without deleting it.

11. Save your project.

### Specifying which designs to save

The last step to completing your *Study* definition is to specify which designs you want HEEDS to save.

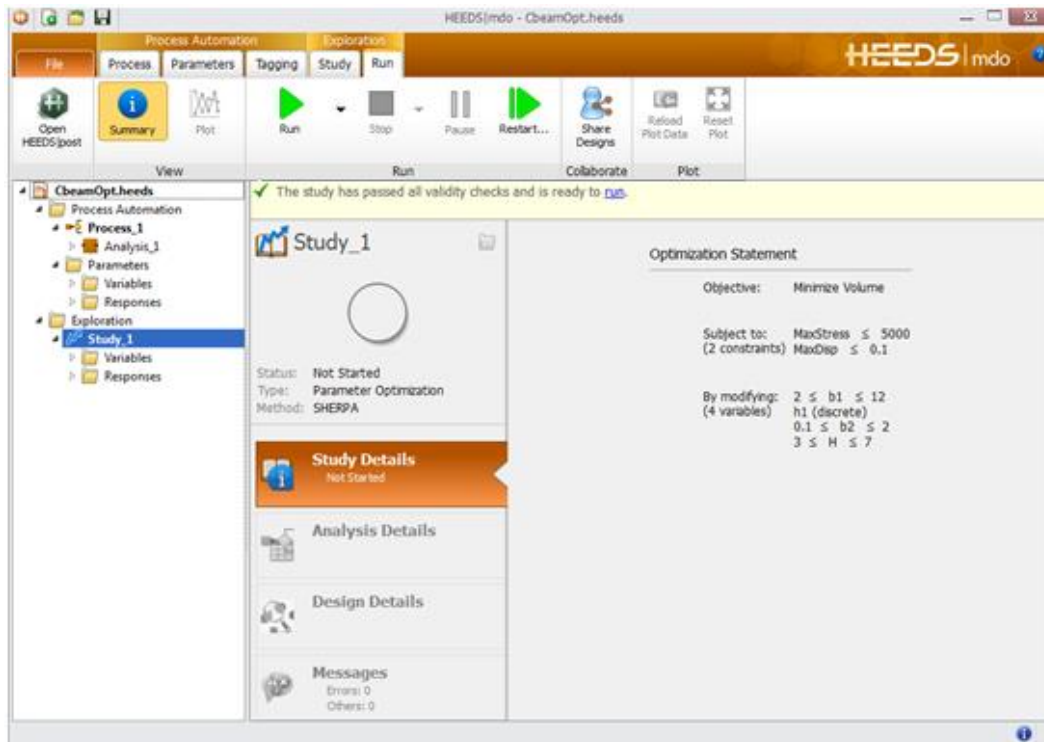
1. In the *Study Manager*, locate **Saved Designs** and use the default **Last best design**:

2. Use the default values for the remaining options. Your completed *Study* definition should look like this:

3. Save your project.

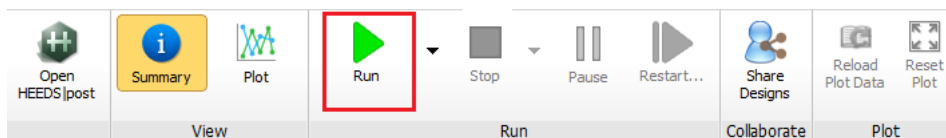
## Step 6: Running the Study

You are ready to execute your study. Select **Run** from the tab bar. The *Run* page appears with the summary of your *Study Details* and *Optimization Statement*:



### To run the study

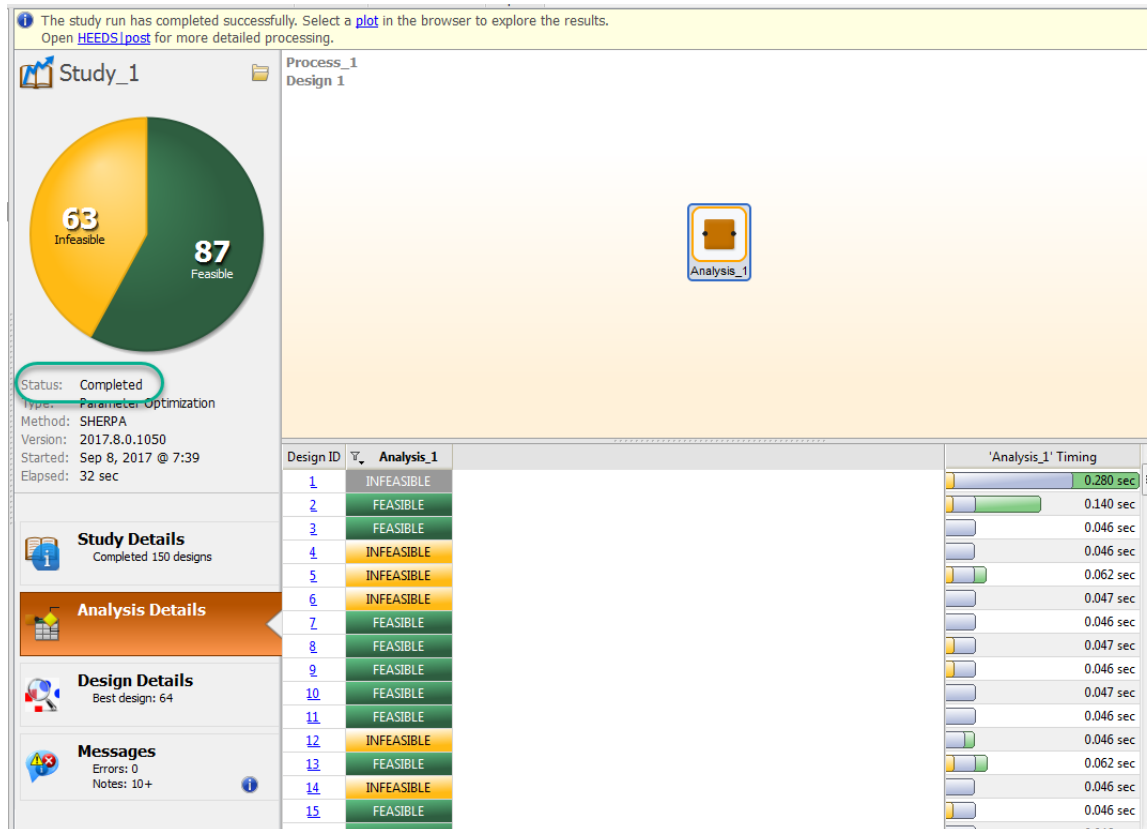
Click **Run** to execute the study.



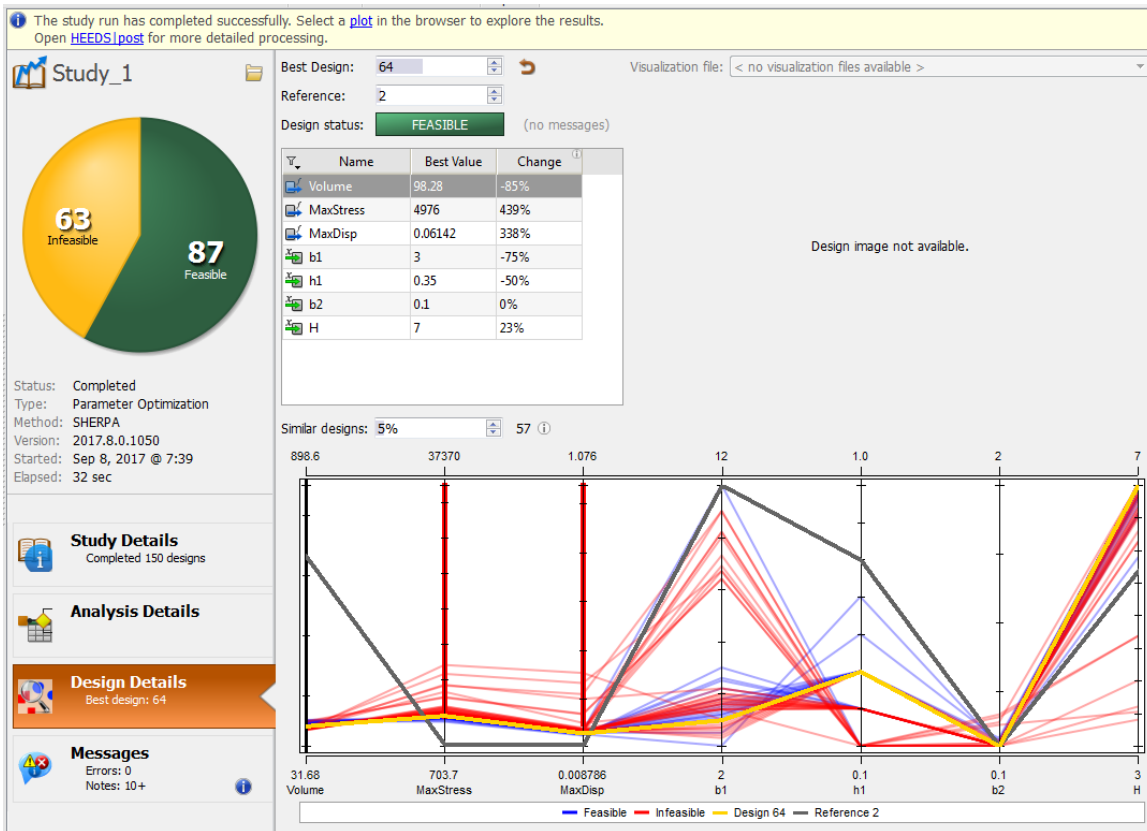
## Monitor your study

HEEDS provides you with a **Summary** and **Plot** option on the *Ribbon* to monitor design evaluations in progress. Our C-Beam study will run quickly, but we'll have a few moments to review the progress of the run and the *Objective History* plot.

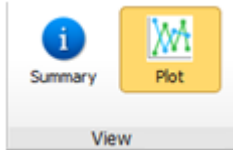
1. As the study runs, watch the pie chart as it updates.
2. Click on the **Analysis Details** tab. The information on this page will update and show **Completed** when HEEDS is finished with a design. When MDO imports the results the **Completed** are distributed to **Feasible** or **Infeasible**. The **Timing** column shows the timing of the selected analysis and can be used to identify a bottleneck in the analysis.



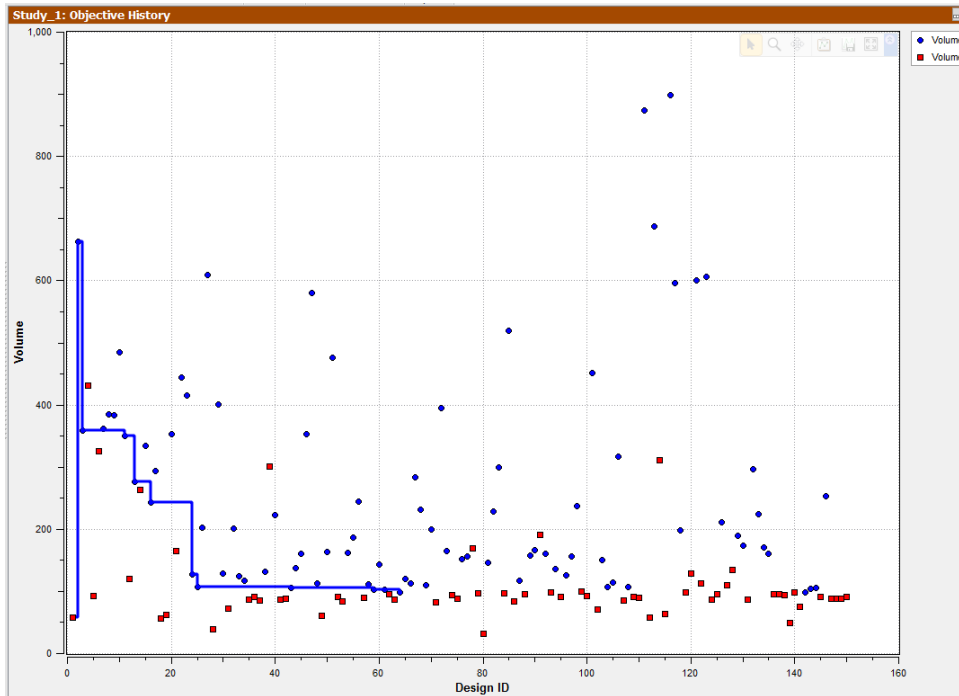
- Click the **Design Details** tab. When the study is complete, you can review the information here to compare the best design with the baseline. The parallel plot on this page shows other designs that are similar to the best design for the selected parameter.



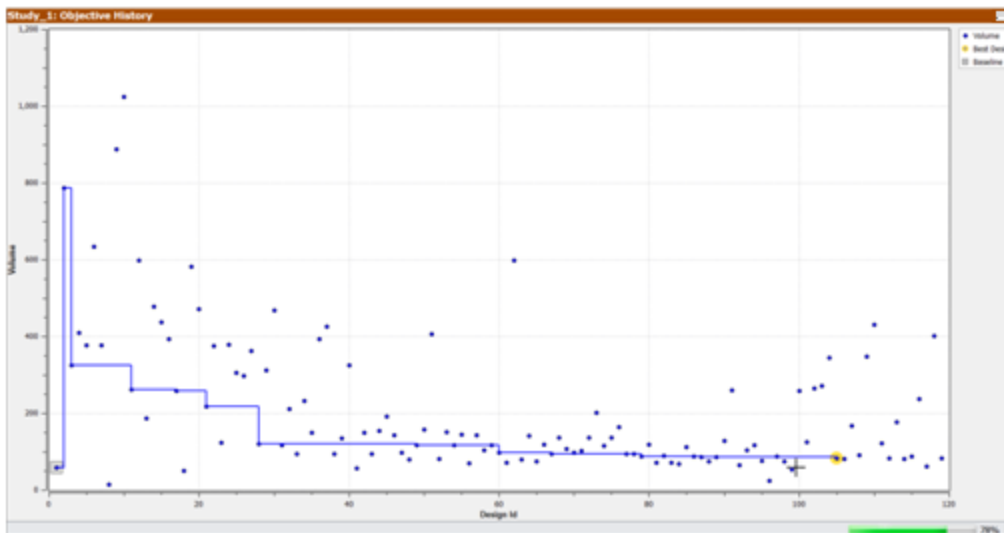
4. Next let's view the *Objective History* plot for the study. On the *Ribbon*, click **Plot**.



The **Plot** command displays the **Objective History** plot as the default:



5. On the *Ribbon*, click **Reload Plot Data**. The *Objective History* plot updates with the most current data available for the run:



6. HEEDS generates additional default post-processing plots for you to explore: **Parallel Data**, **Design Table**, **Constraint Violations**, **Correlation**, or **Performance History**. To view any of these, click **Plot** on the *Ribbon*, then move to the **Plots** node on the tree and make a selection.



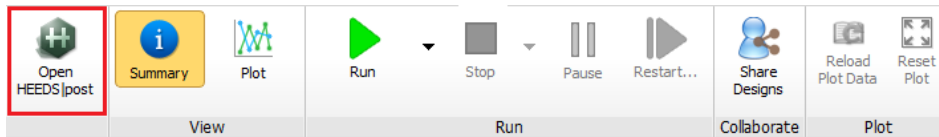
**Note:** To see an updated plot for any plot type, click the **Reload Plot Data** command on the *Ribbon*.

## Step 7: Viewing Results Using HEEDS POST

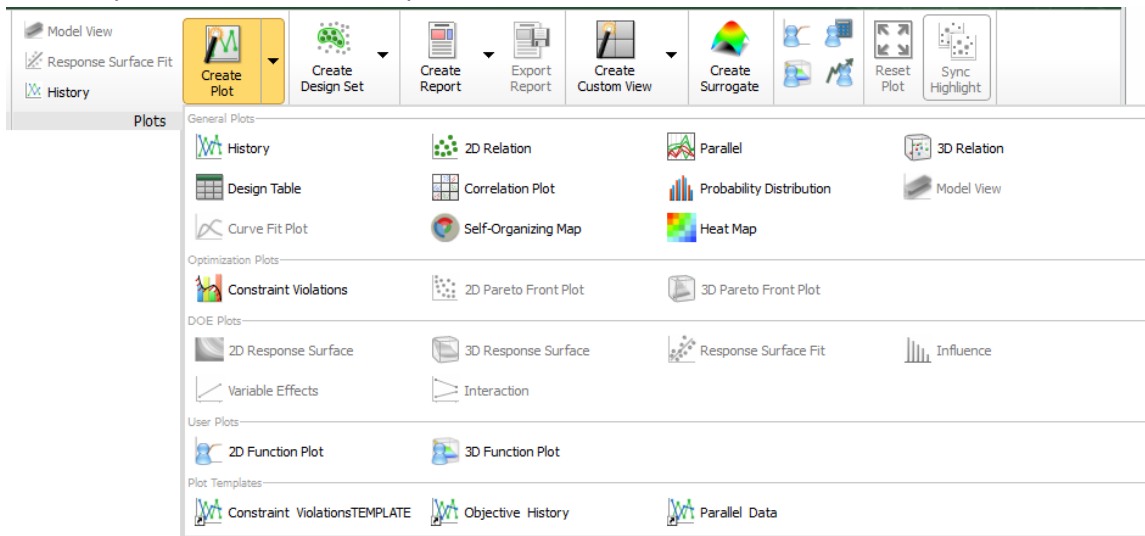
HEEDS POST gives you access to all of the plots that are available in HEEDS MDO. In addition, HEEDS POST provides you with advanced post-processing capabilities to review results. In the following section, we will use HEEDS POST to explore several of the plot types available for this run: **History**, **2D Relation**, **3D Relation**, **2D Bubble**, **Parallel**, **Design Table**, and **Constraint Violation**.

### Generate plots using HEEDS POST

1. On the HEEDS MDO *Ribbon*, click **Open HEEDS|post**:

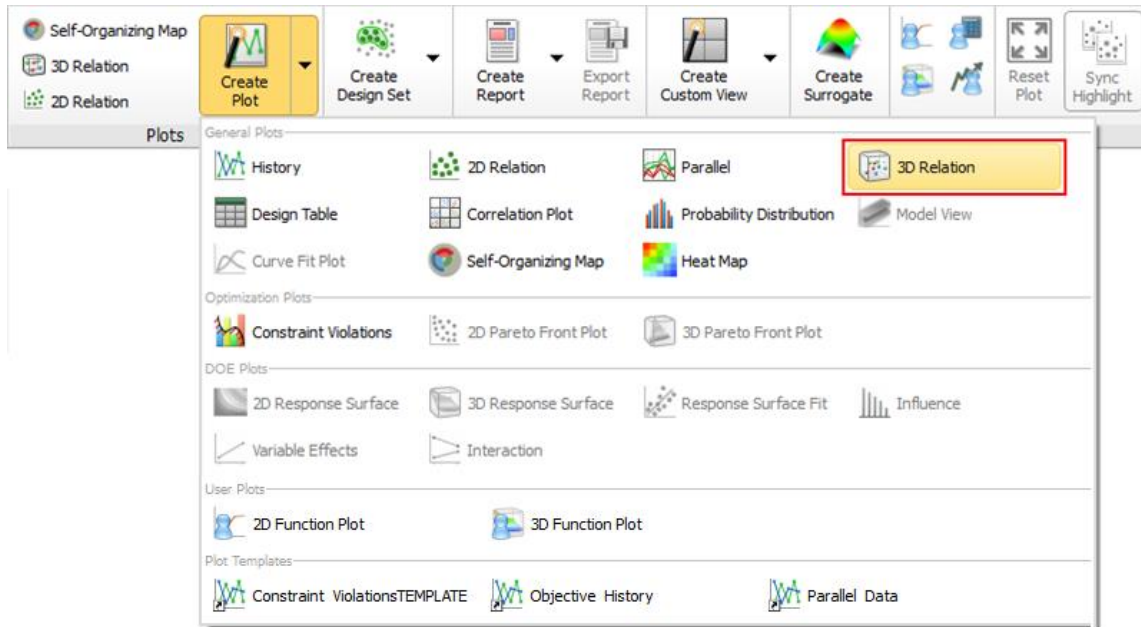


2. You have several options when plotting data in HEEDS POST. You can use the default plots from the **Plots** node on the tree, or you can make new plots using the *Plot* tools and *Create Plot Wizard* on the *Ribbon*. For this example, we'll use the default plots and the tools from the *Plots* section of the *Ribbon*:



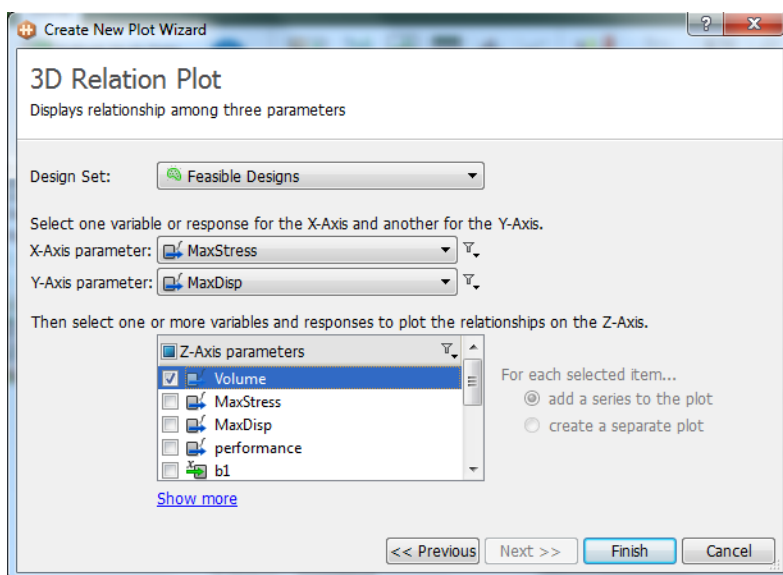
## Create a 3D relation plot

1. On the *Ribbon*, select the **3D Relation** plot tool:

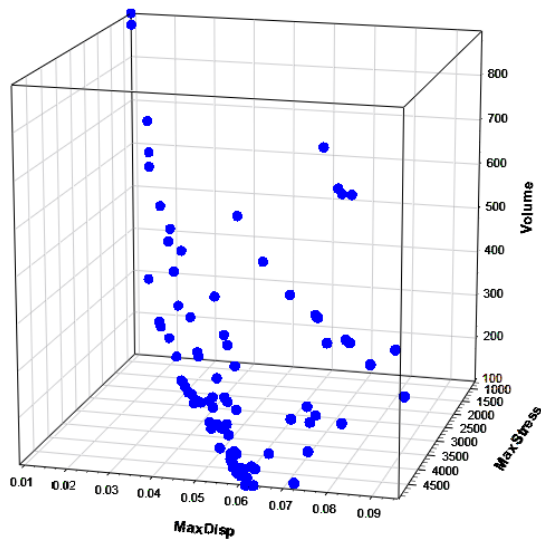


2. The *Create New Plot Wizard* appears. Enter the following in the *Wizard*:
  - Locate the **Design Set** dropdown and select: **Feasible Designs**.
  - Locate the **X-Axis Parameter** dropdown and select **MaxStress**.
  - Locate the **Y-Axis Parameter** dropdown and select **MaxDisp**.
  - Locate the **Z-Axis parameters** list and select **Volume**.

Your *Wizard* should look like this:



- Click **Finish** to generate the plot:

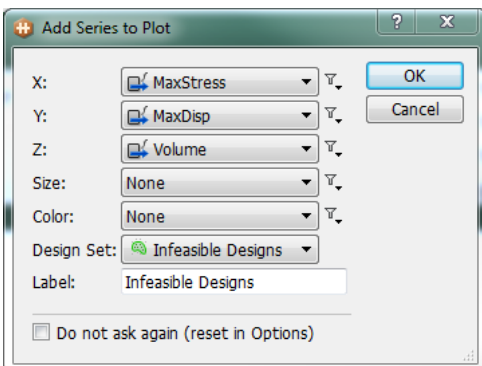


Since only feasible solutions were plotted, all of the data points appear in the feasible region of the *3D Relation* plot.

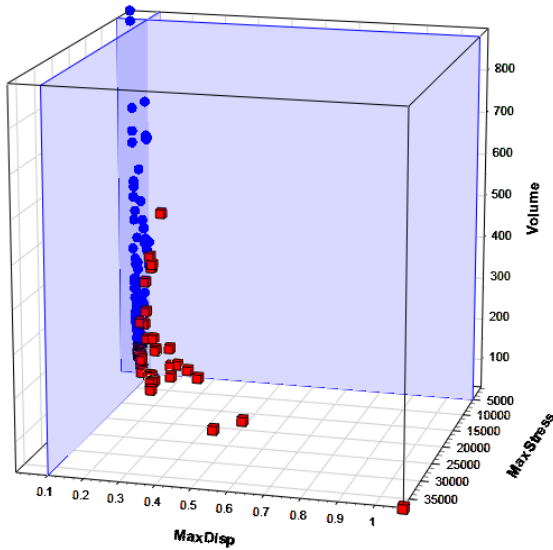
#### ► Add infeasible designs to the 3D relation plot

- Click the **+** button in the **Series Data** area of the tool ribbon to add another series to the plot.
- In the *Add Series to Plot* dialog box enter the following:
  - For the **X-Axis Parameter**, select **MaxStress** from the dropdown.
  - For the **Y-Axis Parameter**, select **MaxDisp** from the dropdown.
  - For the **Z-Axis Parameter**, select **Volume** from the dropdown.
  - For the **Design Set**, select **Infeasible Designs** from the dropdown.
  - In the **Label** field, enter **Infeasible Designs**.

The *Add Series to Plot* dialog box should look like this:

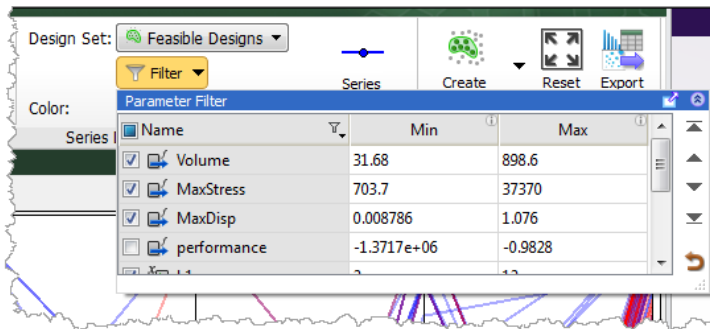


- Click **OK** to add the series:

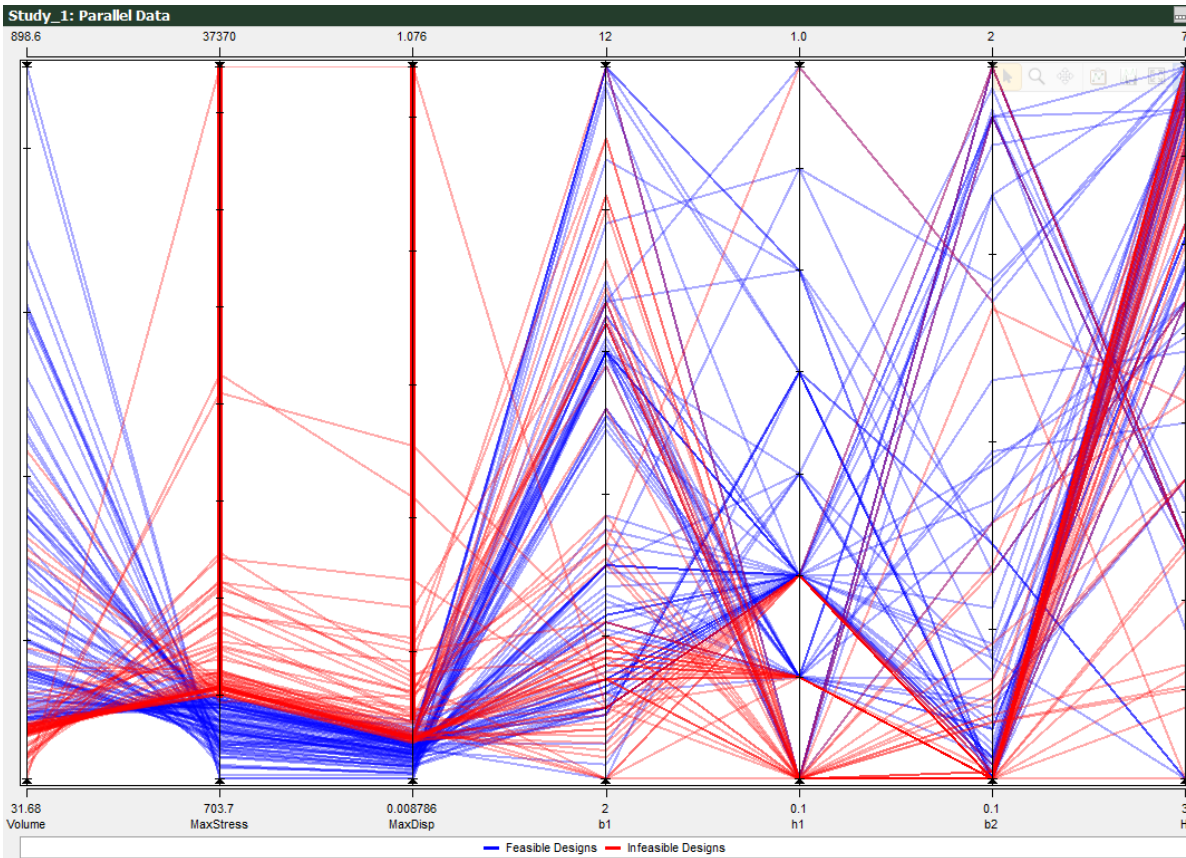


**Note:** The planes represent the constraint boundaries.

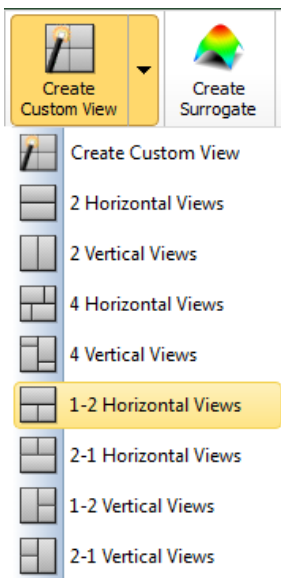
- Select the **Parallel Data** plot that was created by default. Select the **Filter** button on the **Plot** tool ribbon tab. The filter panel lets you specify which designs are to be included in the plotted data:



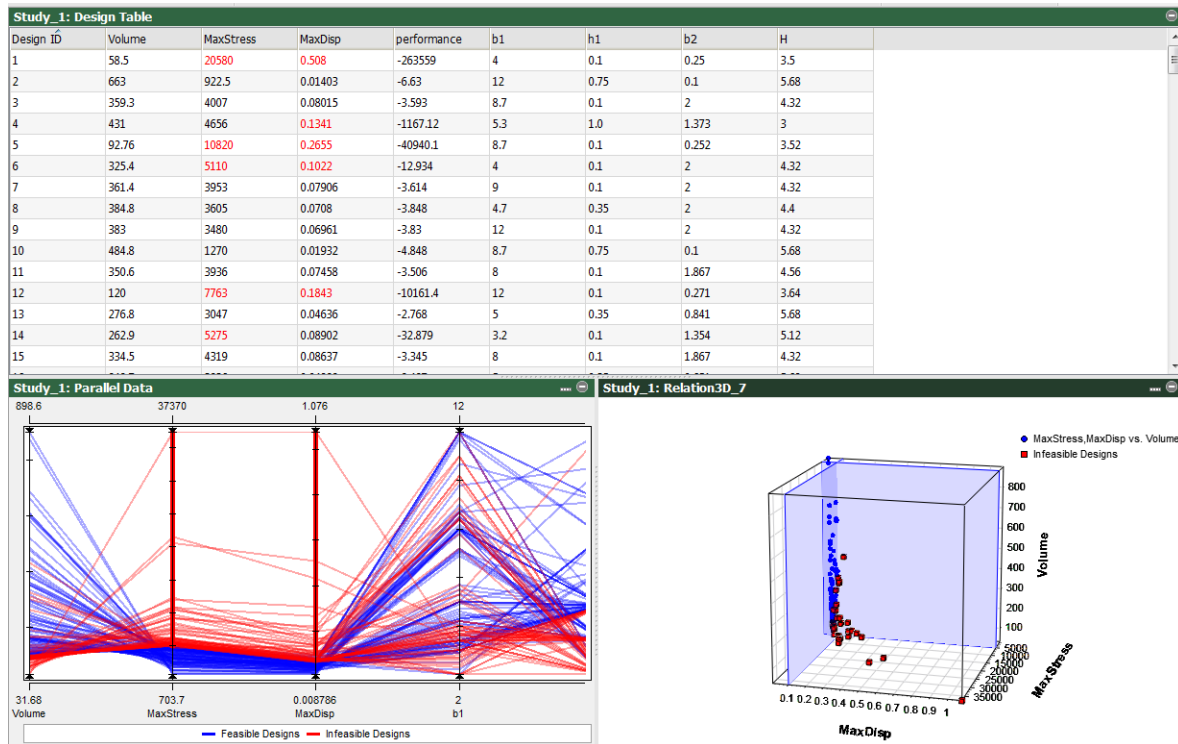
By default, the feasible and infeasible designs are plotted for us. The feasible designs are shown in *blue*, and the infeasible designs in *red*. In addition, the baseline design is shown in *black* and the optimized design in *yellow*. For responses containing constraints, vertical *red* bars indicate the constraint values. Designs passing through the constraint bars are infeasible designs:



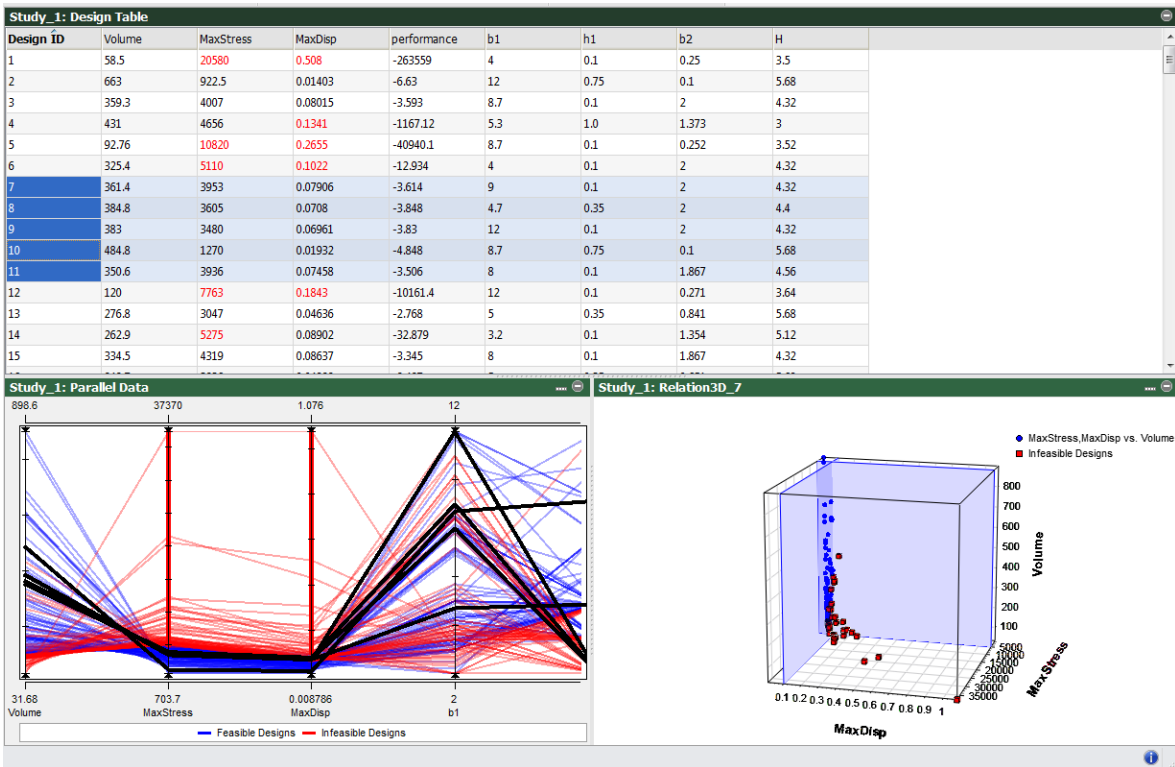
5. Create a *Plot View* that shows the **Parallel Data** with the **3D Relation** plot and **Design Table**. To do this select the appropriate *Plot View* on the *Ribbon* or create a custom view:



6. Select the appropriate plots such that the **Design Table** is on top and the **Parallel Data** and **3D Relation** plot are on the bottom:



7. Select a group of designs in the **Design Table**, and notice that because **Sync Highlight** is activated on the **Ribbon**, the same designs are selected in the other plots in the **Plot View**:



## Step 8: Reviewing the Log Files

1. In the **Ribbon**, click **Return to HEEDS MDO**.
2. In the **Run** tab, click the **Summary** button.
3. In the **Study Summary**, click the **Messages** tab.

This is where you can review the study and analysis log files and learn about any errors that exist in your study.



## Example 3: 3-Bar Truss Mechanism

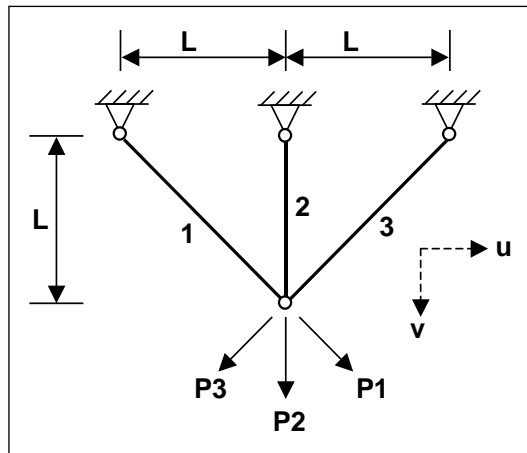


Fig. 1. A 3-bar truss

The planar 3-bar truss shown in fig. 1 is to be designed to minimize the total weight of the system. The cross-sectional area of each member is varied, and constraints are placed on the displacements, the stresses, the fundamental frequency, and the axial internal forces.

Three analyses are used to analyze the performance of each potential design. The first analysis calculates the internal forces, stresses, and displacements for each truss member under the applied loads. The second analysis evaluates the natural frequency of the system, and the third checks for buckling within each of the three members.

This example illustrates the use of multiple analysis tools and the ability of HEEDS MDO to use the output of one analysis as input data for another analysis within the same process.

### The problem

The current optimization problem is defined as follows:

<b>Minimize:</b>	Total Weight	
<b>Such that:</b>	$U_{min}$	$\geq -0.005$ inches
	$V_{min}$	$\geq -0.005$ inches
	$U_{max}$	$\leq 0.005$ inches
	$V_{max}$	$\leq 0.005$ inches
	$\sigma_{1min}$	$\geq -5000.0$ psi
	$\sigma_{2min}$	$\geq -5000.0$ psi
	$\sigma_{3min}$	$\geq -5000.0$ psi
	$\sigma_{1max}$	$\leq 20000.0$ psi
	$\sigma_{2max}$	$\leq 20000.0$ psi
	$\sigma_{3max}$	$\leq 20000.0$ psi
	freq	$\geq 2500$ Hz
	Buckle_1*	$\leq 1$
	Buckle_2*	$\leq 1$
	Buckle_3*	$\leq 1$

\*  $Buckle_i = (\text{Axial Force of member } i) / (\text{Force required to buckle member } i)$

This example was taken from *Introduction to Optimum Design*, Second Edition, by J. S. Arora (California: Elsevier Academic Press, 2004)

**By varying:**  $0.1 \text{ in}^2 \leq \text{Area1} \leq 10 \text{ in}^2$   
 $0.1 \text{ in}^2 \leq \text{Area2} \leq 10 \text{ in}^2$   
 $0.1 \text{ in}^2 \leq \text{Area3} \leq 10 \text{ in}^2$

The steps needed to complete an entire analysis for a given set of bar areas are shown in fig. 2.

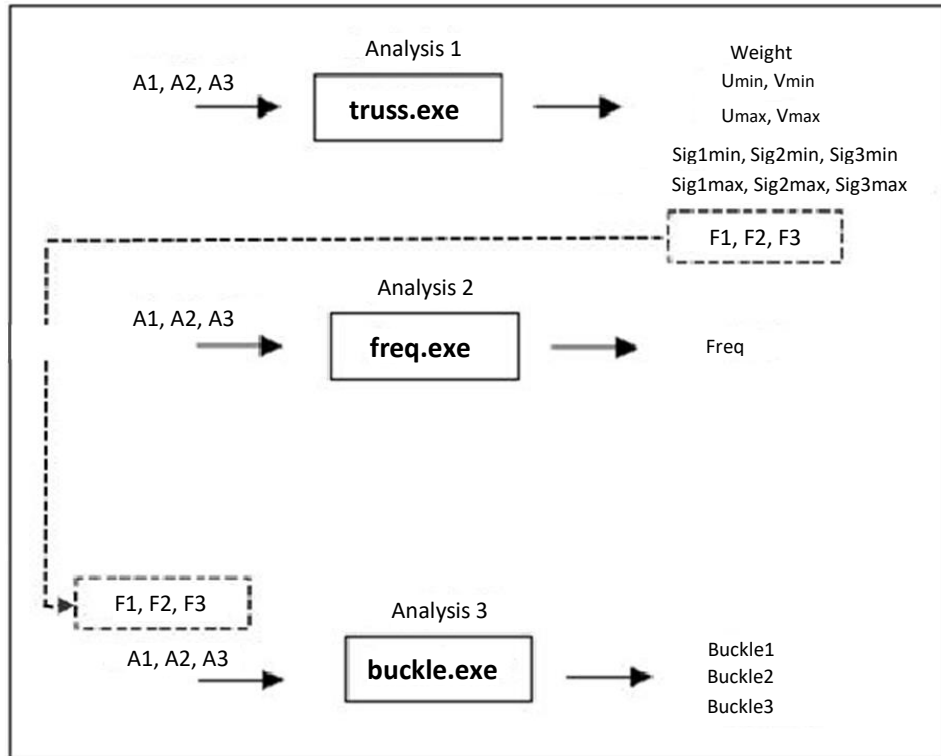


Fig. 2. Graphical flow of each analysis.

### The baseline design

This example uses three input files and three analysis tools. The files for our example are found in the **Truss** folder in the **Examples** directory:

- The default path for Windows is:  
**C:\HEEDS\MDO\Version\Examples\Truss**
- The default path for Linux and UNIX is:  
**/opt/HEEDS/MDO/Version/Examples/Truss**

If you installed HEEDS MDO in a different directory, look for the files there instead.

### Input files

The input files and their contents are as follows:

File Name	Description
<b>truss.in</b>	E, $\rho$ <i>Area1, Area2, Area3</i> L Magnitude and angle of load for each load case
<b>freq.in</b>	E, $\rho$ <i>Area1, Area2, Area3</i> L
<b>buckle.in</b>	E, L, $\beta$ <i>Area1, Area2, Area3</i> <i>Buckle1, Buckle2, Buckle3</i> ( $\rho$ = weight density, $\beta = I / A^2$ )

### Analysis tools

The three analysis tools are as follows:

File Name	Description
<b>truss.bat</b>	Performs a static analysis using <b>truss.in</b> .
<b>freq.bat</b>	Performs an eigenvalue analysis using <b>freq.in</b> .
<b>Buckle.bat</b> (Windows) <b>Buckle</b> (Linux)	Performs a buckling analysis using <b>buckle.in</b> .

### Output files

The three output files are as follows:

File Name	Description
<b>truss.out</b>	Output from truss.bat
<b>freq.out</b>	Output from freq.bat
<b>Buckle.out</b>	Output from buckle.bat

---

## Step 1: Starting a New Project

Before you create your project, create a directory on your system specifically for your HEEDS project. Be sure to choose a location where you have write permissions.

1. Click the **File** tab to display the *File* page. Select **New**. A new project appears on your screen.
2. Click the **File** tab, and then select **Save**. The *Save Project As* dialog appears.
3. Browse to the directory that you have set up to save this project, and enter **TrussEx** as the project name.
4. Click **Save**. The file is saved with a **.heeds** extension.

---

## Step 2: Defining the Process

We begin our project at the level of *Process Automation*. Our first task is to identify the analyses of our *Process*. We will identify three analyses, each one with a different analysis tool, input file, and output file.

## Identifying the analyses

1. Select the **Process** tab. The *Process* page appears.
2. Click the **Analysis\_1** tool from the *Process View* on your screen:

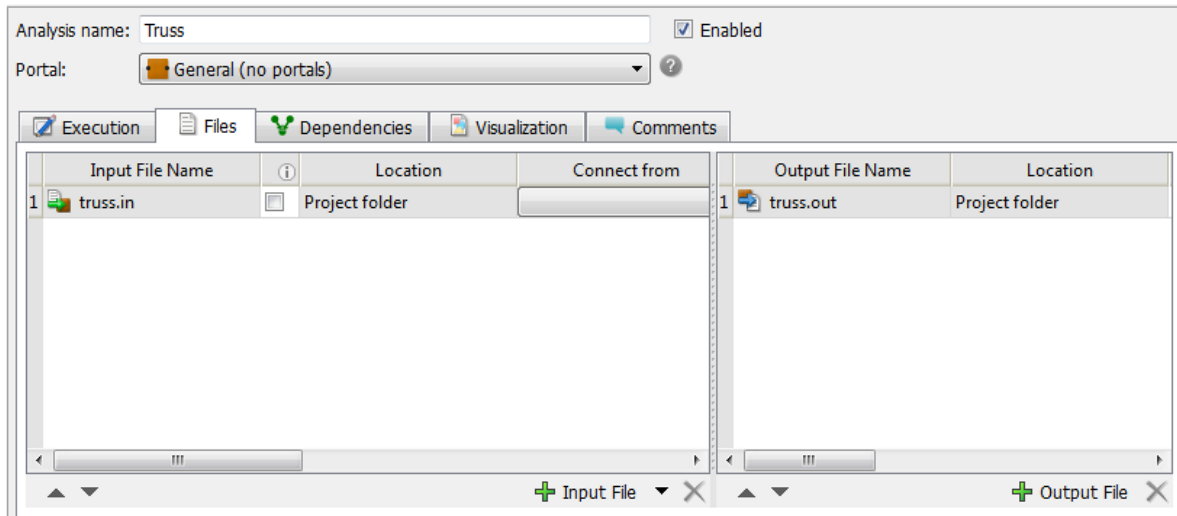


The *Analysis Manager* opens with the **Execution** tab selected:

3. In the **Analysis name** field, rename **Analysis\_1** to **Truss**.
4. In the **Execution command** field, click the browse icon to select **truss.bat**.
5. Next, add the input file and output file. Click the **Files** tab and then click **+ Input File**:

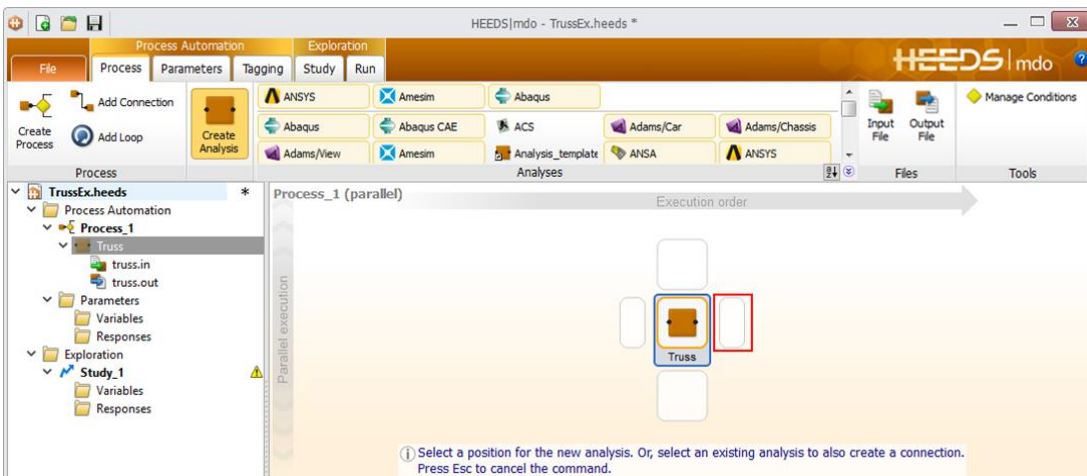
6. Navigate to the **Truss** folder, select **truss.in**, and click **Open**. Your analysis definition includes **truss.in** as an input file.
7. Now click **+ Output File**.
8. Navigate to the **Truss** folder, select **truss.out**, and click **Open**. Your analysis definition includes **truss.out** as an output file.

Your *Analysis Manager* should look like this:

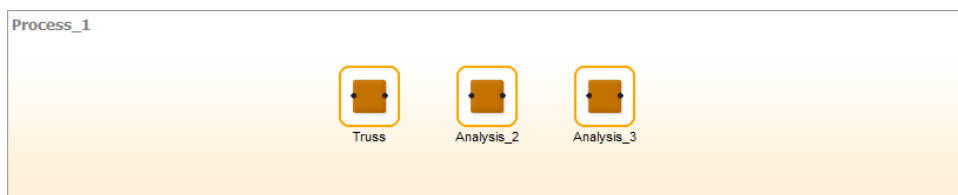


9. Now let's add the second and third analysis to our *Process*. On the *Ribbon*, select **Create Analysis**.

A prompt appears in the *Process View* asking us to identify the execution order of our analyses. We'll select the position to the right of **Truss** so that **Analysis\_2** follows **Truss**.



10. Add a third analysis to *Process\_1*, and select the order so that **Analysis\_3** follows **Analysis\_2**. When you finish, your *Process View* should look like this:



11. We will rename and define the new analyses in the next steps. Use the information in the following table to define **Analysis\_2**:

<b>Analysis Name</b>	<b>Freq</b>
<b>Portal</b>	<b>General (no portals)</b>
<b>Execution Command</b>	<b>freq.bat</b>
<b>Input File</b>	<b>freq.in</b>
<b>Output File</b>	<b>freq.out</b>

Your completed definition for *Freq* should look like this:

Analysis name: Freq ☒ Enabled

Portal: General (no portals) ?

Execution Files Dependencies Visualization Comments

	Input File Name	Location	Connect from		Output File Name	Location
1	freq.in	Project folder		1	freq.out	Project folder

+ Input File -x + Output File -x

12. Use the information in the following table to define **Analysis\_3**:

<b>Analysis Name</b>	<b>Buckle</b>
<b>Portal</b>	<b>General (no portals)</b>
<b>Execution Command</b>	<b>buckle.bat</b>
<b>Input File</b>	<b>buckle.in</b>
<b>Output File</b>	<b>buckle.out</b>

Your completed definition for *Buckle* should look like this:

Analysis name:  ☒ Enabled

Portal:  ?

Execution | Files | Dependencies | Visualization | Comments

	Input File Name	Location	Connect from		Output File Name	Location
1	buckle.in	Project folder		1	buckle.out	Project folder

+ Input File - X + Output File - X

13. Save your project.



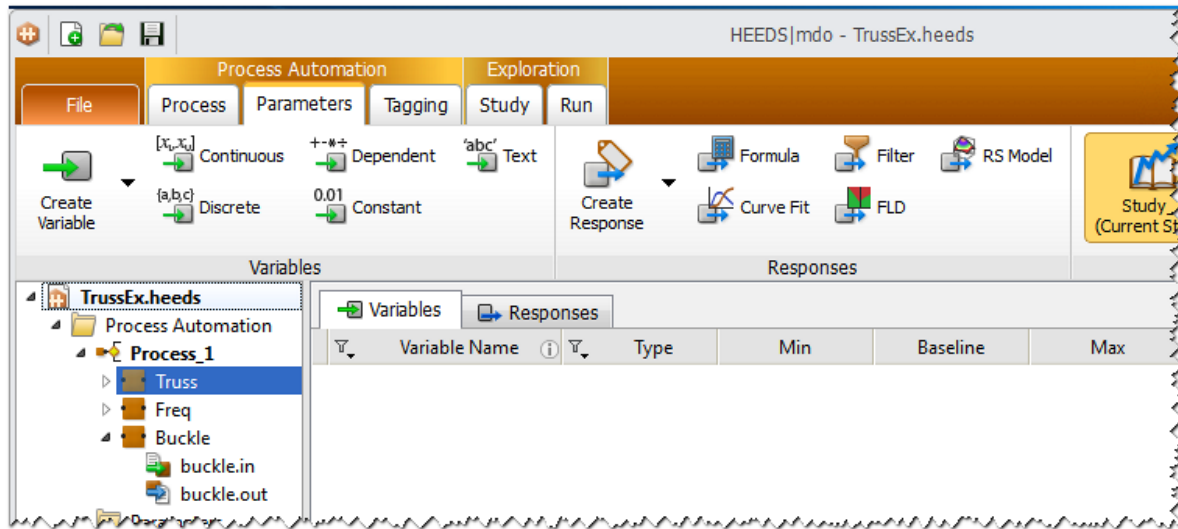
**Note:** The analyses will be performed in the order in which they appear in the Process View. If you want to change the position of an analysis, you can select the analysis in the Process View, and then drag and drop it into the desired position.

### Step 3: Defining the Parameters

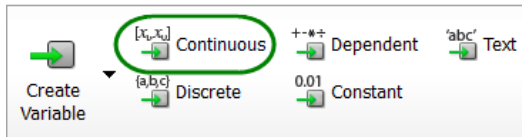
The next task of *Process Automation* is defining *Parameters*. This task includes identifying the variables and responses that we want to include in our project. Our *Truss* example has a more complex set of project variables and responses than the previous examples.

## Defining the project variables

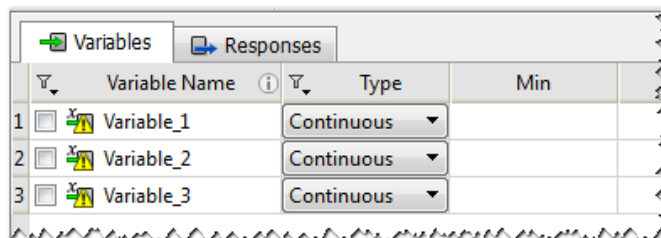
1. Click the **Parameters** tab. The *Parameters* page opens to the *Variables Manager*. You can also define the parameters on the **Study** tab.



2. Locate the **Continuous** variable tool on the *Ribbon*, and click three times to add three continuous variables to your project.



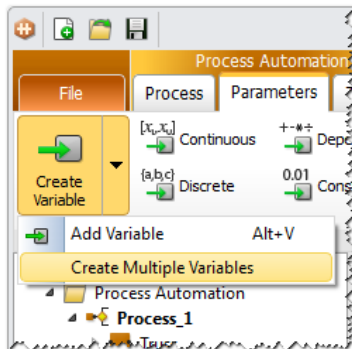
Your variables should look like this:



- Rename and define your variables as indicated below. To save time, you can copy the values from one variable, and then paste them directly into the corresponding columns for the other two variables in the *Variables Manager*.

Variable Name	Min	Baseline	Max	Resolution	Comment
A1	0.1	5	10	991	Area of bar 1
A2	0.1	5	10	991	Area of bar 2
A3	0.1	5	10	991	Area of bar 3

Alternatively, you can use **Create Multiple Variables**, which is on the *Ribbon* under **Create Variable**, to create many variables with the same definition:



- Check to see that your variables look like this:

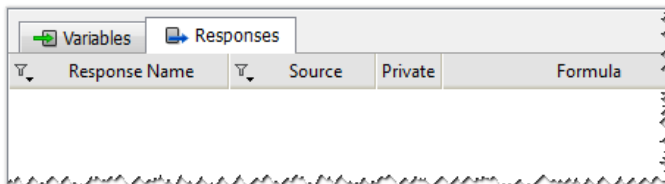
 A screenshot of the 'Variables' tab in the Variables Manager. It shows a table with three variables: A1, A2, and A3. Each variable is of type 'Continuous' and has the same values for Min (0.1), Baseline (5), Max (10), and Resolution (991). The 'Distribution' column has dropdown menus, and the 'Comment' column contains 'Area of bar 1', 'Area of bar 2', and 'Area of bar 3' respectively.
 

	Variable Name	Type	Min	Baseline	Max	Resolution	Distribution	
1	A1	Continuous	0.1	5	10	991		Area of bar 1
2	A2	Continuous	0.1	5	10	991		Area of bar 2
3	A3	Continuous	0.1	5	10	991		Area of bar 3

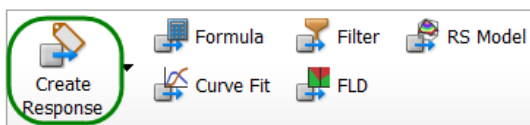
- Save your project.

### Identifying the project responses

- From the *Parameters* page, click the **Responses** tab to open the *Responses Manager*:



- Use the **Create Response** tool on the *Ribbon* to add **18** responses to your project:



3. Rename and define the responses using the information in the table below. To save time, you can copy and paste the information:

Response Name	Source	Comment
<b>Weight</b>	Tag	Total weight
<b>Umin</b>	Tag	Min disp in horizontal direction
<b>Vmin</b>	Tag	Min disp in vertical direction
<b>Umax</b>	Tag	Max disp in horizontal direction
<b>Vmax</b>	Tag	Max disp in vertical direction
<b>Sig1min</b>	Tag	Min stress in bar 1
<b>Sig2min</b>	Tag	Min stress in bar 2
<b>Sig3min</b>	Tag	Min stress in bar 3
<b>Sig1max</b>	Tag	Max stress in bar 1
<b>Sig2max</b>	Tag	Max stress in bar 2
<b>Sig3max</b>	Tag	Max stress in bar 3
<b>F1</b>	Tag	Axial force of bar 1
<b>F2</b>	Tag	Axial force of bar 2
<b>F3</b>	Tag	Axial force of bar 3
<b>Freq</b>	Tag	Fundamental frequency
<b>Buckle1</b>	Tag	Axial load over buckling load in bar 1
<b>Buckle2</b>	Tag	Axial load over buckling load in bar 2
<b>Buckle3</b>	Tag	Axial load over buckling load in bar 3

Your *Responses* should look like this:

	Response Name	Source	Formula	Comment
1	Weight	Tag		Total weight
2	Umin	Tag		Min disp in horizontal direction
3	Vmin	Tag		Min disp in vertical direction
4	Umax	Tag		Max disp in horizontal direction
5	Vmax	Tag		Max disp in vertical direction
6	Sig1min	Tag		Min stress in bar 1
7	Sig2min	Tag		Min stress in bar 2
8	Sig3min	Tag		Min stress in bar 3
9	Sig1max	Tag		Max stress in bar 1
10	Sig2max	Tag		Max stress in bar 2
11	Sig3max	Tag		Max stress in bar 3
12	F1	Tag		Axial force of bar 1
13	F2	Tag		Axial force of bar 2
14	F3	Tag		Axial force of bar 3
15	Freq	Tag		Fundamental frequency
16	Buckle1	Tag		Axial load over buckling load in bar 1
17	Buckle2	Tag		Axial load over buckling load in bar 2
18	Buckle3	Tag		Axial load over buckling load in bar 3

You may need to scroll down on your HEEDS screen to see all of the responses.

## Step 4: Tagging the Input and Output Files

Tagging the input and output files is more complex in this example because the variables and responses are scattered among three input files and three output files. We will begin with tagging the variables for our input file, *truss.in*.

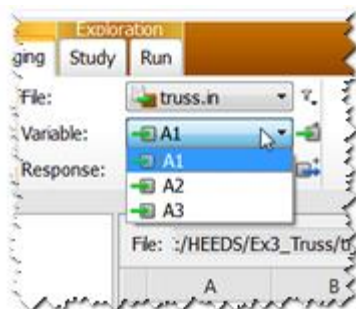
### Tagging the input files

1. Select the **Tagging** tab. At the prompt in the *Process View*, select **truss.in** from the dropdown. The parsed input file is displayed:

File: C:/HEEDS/Ex3\_Truss/truss.in

	A	B	C	D
1	1.0E+7	0.100		
2	8.29052	1.2013	5.74423	
3	10.0			
4	3			
5	40000.00	45.0		
6	30000.00	90.0		
7	20000.00	135.0		
8				
9				

2. On the *Ribbon*, open the drop-down for **Variable**. The variables available for you to tag are displayed. Select **A1**. This is the variable you will tag first:



3. The values for variables **A1**, **A2**, and **A3** are found in row 2 on the table. Tag the values as shown below:

File: C:/HEEDS/Ex3\_Truss/truss.in Delimiters: ☒ , ☐ : ☒

	A	B	C	D
1	1.0E+7	0.100		
2	8.29052	1.2013	5.74423	
3	10.0			
4	3			
5	40000.00	45.0		
6	30000.00	90.0		
7	20000.00	135.0		
8				
9				

Arrows point from the values in row 2 to the variable names: A1 points to 8.29052, A2 points to 1.2013, and A3 points to 5.74423.

Your tagged **truss.in** should look like this:

File: IEEDS/Ex3_Truss/truss.in Delimiters: <input checked="" type="checkbox"/> , <input type="checkbox"/> : <input checked="" type="checkbox"/> ;				
	A	B	C	D
1	1.0E+7	0.100		
2	A1	A2	A3	
3	10.0			
4	3			
5	40000.00	45.0		
6	30000.00	90.0		
7	20000.00	135.0		
8				
9				
10				
11				

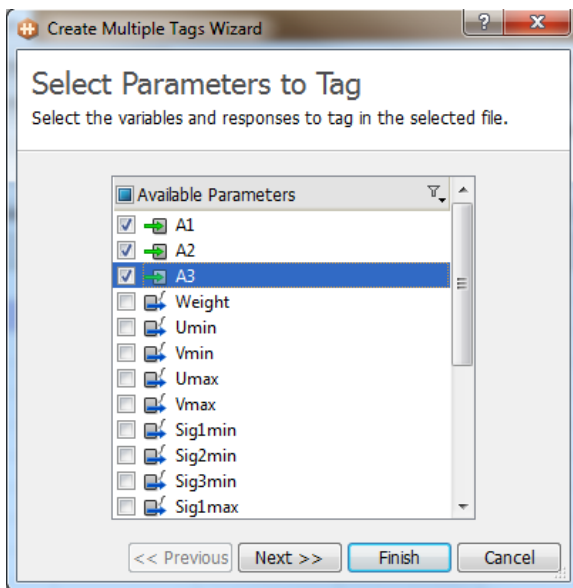
- The values for **A1**, **A2**, and **A3** must also be tagged in the files **freq.in** and **buckle.in**. From the **File** drop-down on the *Ribbon*, select **freq.in**.
- The values to tag for **freq.in** are in row 2 below. This time, let's tag our values using the **Multi-Tag** tool.

File: C:/HEEDS/Ex3_Truss/freq.in Delimiters: <input checked="" type="checkbox"/>				
	A	B	C	
1	1.0E+7	0.0002588		
2	8.29052	1.2013	5.74423	
3	10.0			
4				
5				

- Hold your **Shift** key while selecting all three cells in row 2. Then move to the *Ribbon* and click **Multi-Tag**:

freq.in				
A1				
Tagging				
File: C:/HEEDS/Ex3_Truss/freq.in Delimiters: <input checked="" type="checkbox"/> , <input type="checkbox"/> : <input checked="" type="checkbox"/> ; <input checked="" type="checkbox"/> tab <input checked="" type="checkbox"/> space				
	A	B	C	
1	1.0E+7	0.0002588		
2	8.29052	1.2013	5.74423	
3	10.0			
4				
5				

7. The *Create Multiple Tags Wizard* appears. Select the variables you want to tag, in this case, **A1**, **A2**, and **A3**, and then click **Finish**. HEEDS tags the items in order.



8. Your tagged input file should look like this:

File: C:/HEEDS/Ex3\_Truss/freq.in

	A	B	C
1	1.0E+7	0.0002588	
2	A1	A2	A3
3	10.0		

9. Now, from the **File** drop-down on the *Ribbon*, select **buckle.in**. We have two sets to tag for this file. The first set to tag is **A1**, **A2**, and **A3**. The values for this set are in row 2. Tag these now:

File: C:/HEEDS/Ex3\_Truss/buckle.in Delimiters: ☒ , ☐ : ☒ ; ☒ tab ☒

	A	B	C
1	1.0E+7	10.0	1.0
2	8.29052	1.2013	5.74423
3	3.67222E+04	4.63556E+03	-3.27784E+03
4			

10. The second set to tag is **F1**, **F2**, and **F3**. The values for this set are in row **3** below. Use the **Multi-Tag** tool to tag these variables. You'll have to scroll down on the *Wizard* to locate the variables.

File: C:/HEEDS/Ex3\_Truss/buckle.in Delimiters: ☒ , ☐ :

	A	B	C
1	1.0E+7	10.0	1.0
2	A1	A2	A3
3	3.67222E+04	4.63556E+03	-3.27784E+03
4			

Your completed **buckle.in** should look like this:

File: C:/HEEDS/Ex3\_Truss/buckle.in

	A	B	C
1	1.0E+7	10.0	1.0
2	A1	A2	A3
3	F1	F2	F3

11. Save your project.

### Tagging truss.out

Most of the responses are in **truss.out**. There are a lot of them, so we'll take them in sets:

- On the *Ribbon*, open the drop-down for **File** and select, **truss.out**.
- On the *Ribbon*, open the drop-down for **Response**, and select **Weight**, which you will tag first.
- Scroll through the table of the parsed file and locate the value **0.21049E+02** in column **A**, row **35**. Tag this value for **Weight**:

File: ettingStarted\Truss\truss.out Delimiters: ☒ , ☐ : ☒ ; ☒ tab

	A	B	C
32	Member	Forces	
33	-0.23654E+04	0.33452E+04	0.17635E+05
34	Structure	Mass	
35	0.21049E+02		
36			
37			
38	Overall	Results:	

Your tagged response should like this:

File: ettingStarted\Truss\truss.out Delimiters: ☒ , ☐ : ☒ ; ☒ tab

	A	B	C
32	Member	Forces	
33	-0.23654E+04	0.33452E+04	0.17635E+05
34	Structure	Mass	
35	Weight		
36			
37			
38	Overall	Results:	

4. Scroll, if necessary, to see lines **40** and **42**, where the **Umin**, **Vmin**, **Umax**, and **Vmax** values are located. Tag these as follows:

File: \_GettingStarted\Truss\truss.out Delimiters: ☒

	A	B	C
38	Overall	Results:	
39	Minimum	Displacement	Field
40	-0.33553E-02	0.27847E-02	
41	Maximum	Displacement	Field
42	0.50001E-02	0.50000E-02	
43	Minimum	Stress	Field
44	-0.28531E+03	0.27847E+04	-0.57063E+03
45	Maximum	Stress	Field

You tagged responses should look like this:

File: \_GettingStarted\Truss\truss.out Delimiters: ☒

	A	B	C
38	Overall	Results:	
39	Minimum	Displacement	Field
40	Umin	Vmin	
41	Maximum	Displacement	Field
42	Umax	Vmax	
43	Minimum	Stress	Field
44	-0.28531E+03	0.27847E+04	-0.57063E+03

5. The values for the **Sig1min** and **Sig1max** responses are in rows **44** and **46**. Tag these now:

File: gStarted\Truss\truss.out Delimiters: ☒ , ☐ : ☒ ; ☒ tab ☒ space "="

	A	B	C
39	Minimum	Displacement	Field
40	Umin	Vmin	
41	Maximum	Displacement	Field
42	Umax	Vmax	
43	Minimum	Stress	Field
44	-0.28531E+03	0.27847E+04	-0.57063E+03
45	Maximum	Stress	Field
46	0.44294E+04	0.50000E+04	0.30700E+04
47	Minimum	Member	Forces
48	-0.23654E+04	0.33452E+04	-0.32779E+04

Annotations: Sig1min points to row 44, Sig1max points to row 46, Sig2min points to row 44, Sig2max points to row 46, Sig3min points to row 44, Sig3max points to row 46.

Your tagged responses should look like this:

File: \_GettingStarted\Truss\truss.out Delimiters: ☒ , ☐ :

	A	B	C
39	Minimum	Displacement	Field
40	Umin	Vmin	
41	Maximum	Displacement	Field
42	Umax	Vmax	
43	Minimum	Stress	Field
44	Sig1min	Sig2min	Sig3min
45	Maximum	Stress	Field
46	Sig1max	Sig2max	Sig3max
47	Minimum	Member	Forces
48	-0.23654E+04	0.33452E+04	-0.32779E+04

6. The last responses that need to be tagged in this file are **F1**, **F2**, and **F3**. They are located in row **48** (the last row) of the file:

File: \_GettingStarted\Truss\truss.out Delimiters: ☒ , ☐ :

	A	B	C
45	Maximum	Stress	Field
46	Sig1max	Sig2max	Sig3max
47	Minimum	Member	Forces
48	-0.23654E+04	0.33452E+04	-0.32779E+04
49			

Annotations: F1 points to row 48, F2 points to row 48, F3 points to row 48.

Your completed **truss.out** should look like this:

File: \_GettingStarted\Truss\truss.out

	A	B	C	D
31	-0.28531E+03	0.27847E+04	0.30700E+04	
32	Member	Forces		
33	-0.23654E+04	0.33452E+04	0.17635E+05	
34	Structure	Mass		
35	Weight			
36				
37				
38	Overall	Results:		
39	Minimum	Displacement	Field	
40	Umin	Vmin		
41	Maximum	Displacement	Field	
42	Umax	Vmax		
43	Minimum	Stress	Field	
44	Sig1min	Sig2min	Sig3min	
45	Maximum	Stress	Field	
46	Sig1max	Sig2max	Sig3max	
47	Minimum	Member	Forces	
48	F1	F2	F3	

7. Save your project.

### Tagging freq.out

1. Only one response needs to be tagged in **freq.out**. On the *Ribbon*, open the drop-down for **File** and select, **freq.out**. Then open the drop-down for **Response** and select **Freq**.
2. The **Freq** response is in column **A**, row **4** (the last row) of the file. Tag this value:

File: \_GettingStarted\Truss\freq.out Delimited

	A	B	C
1			
2			
3	Fundamental	Frequency:	
4	0.25000E+04		
5			

An orange arrow points from the value 0.25000E+04 in row 4, column A to the label **freq**.

Your tagged response should look like this:

File: \_GettingStarted\Truss\freq.out Delimiter

	A	B	C
1			
2			
3	Fundamental	Frequency:	
4	Freq		
5			

### Tagging buckle.out

1. On the *Ribbon*, open the drop-down for **File** and select, **buckle.out**.
2. Open the drop-down for **Response** and select **Buckle1**.
3. In **buckle.out**, we need to tag **Buckle1**, **Buckle2**, and **Buckle3** in the last row of the file:

File: \_GettingStarted\Truss\buckle.out Delimiters: ☒ , ☐ : ☒ ; ☒ tab ☒ space "=( )"

	A	B	C
1			
2			
3	Percentage	To	Buckling
4	-0.10827E-02	-0.32546E-02	0.20131E-03
5			

Your tagged **buckle.out** should look like this:

File: \_GettingStarted\Truss\buckle.out Delimiter

	A	B	C	D
1				
2				
3	Percentage	To	Buckling	
4	Buckle1	Buckle2	Buckle3	
5				

4. Save your project.

## Step 5: Defining the Study

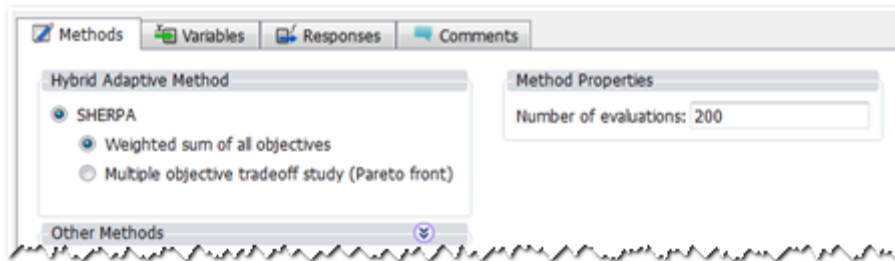
We have completed the *Process Automation* tasks of our Project. Next, we will move to the *Exploration* level tasks: *Study* and *Run*. We will begin with *Study*, and assemble our problem definition.

### Selecting the search method

Our first task is to select the search method for the study.

1. On the *Ribbon*, select the **Study** tab. The *Study Manager* appears.
2. On the *Study Manager*, click the **Methods** tab. The *Methods View* appears.

3. Select the search method that you want to use for your study. For this example, choose the default, **SHERPA – Weighted sum of all objectives**. For **Number of evaluations**, enter **200**. The completed *Method definition* for your study should look like this:



### Completing the variable definition

In this example, we have three variable definitions:

Click the **Variables** tab on the *Study Manager*. The variable definitions for this example were completed on the **Parameters** tab. No further action is required.

### Completing the response definition

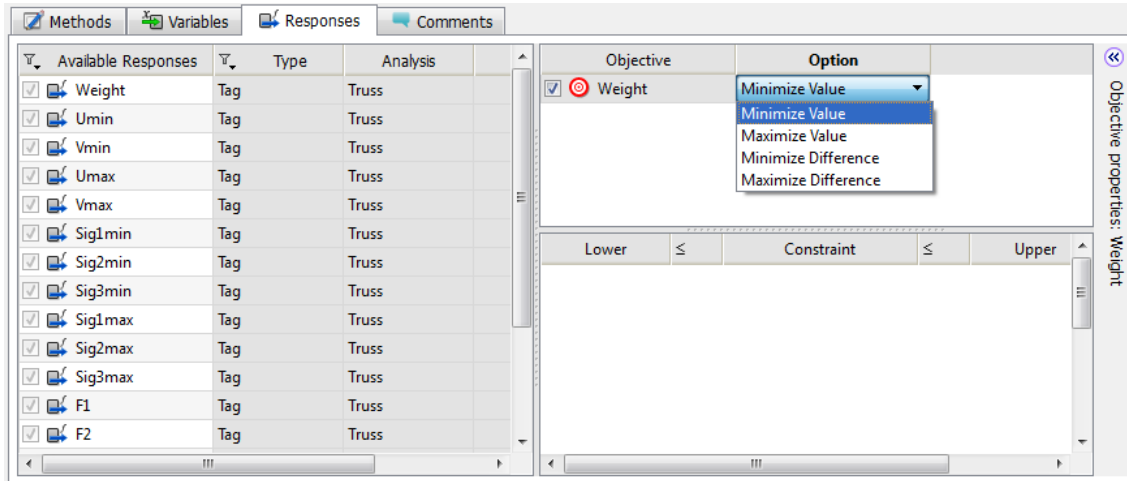
In this procedure, we will define one objective and 14 constraints for our design study.

1. On the *Study Manager*, select the **Responses** tab. All available responses for our analysis are listed here. From these, we will define one objective and fourteen constraints for our study.
2. On the *Ribbon*, click **Add Objective**, and then select **Weight** from the drop-down:

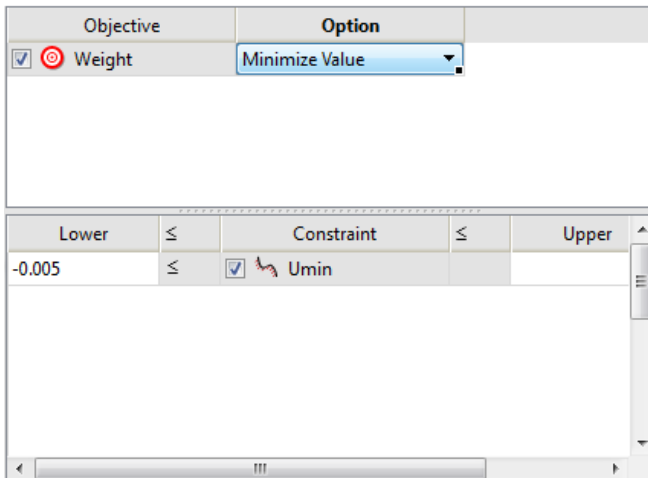


**Weight** appears in the *Study Manager* as an **Objective**.

3. On the *Study Manager*, locate the **Option** field, and then select **Minimize Value**:



4. Our next task is to add a constraint to our study. We can do this using the **Add Constraint** tool on the *Ribbon*, or we can go to the *Study Manager* and simply select the constraint from the *Responses* list, and then drag and drop it into the *Constraint View*. Let's try the drag-and-drop option. Drag **Umin** from the *Responses* list and drop it into the *Constraint View*.
5. On the *Constraint View* of the *Study Manager*, select the cell  $\leq$  (**Lower**) for **Umin**, and enter **-0.005** for the value.
6. Use the defaults for the remaining properties. So far, our *Objective* and *Constraint* definitions should look like this:



7. Using the procedure described in steps 4 through 6, define the remaining constraints according to the table below. Drag multiple constraints at once by holding your **Ctrl** key while selecting the constraints.

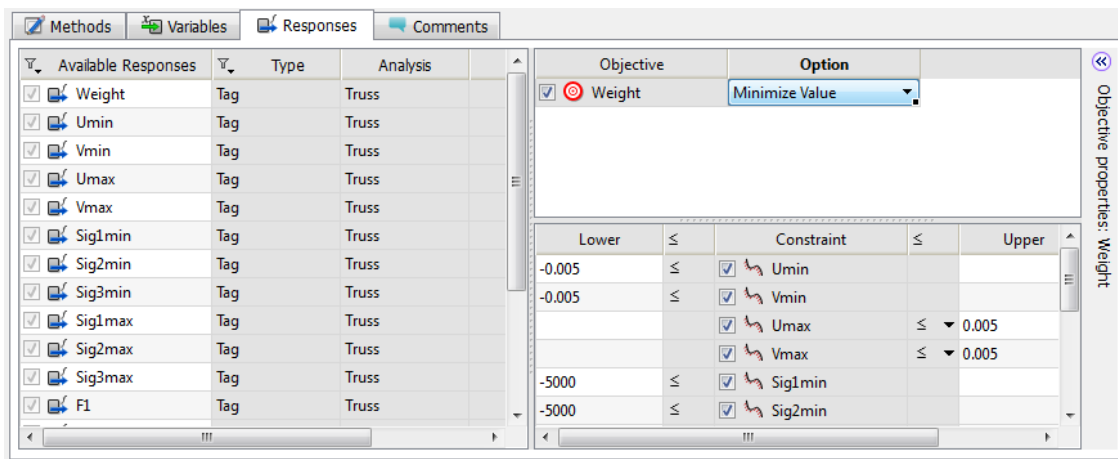
Add Constraint	Option $\leq$ (Upper)	Option $\leq$ (Lower)
<b>Vmin</b>		-0.005
<b>Umax</b>	0.005	
<b>Vmax</b>	0.005	
<b>Sig1min</b>		-5000
<b>Sig2min</b>		-5000
<b>Sig3min</b>		-5000

Add Constraint	Option $\leq$ (Upper)	Option $\leq$ (Lower)
<b>Sig1max</b>	20000	
<b>Sig2max</b>	20000	
<b>Sig3max</b>	20000	
<b>Freq</b>		2500
<b>Buckle1</b>	1	
<b>Buckle2</b>	1	
<b>Buckle3</b>	1	



**Note:** The order of the constraints could be different depending on which method (**drag-and-drop** or **Add Constraint** tool) you use to add the constraints.

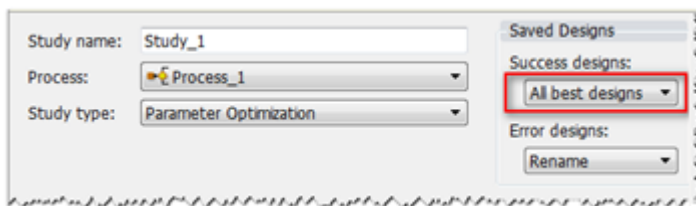
Your finished *Constraints* definition should look like this:



## Specifying which designs to save

In our previous examples, HEEDS MDO saved only the files for the *best design*, which is the default setting. For this run, let's change the option so that every time HEEDS finds a design that is better than any evaluated so far, HEEDS saves that design's files:

1. In the *Study Manager*, locate **Saved Designs**, and from the **Success designs** pulldown, select: **All best designs**. HEEDS will save the files for all best designs.



2. Use the default values for the remaining options. Your completed *Study* definition should look like this:

✓ The study has passed all validity checks and is ready to [run](#).

Study name:

Process:

Study type:

Saved Designs: Success designs:  Error designs:

Run Options: ☒ Save restart data after each evaluation ☐ Do not stop HEEDS for a design-based error

Methods Variables Responses Relationships Comments

Available Responses	Type	Analysis	Objective	Option
<input checked="" type="checkbox"/> Weight	Tag	Truss	<input checked="" type="checkbox"/> Weight	Minimize Value
<input checked="" type="checkbox"/> Umin	Tag	Truss		
<input checked="" type="checkbox"/> Vmin	Tag	Truss		
<input checked="" type="checkbox"/> Umax	Tag	Truss		
<input checked="" type="checkbox"/> Vmax	Tag	Truss		
<input checked="" type="checkbox"/> Sig1min	Tag	Truss		
<input checked="" type="checkbox"/> Sig2min	Tag	Truss		
<input checked="" type="checkbox"/> Sig3min	Tag	Truss		
<input checked="" type="checkbox"/> Sig1max	Tag	Truss		
<input checked="" type="checkbox"/> Sig2max	Tag	Truss		
<input checked="" type="checkbox"/> Sig3max	Tag	Truss		
<input checked="" type="checkbox"/> F1	Tag	Truss		
<input checked="" type="checkbox"/> F2	Tag	Truss		
<input checked="" type="checkbox"/> F3	Tag	Truss		
<input checked="" type="checkbox"/> Freq	Tag	Freq		
<input checked="" type="checkbox"/> Buckle1	Tag	Buckle		
<input checked="" type="checkbox"/> Buckle2	Tag	Buckle		
<input checked="" type="checkbox"/> Buckle3	Tag	Buckle		

Lower	≤	Constraint	≤	Upper
-0.005	≤	<input checked="" type="checkbox"/> Umin		
-0.005	≤	<input checked="" type="checkbox"/> Vmin		
		<input checked="" type="checkbox"/> Umax	≤	0.005
		<input checked="" type="checkbox"/> Vmax	≤	0.005
-5000	≤	<input checked="" type="checkbox"/> Sig1min		
-5000	≤	<input checked="" type="checkbox"/> Sig2min		
-5000	≤	<input checked="" type="checkbox"/> Sig3min		
		<input checked="" type="checkbox"/> Sig1max	≤	20000
		<input checked="" type="checkbox"/> Sig2max	≤	20000
		<input checked="" type="checkbox"/> Sig3max	≤	20000
2500	≤	<input checked="" type="checkbox"/> Freq		
		<input checked="" type="checkbox"/> Buckle1	≤	1
		<input checked="" type="checkbox"/> Buckle2	≤	1
		<input checked="" type="checkbox"/> Buckle3	≤	1

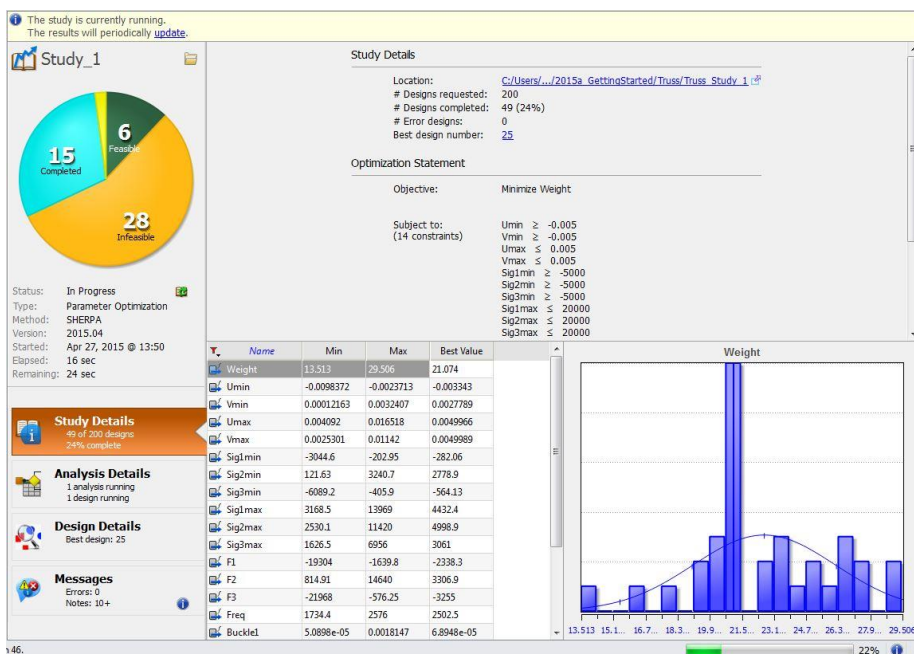
3. Save your project.

## Step 6: Running the Study

You are ready to execute your study. Select **Run** from the tab bar. The *Run* page appears with a summary of your *Study Details* and *Optimization Statement*.

### To run the study

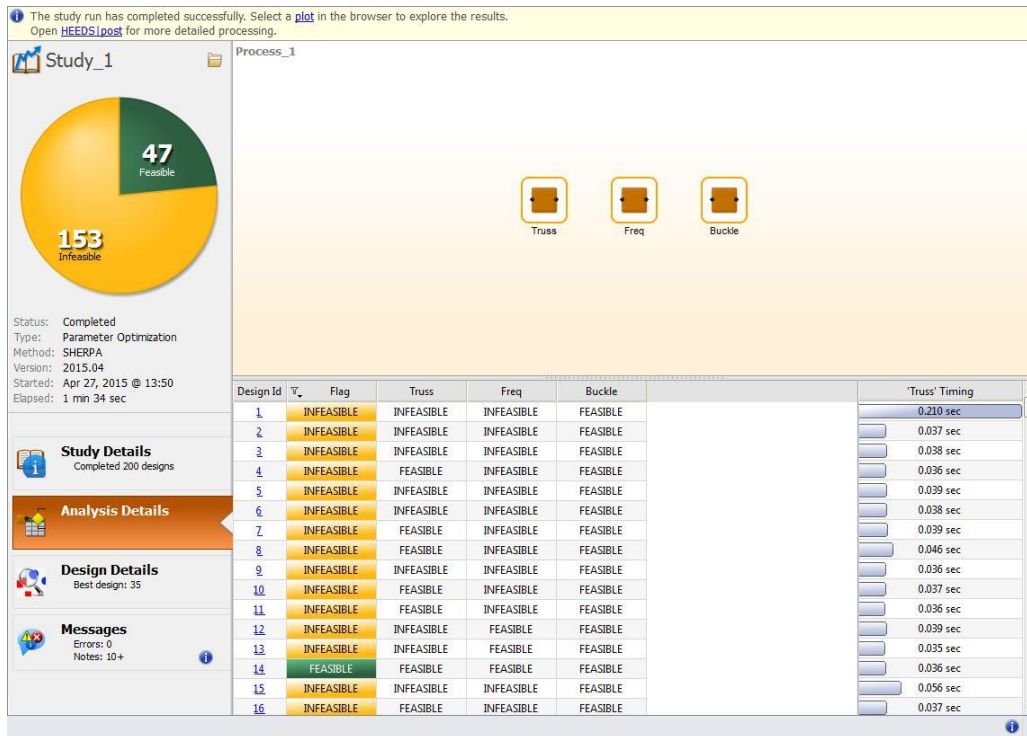
On the *Ribbon*, click **Run** to execute the study. The *Run* page will show you the progress of your running study with summary information.



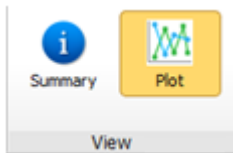
## Monitor your study

HEEDS provides you with a **Summary** and **Plot** tool on the *Ribbon* to monitor design evaluations in progress. This study will run fairly quickly, but we'll have time to review the *Log* file and *Objective History* plot.

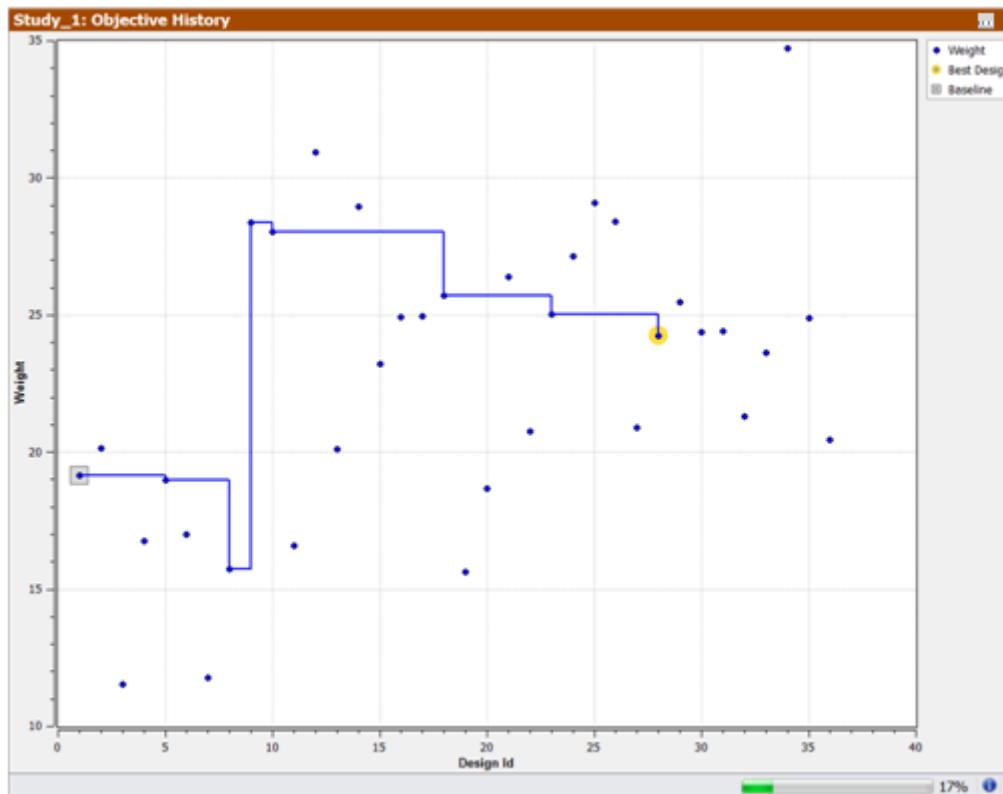
1. As the study runs, watch the pie chart as it updates.
2. Click on the **Analysis Details** tab. The information on this page will update and show **Completed** when HEEDS is finished with a design. When MDO imports the results the **Completed** are distributed to **Feasible** or **Infeasible**. The **Timing** column shows the timing of the selected analysis and can be used to identify a bottleneck in the analysis.



3. Next let's view the *Object History* plot for your study. From the *Ribbon*, click **Plot**.

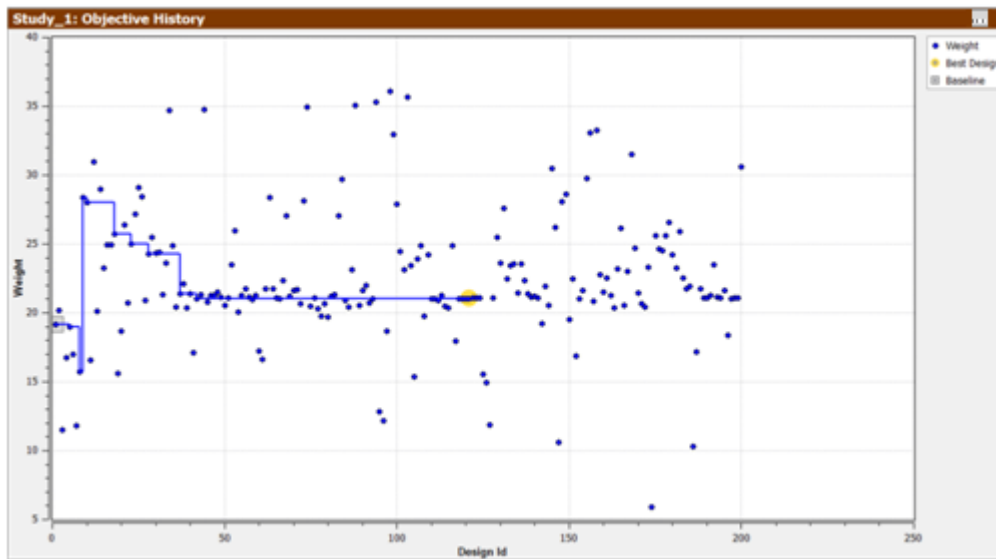


The **Plot** tool displays the **Objective History** plot as the default:



**Note:** Your plots may not look exactly the same as the ones we show here, but the results will be similar.

4. On the *Ribbon*, click **Reload Plot Data**. The *Objective History* plot updates with the most current data available for the run:



5. HEEDS generates additional default post-processing plots for you to explore: **Parallel Data**, **Design Table**, **Constraint Violations**, or **Performance History**. To view any of these, click **Plot** on the *Ribbon*, then move to the **Plots** node on the tree and make a selection.



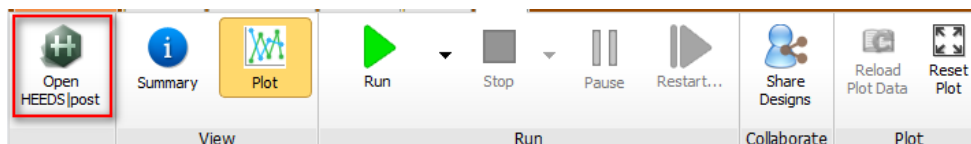
**Note:** To see an updated plot for any plot type, click **Reload Plot Data** on the *Ribbon*. **Reload Plot Data** is available only during a run, not after the study has completed.

## Step 7: Viewing Results Using HEEDS POST

HEEDS POST gives you access to all of the plots that are available in HEEDS MDO. In addition, HEEDS POST provides you with advanced post-processing capabilities to review results. In the following section, we will use HEEDS POST to explore some plot types available for this run: **Parallel**, **Design Table**, and **Constraint Violation**.

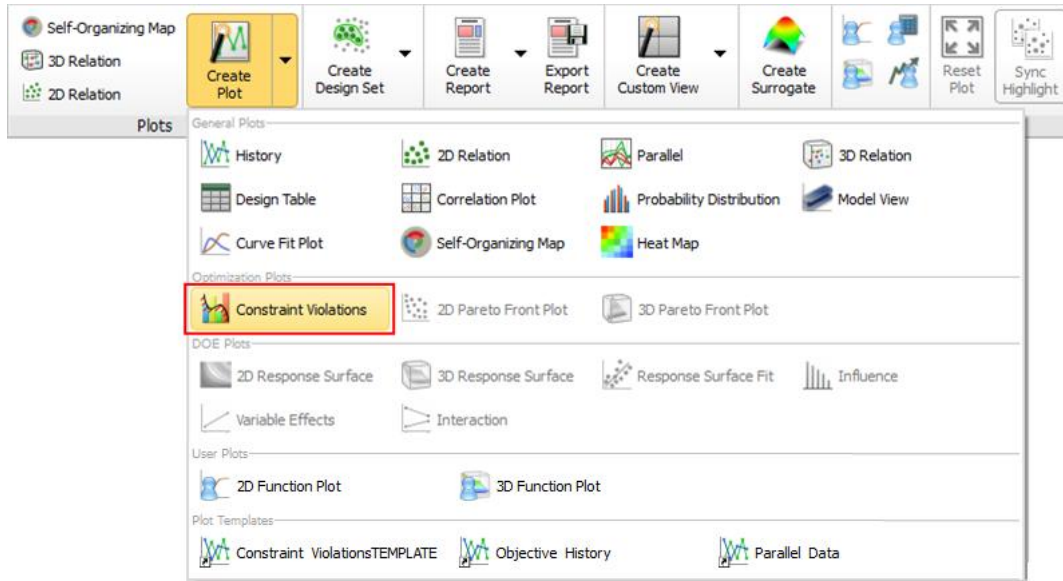
### Generate plots using HEEDS POST

On the HEEDS MDO *Ribbon*, click **Open HEEDS|post**:



## Constraint violation plot

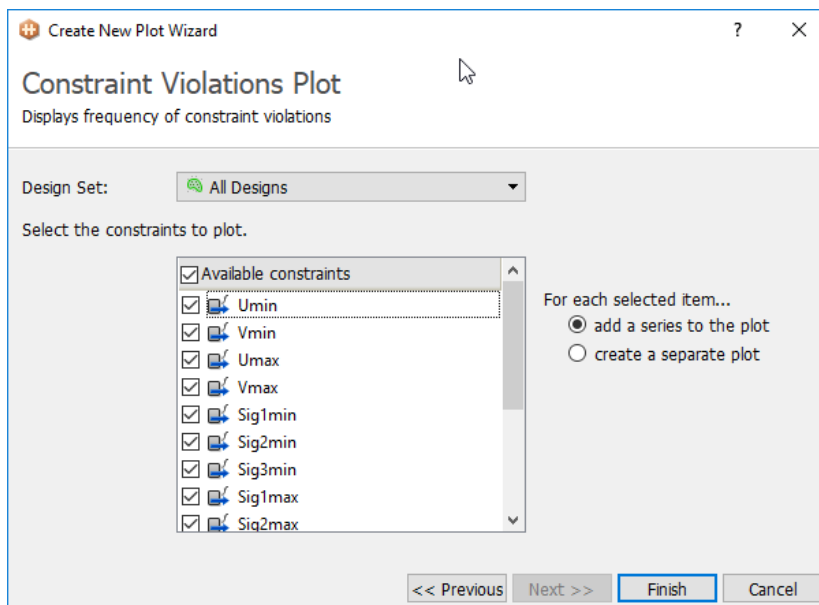
1. On the *Ribbon*, locate the *Optimization Plots* tools, then click **Constraint Violations**:



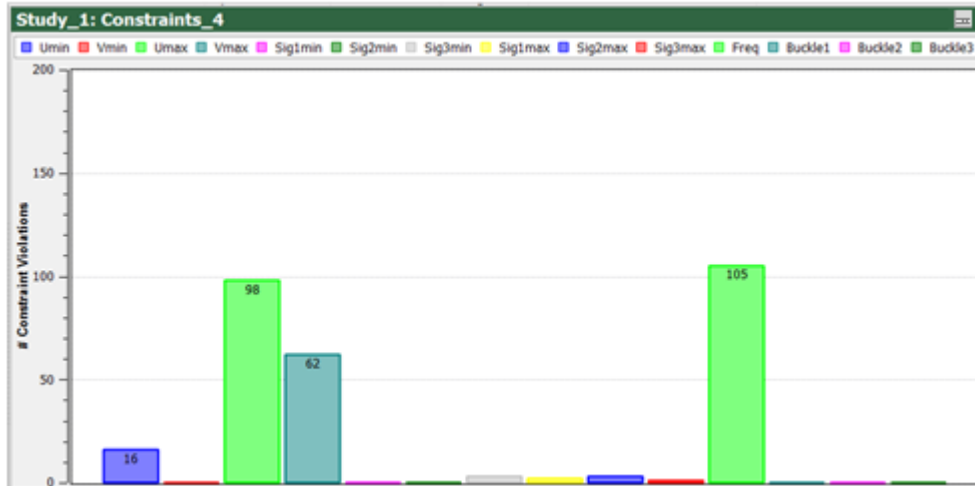
The *Create New Plot Wizard* appears.

2. Enter the following in the *Wizard*:
  - In the **Design Set** drop-down, select **All Designs**.
  - In the **Select the Constraints to plot** list, select **Available constraints**. This option selects all available constraints.

Your *Wizard* should look like this:



- Click **Finish** to generate the plot:

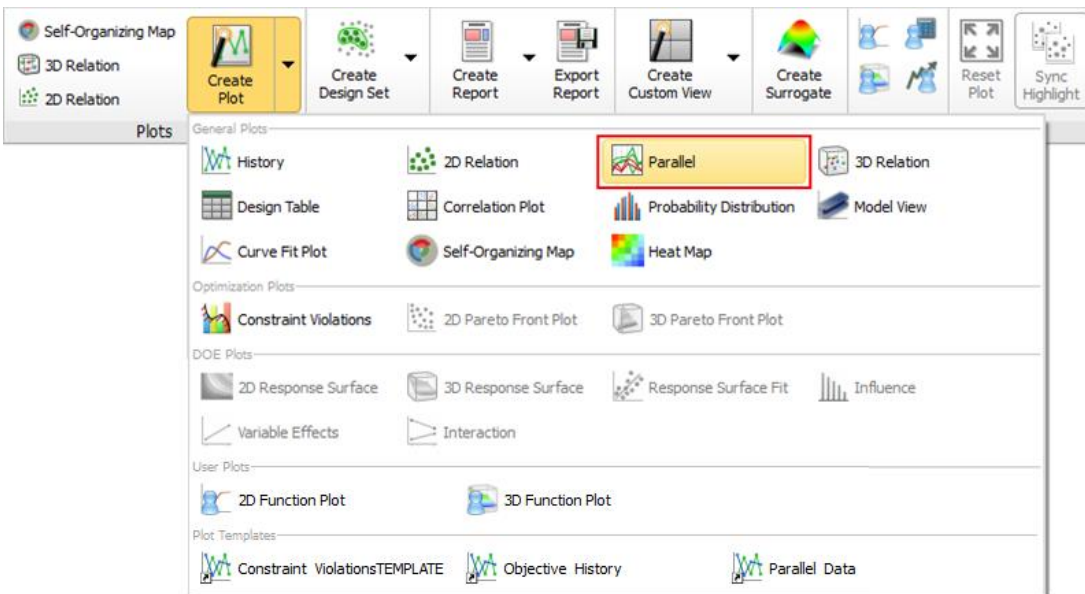


Since all of the designs are plotted, we can easily see that **Umax** and **Freq** had the highest number of constraint violations.

## Parallel Plot

Next we'll create a *Parallel* plot.

- On the *Ribbon*, click **Create Plot**, and then select **Parallel**:

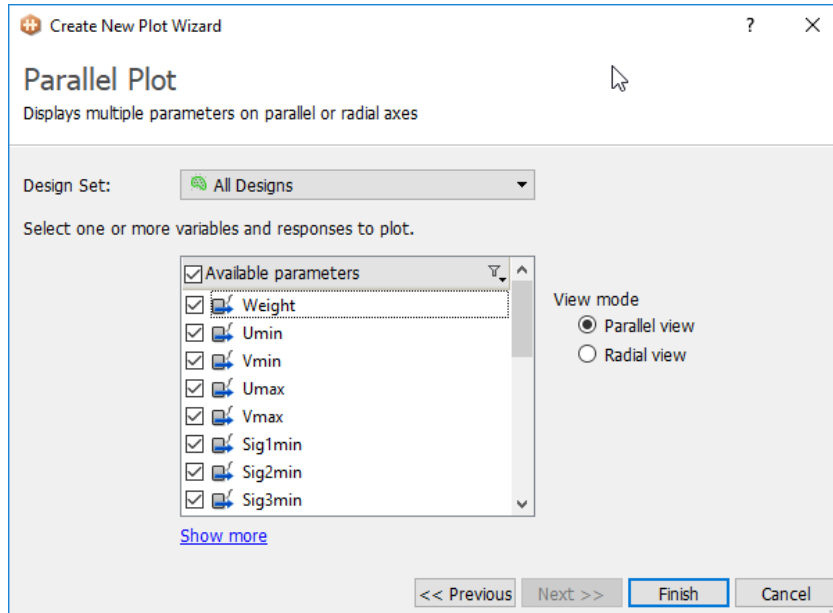


The *Create New Plot Wizard* appears.

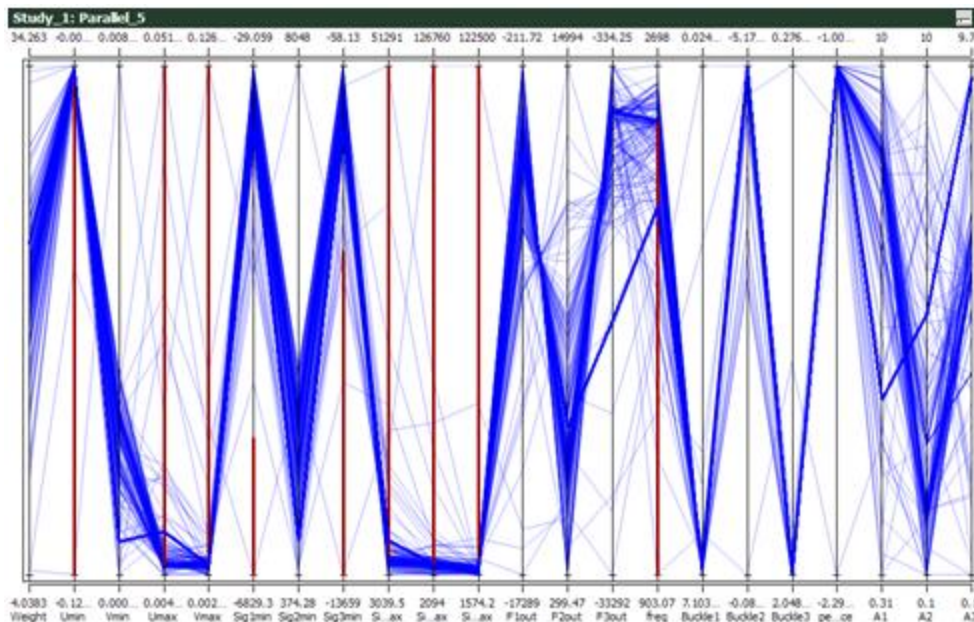
2. Enter the following in the *Wizard*:

- In the **Design Set** drop-down, select **All Designs**.
- Click **Available parameters**. All parameters are selected using this option.

Your *Wizard* should look like this:



3. Click **Finish** to generate the plot:

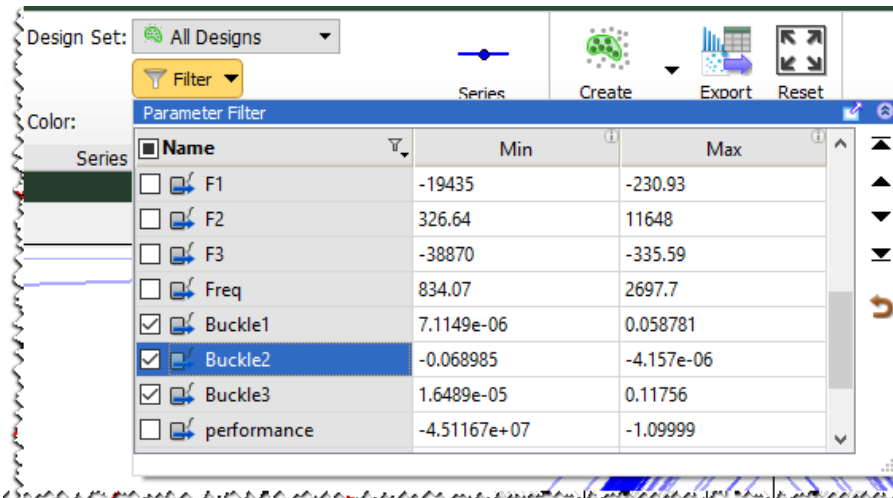


The blue lines represent *All Designs*, the red lines represent the *Best Design*, and the gray lines represent the *Baseline Design*.

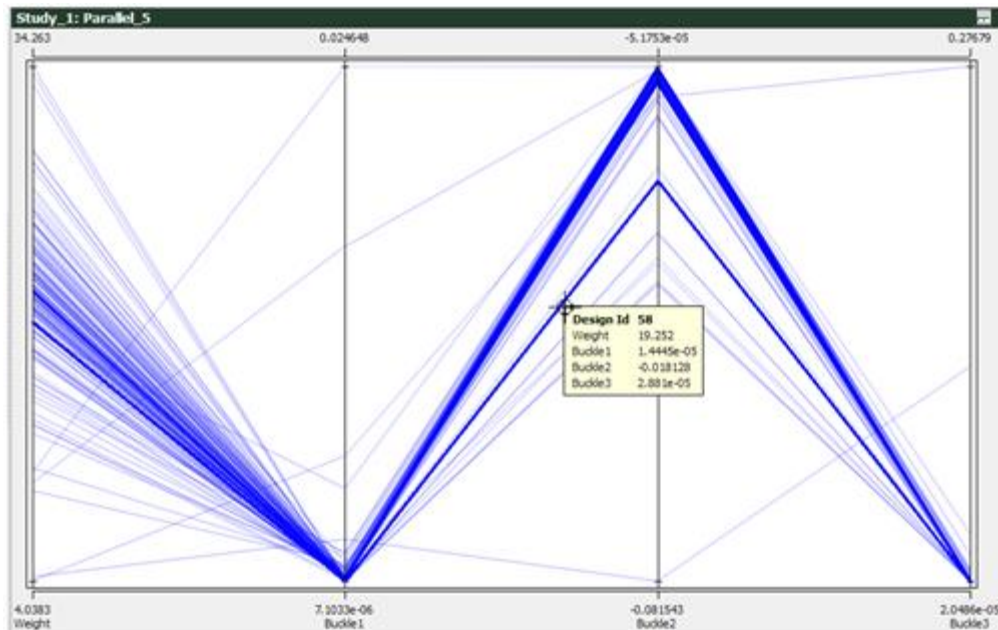
- To see particular designs, select them using the **Filter** button in the **Series Data** group. To see designs that fall into a particular range, adjust the values in the **Min** and **Max** fields.

You can click and drag the arrows at the top and bottom of each design to change the minimum and maximum values. You can also zoom in and out on the axes and on individual designs by rotating the mouse wheel.

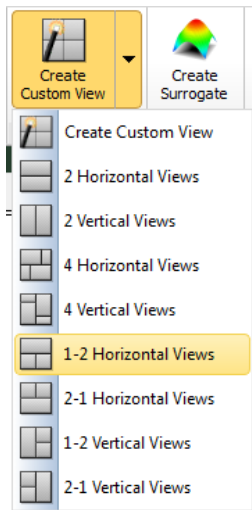
- Let's look at some specific designs. Select **Filter**, and then select **Weight** from the **Name** list. Scroll down and select **Buckle1**, **Buckle2**, and **Buckle3**. Your filter should look like this:



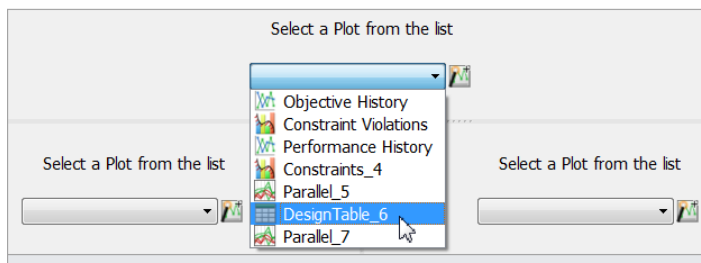
- To see information about a particular design, select a line on the plot. The line darkens and a window with the corresponding design information appears. The line selected on the plot below shows details for *Design 58*:



7. Create a plot view that shows the *Parallel* data, *Constraint Violation* data, and the *Design Table*. To do this select the appropriate plot view from the *Plot Views* area on the *Ribbon* or create a *Custom View*.

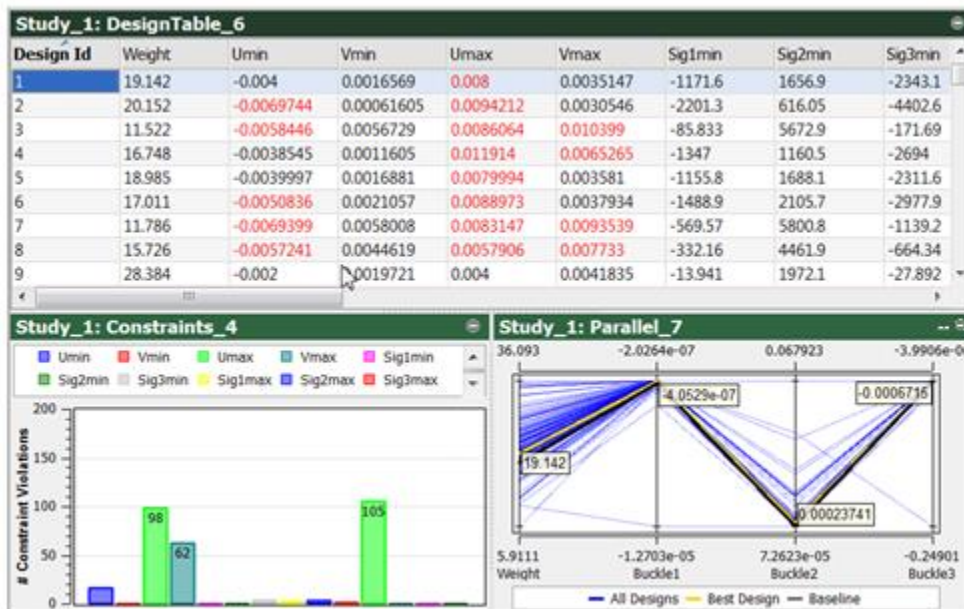


8. Select the appropriate plots such that the *Design Table* is on top and the *Constraint Violation* and *Parallel Plot* are on the bottom. Begin with the top dropdown and select **Design Table**. Next select **Constraint Violations** from the left drop-down and **Parallel** from the right.



**Note:** If there is no design table plot in the project, use steps 1-3 in this example to create a design table instead of a parallel plot.

9. Three plots should appear on your screen. Pick the first design in the *Design Table*. Notice that because **Sync Highlight** is activated on the *Ribbon*, the design you picked is also selected in the other two plots:



## Step 8: Animating the plot

Plot animations can be useful to visualize constructive aspects of a study. Most plots animate the appearance of designs as HEEDS evaluated them. When the **Animation** tab is selected, the plot only displays data defined by the animation settings.

1. Select the *Objective History* plot from the tree view.
2. Select the **Animation** tab.
3. Click the **Play** button.
4. Animate the *Parallel Plot*, adjusting the animation speed as necessary.

The points appear on the plot as the animation progresses. Animating the parallel plot can help to identify trends when the density of lines in an area increases.

## Example 4: DOE Study of a Coil Spring

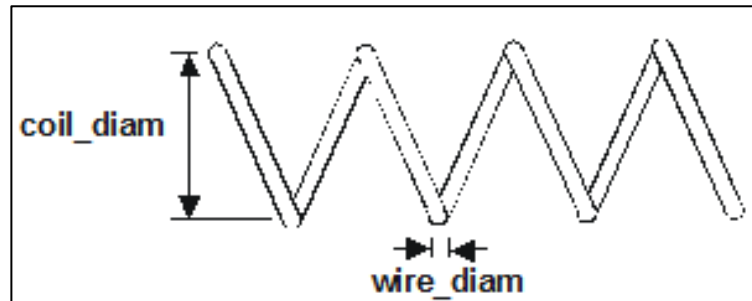


Fig.1. Coil spring

This example shows you how to set up a simple DOE study.

### The problem

A coil spring (fig. 1) is fixed on one end and pulled at the other with a force of 10 lbs. We want to determine the effects of three variables on four selected responses:

<b>Variables:</b>	<i>coil_diam</i>	Mean diameter of the coil
	<i>wire_diam</i>	Diameter of the wire
	<i>num_coils</i>	Number of active coils
<b>Responses:</b>	<i>deflection</i>	Total deflection of coil
	<i>shear_stress</i>	Maximum shear stress
	<i>frequency</i>	Natural frequency of spring
	<i>mass</i>	Total mass of coil, including two inactive coils

### The baseline design

For this example, the executable *spring.exe* was created to solve for *deflection*, *shear\_stress*, *frequency*, and *mass*, using the following equations:

$$\text{deflection} = 80 * \text{coil\_diam}^3 * \text{num\_coils} / (\text{wire\_diam}^4 * G)$$

$$\text{shear\_stress} = 8 * k * 10 * \text{coil\_diam} / (\pi * \text{wire\_diam}^3)$$

$$\text{where } k = (4 * \text{coil\_diam} - \text{wire\_diam}) / (4 * (\text{coil\_diam} - \text{wire\_diam})) + 0.615 * \text{wire\_diam} / \text{coil\_diam}$$

$$\text{frequency} = 88248 * \text{wire\_diam} / (2 * \pi * \text{num\_coils} * \text{coil\_diam}^2)$$

$$\text{mass} = \frac{1}{4} (2 + \text{num\_coils}) * \pi^2 * \text{coil\_diam} * \text{wire\_diam}^2 * 7.38e-4$$

The files for our example are found in the *Spring* folder in the *Examples* directory:

- The default path for Windows is: **C:\HEEDS\MDO\Version\Examples\Spring**
- The default path for Linux/UNIX is: **/opt/HEEDS/MDO/Version/Examples/Spring**

If you installed HEEDS MDO in a different directory, look for the files in that directory instead. This example includes three files:

File Name	Description
<b>spring.in</b>	The input file.
<b>spring.out</b>	The output file.
<b>spring.exe</b> (Windows) <b>spring</b> (Linux)	The file that contains the analysis code.

## Step 1: Starting a New Project

Before you create your project, create a directory on your system specifically for your HEEDS project. Be sure to choose a location where you have write permissions.

Start a new project, and save it in your project directory as **SpringEx**. Your project is saved as **SpringEx.heeds**.

## Step 2: Defining the Process

Our first task at the level of *Process Automation* is to identify the analyses of our *Process*. For this problem we will identify one analysis for our *Process*.

### Identifying the analyses

1. Select the **Process** tab. The *Process* page appears.
2. Click the **Analysis\_1** tool from the *Process View* to display the *Analysis Manager*.



The *Analysis Manager* opens with the **Execution** tab selected:

Analysis name: Analysis\_1 ☒ Enabled

Portal: General (no portals)

Execution | Files | Dependencies | Visualization | Comments

Compute resource: Local

Execution command:

Command options:

Num. designs to execute simultaneously: 1

**Analysis Execution Options**

Run in: Analysis folder

Run condition: Always Run

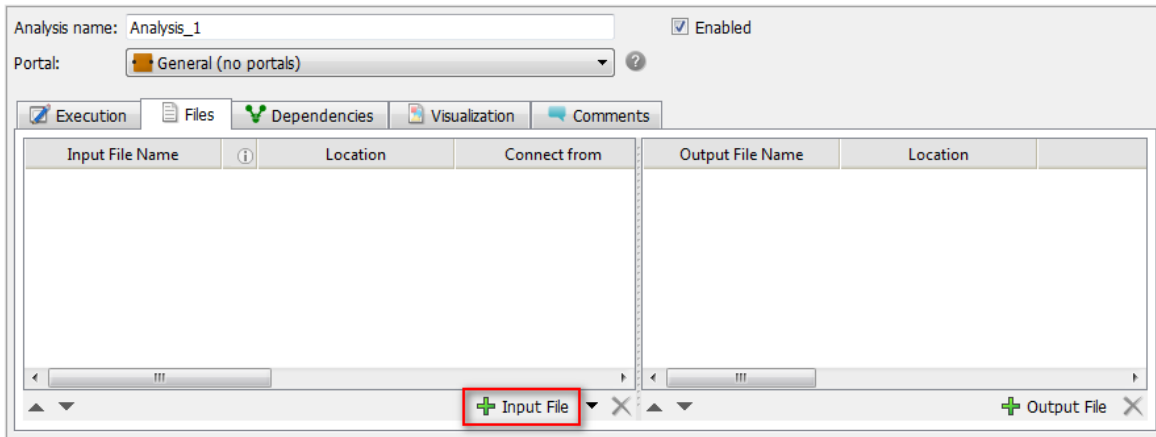
Finished condition: None

Success condition: None

Advanced: Default

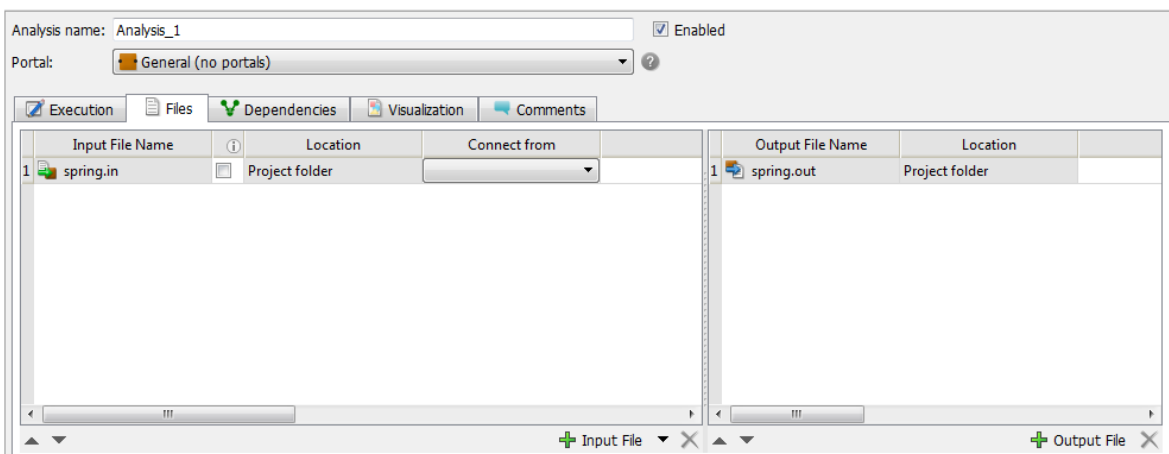
3. Locate the **Execution command** field and click to browse for a file.
4. Navigate to the **Spring** example directory and open **Spring.exe**. **Spring.exe** is added to the analysis definition.
5. Locate the **Command options** field, and enter **spring.in spring.out**.

6. Next, we'll add the input file and output file. Click the **Files** tab and then click **+ Input File**.



7. Navigate to the **Spring** folder, select **spring.in**, and click **Open**. Your *Analysis* definition includes **spring.in** as an input file.
8. Now click **+ Output File**, and navigate to the **Spring** folder, select **spring.out**, and click **Open**. Your *Analysis definition* includes **spring.out** as an output file.

Your *Analysis* definition should look like this:

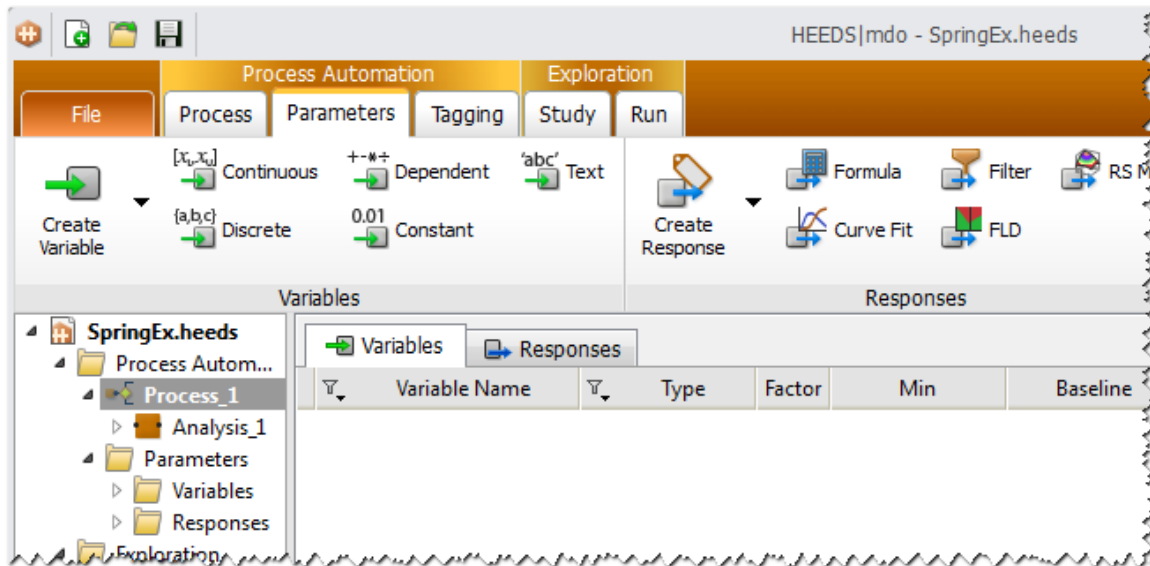


### Step 3: Defining the Parameters

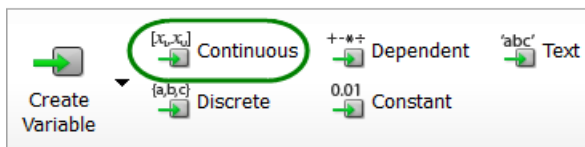
Just as in our parameter optimization projects, we must define the project variables and responses. In a DOE study, only continuous and discrete variables can be used as factors (parameters which will be modified during the DOE study execution), although you can have other types of variables in the project. In our project, all three variables are continuous.

## Defining the project variables

1. Click the **Parameters** tab and then click the **Variables** tab. The *Parameters* page opens to the *Variables Manager*. You can also define the parameters on the **Study** tab.



2. On the *Ribbon*, use the **Continuous** variable tool to add **three** continuous variables to your project.



3. Define the variables as follows:

Name	Type	Min	Baseline	Max	Comment
coil_diam	Continuous	0.25	1	1.3	Mean coil diameter
wire_diam	Continuous	0.05	0.1	0.2	Wire diameter
num_coils	Continuous	2	5	15	Number of active coils

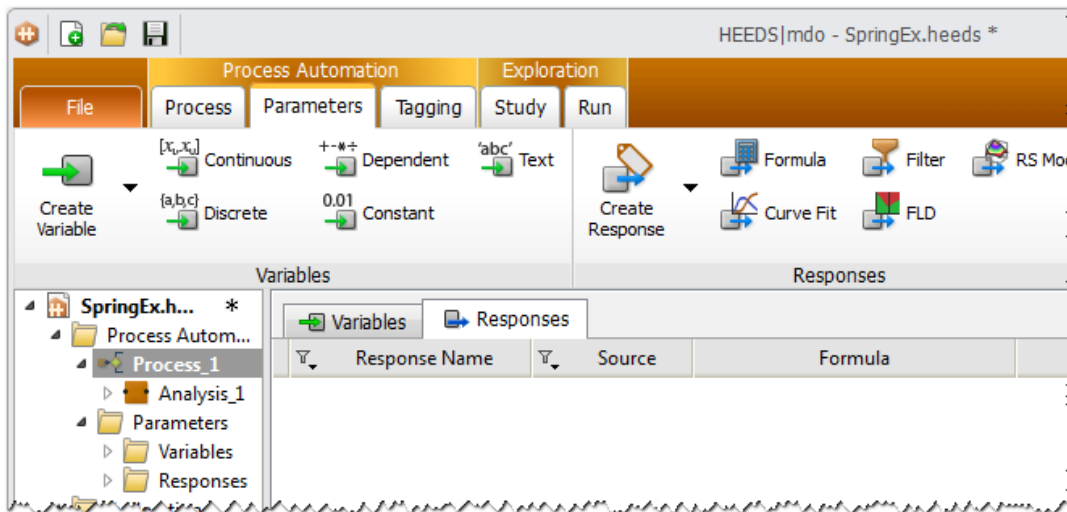
Your variables should look like this:

	Variable Name	Type	Factor	Min	Baseline	Max	Comment
1	coil_diam	Continuous	A	0.25	1	1.3	
2	wire_diam	Continuous	B	0.05	0.1	0.2	
3	num_coils	Continuous	C	2	5	15	

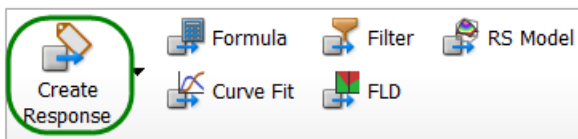
4. Save your project.

## Identifying the project responses

1. From the *Parameters* page, click the **Responses** tab to open the *Responses Manager*:



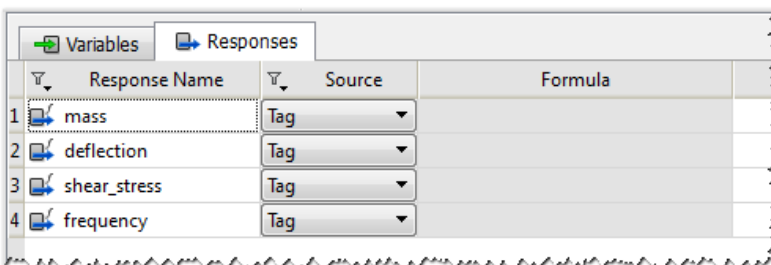
2. On the *Ribbon*, use the **Create Response** tool to add **four** responses to your project:



3. Define the responses as follows:

Name	Source	Comment
mass	Tag	Total mass of coil
deflection	Tag	Total deflection of coil
shear_stress	Tag	Maximum shear stress
frequency	Tag	Natural frequency of spring

Your responses should look like this:



4. Save your project.

## Step 4: Tagging the Input and Output Files


Now that we have defined our variables and responses, we can tag the input and output files.

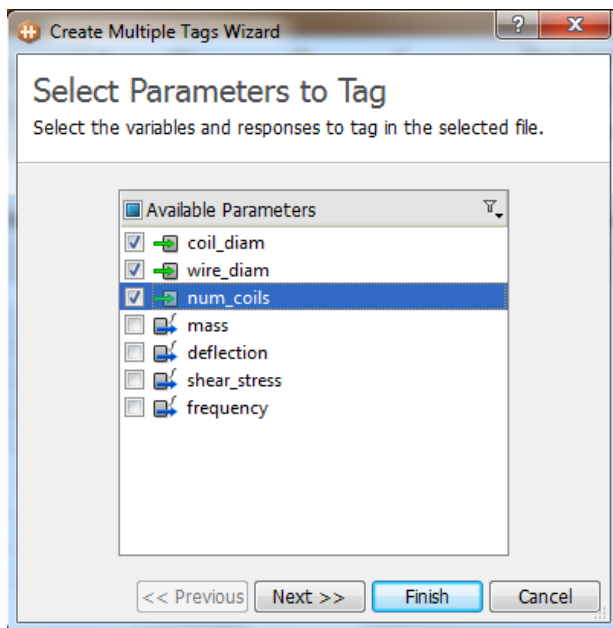
### Tagging the input file

1. Select the **Tagging** tab. Select **spring.in** from the dropdown. The parsed input file is displayed. We'll tag it as follows:

File: C:/HEEDS/Ex4\_Spring/spring.in

	A	B	C	D
1	Coil	Spring	Example	
2				
3	Coil	Diameter:		
4	1.3	coil_diam		
5	Wire	Diameter:		
6	0.20	wire_diam		
7	Number	of	Active	Coils:
8	10	num_coils		
9				

2. On the *Ribbon*, open the dropdown for **Variable** and select **coil\_diam**.
3. Hold your **Ctrl** key, and select these values in the table: **A4**, **A6**, and **A8**.
4. Move to the *Ribbon* and click  **Multi-Tag**.
5. In the *Create Multiple Tags Wizard*, select: **coil\_diam**, **wire\_diam**, and **num\_coils**, and then click **Finish**:



Your tagged **spring.in** should look like this:

File: C:/HEEDS/Ex4\_Spring/spring.in

	A	B	C	D
1	Coil	Spring	Example	
2				
3	Coil	Diameter:		
4	coil_diam			
5	Wire	Diameter:		
6	wire_diam			
7	Number	of	Active	Coils:
8	num_coils			
9				

### Tagging the output file

1. On the *Ribbon*, open the dropdown for **File** and select, **spring.out**. The parsed output file is displayed. We'll tag it as follows:

File: \_GettingStarted\Spring\spring.out Delimiters: ☒ , ☐

	A	B	C	D	E
1	Results	of	Coil	Spring	Analysis
2					
3	Mass	of	Spring:		
4	0.00113673	mass			
5	Deflection:				
6	0.0955217	deflection			
7	Maximum	Shear	Stress:		
8	5093.83	shear_stress			
9	Frequency	of	Surge	Waves:	
10	166.219	frequency			

2. On the *Ribbon*, open the dropdown for **Response** and select **mass**.
3. On the table, while holding your **Ctrl** key, select values: **A4**, **A6**, **A8**, and **A10**.
4. Move to the *Ribbon* and click **Multi-Tag**.
5. On the *Create Multiple Tags Wizard*, select **mass**, **deflection**, **shear\_stress**, and **frequency**, and then click **Finish**.

Your tagged **spring.out** should look like this:

File: \_GettingStarted\Spring\spring.out

	A	B	C	D	E
1	Results	of	Coil	Spring	Analysis
2					
3	Mass	of	Spring:		
4	mass				
5	Deflection:				
6	deflection				
7	Maximum	Shear	Stress:		
8	shear_stress				
9	Frequency	of	Surge	Waves:	
10	frequency				
11					

6. Save your project.

## Step 5: Defining the Study

Up to this point, our procedure for building a DOE project has been the same as the procedure for building a parameter optimization project. From this point on, our procedures will differ.

### Selecting the search method

Our first task is to select the search method for the study.

1. Select the **Study** tab. The *Study Manager* appears.
2. Select the search method that you want to use for your study. For this example, locate **Study type**, open the list box, and select **DOE – Screening/Response Surface**.
3. Next, click the **Methods** tab. The *Methods View* appears with a list of DOE factors.
4. For this study, we want to use all of the listed variables as factors. To add all factors to the study definition, click the check box to the right of **Variable**.  
HEEDS automatically assigns names to the factors, in this case, **A**, **B**, and **C** as you see in the **Factor** column.

Methods Variables Responses Comments						
Select DOE factors and specify evaluation range values.						
Variable	<input checked="" type="checkbox"/>	Fact ⓘ	Baseline	Mode	Low	High
<input checked="" type="checkbox"/> coil_diam	<input checked="" type="checkbox"/>	A	1	Value ▾	0.25	1.3
<input checked="" type="checkbox"/> wire_diam	<input checked="" type="checkbox"/>	B	0.1	Value ▾	0.05	0.2
<input checked="" type="checkbox"/> num_coils	<input checked="" type="checkbox"/>	C	5	Value ▾	2	15

5. Use the defaults for the remaining values on the table.
6. To the right of the *Methods* table is the *Sampling Method* panel. For this project, use the default values for the *Sampling Method* fields.

Your completed study definition should look like this:

✓ The study has passed all validity checks and is ready to [run](#).

Study name:

Process:

Study type:

Surrogates:

Saved Designs

Success designs:

Error designs:

Run Options

☒ Save restart data after each evaluation

☐ Do not stop HEEDS for a design-based error

Random seed:

Methods Variables Responses Comments

Select DOE factors and specify evaluation range values.

Variable	Fact	Baseline	Mode	Low	High
<input checked="" type="checkbox"/> coil_diam	<input checked="" type="checkbox"/> A	1	Value	0.25	1.3
<input checked="" type="checkbox"/> wire_diam	<input checked="" type="checkbox"/> B	0.1	Value	0.05	0.2
<input checked="" type="checkbox"/> num_coils	<input checked="" type="checkbox"/> C	5	Value	2	15

Sampling Method

Number of factors: 3

Number of evaluations: 8

2 Level

☒ Full factorial 8

☐ Partial factorial 4

☐ Taguchi 4

☐ Plackett-Burman N/A

3 Level

☐ Full factorial 27

☐ Partial factorial 9

☐ Taguchi 9,27

Other

☐ Latin hypercube sampling any

☐ Latin hypercube sampling (optimal) any

☐ Time limit:  minutes

☐ Hammersley any

☐ D-Optimal 4-8

☐ Central composite 15

☐ Use existing designs (RSM only) user

7. Save your project.

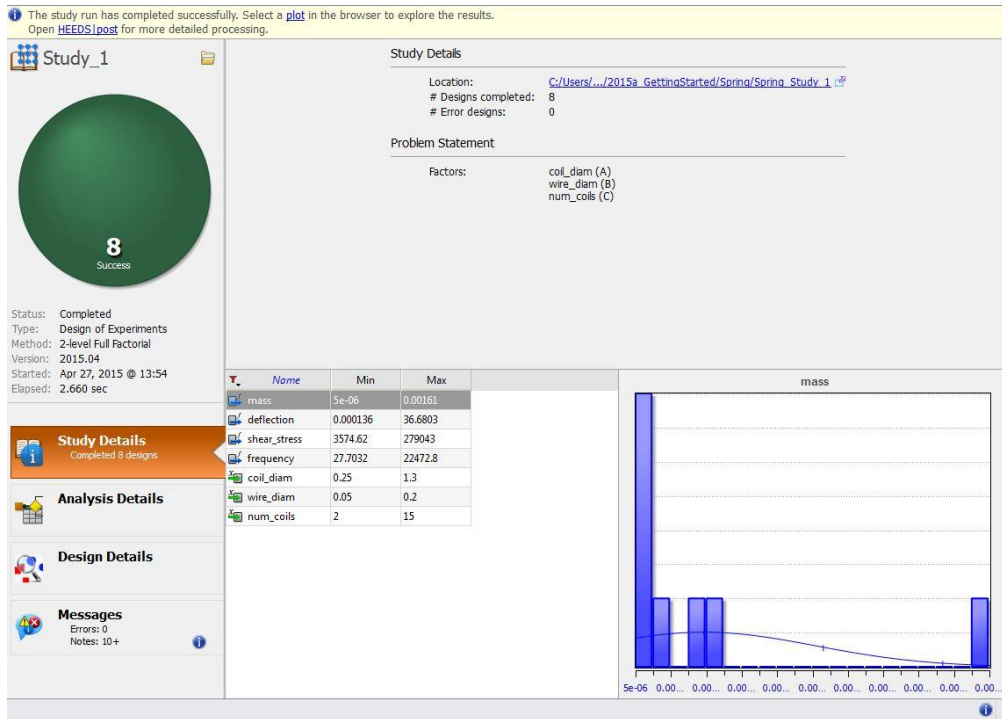
## Step 6: Running the Study

Running and monitoring a DOE study is like running and monitoring a parameter optimization study with one exception. No performance plot is available, as performance is not being evaluated. Performance (i.e., how well the design meets the objective while satisfying the specified constraints) only applies to parameter optimization studies. The variable and response plots are available and can be used to see how the run is progressing.

### To run the study

1. Select **Run** from the tab bar. The *Run* page appears with a summary of your *Study Details* and *Optimization Statement*.

- On the **Ribbon**, click **Run**. Your study summary appears along with the progress of your running study. This study runs very quickly.



- Before we look at the post-processing plots, let's look briefly at the variable plots on the **Run** page:
  - Our study only has eight designs, so there are only eight evaluations. The number of designs evaluated depends on the DOE method chosen (two-level full-factorial in this case) and the number of factors.
  - The values that were selected will come as no surprise, since they are the high and low values for the variable.
  - In a more complex project where design evaluations take longer to run and/or more designs are evaluated, you can check the progress of your DOE study progress bar in the lower right corner.

## Step 7: Viewing Results Using HEEDS POST

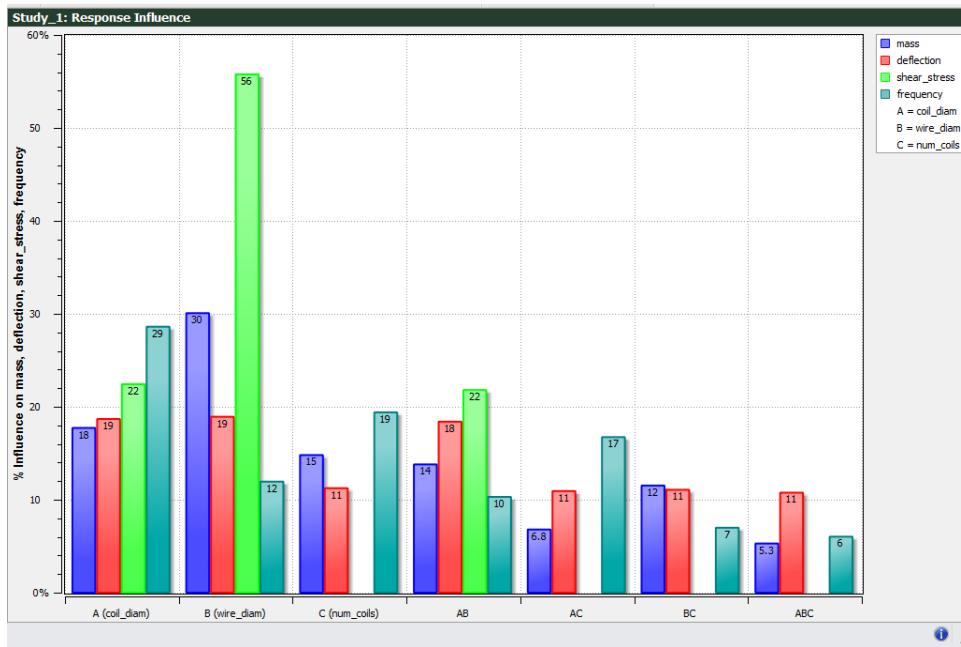
For a DOE study, the post-processing plots are where the real results are. Depending on the type of DOE you run, several different types of plots are available for each response. For our two-level full-factorial DOE, the key plots are outlined below.

### Response influence plots

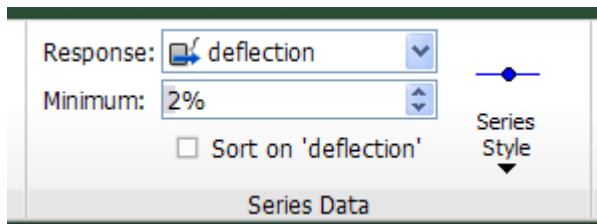
A *Response Influence* plot, also called a *Pareto Chart*, shows the percent of influence each factor and factor combination has on a given response. The combined total will equal 100%.

- On the **Ribbon**, click **Open HEEDS|post**.

2. Select the **Response Influence** plot from the list of default plots available:



3. In this chart, we can see that **wire\_diam** and **coil\_diam** have greater influence on deflection (red bars) than **num\_coils**.
4. Notice that **wire\_diam** has the biggest effect on the mass (blue bars) of the spring.
5. Notice that for **shear\_stress** (green bars), **num\_coils** doesn't have a bar shown. This is because factors or factor combinations with less than 2% influence are not shown. This can be modified adjusting the minimum in the **Series Data** part of the tool ribbon. This is set on a per series basis. (Select *All Series* to apply to all series in the plot.)

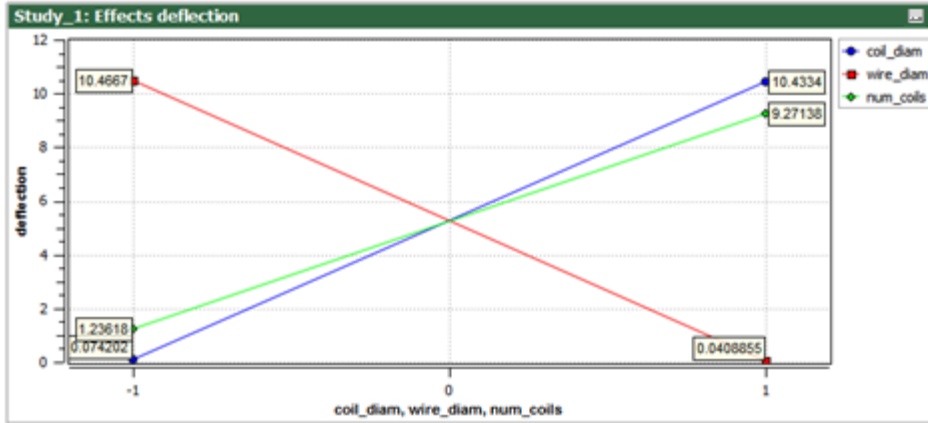


6. Also note that if the "Sort on '*response name*'" box is checked, it will sort the factor groups in order of decreasing influence. If this box is not checked for any series, it will list them in order of the factor ID.

### Factor effects plots

The effects plots show the main effect of each factor on a selected response. In these plots, you can see if, and how much, a factor influences a response.

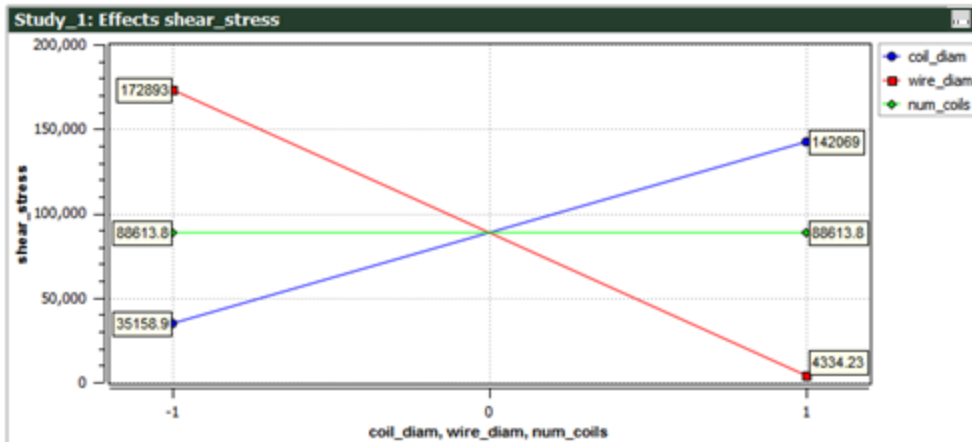
1. Click the **Effects Deflection** plot. The Variable effects for **deflection** appear:



Notice that as the wire diameter (B) increases, the deflection decreases. However, the deflection increases as the coil diameter (A) and number of coils (C) increases:

The slope of the lines correspond to the height of the bars on the *Response Influence* plot:

- The slope of **coil\_diam** and **wire\_diam** are about the same, except in the opposite direction.
  - The slope of **num\_coils** is not as steep as **coil\_diam**. You will see that the deflection increases as the number of coils increases, but the distance between high and low points is not as great as it is for **coil\_diam**.
2. Switch to the **Response Influence** plot. Note that **wire\_diam** and **coil\_diam** have nearly the same percentage influence on **deflection**. The bar for **num\_coils** is considerably shorter.
  3. Display the **Effects** plot for **shear\_stress**.

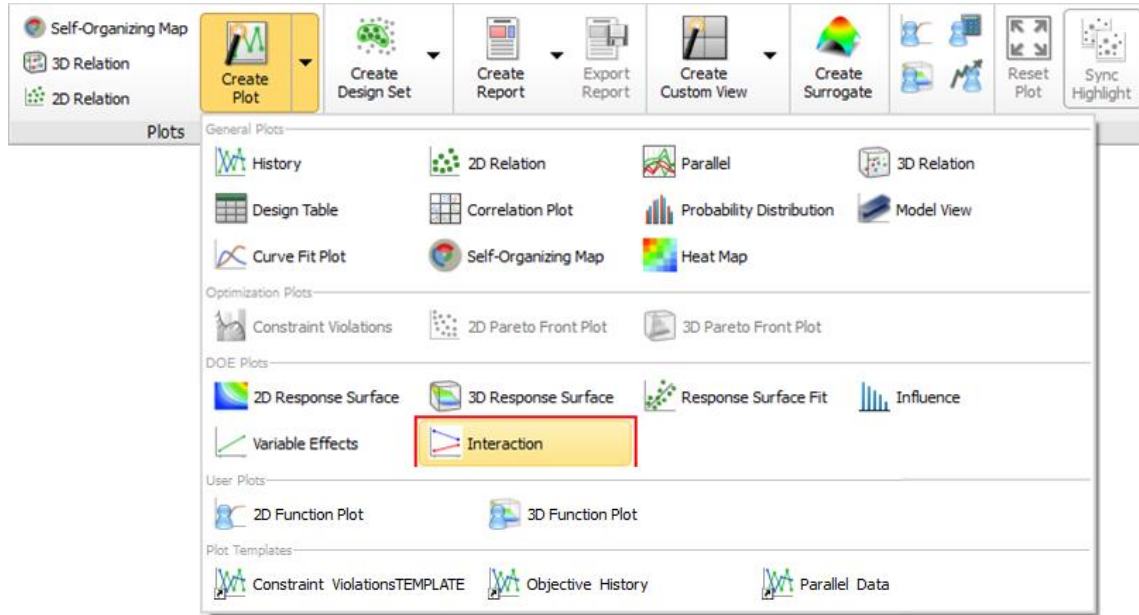


Notice that the plot for **num\_coils** is a horizontal line, as the increase in value had no effect on the response. Because the increase in value had virtually no influence on **shear\_stress**, the bar for **num\_coils** did not appear in the *Response Influence* plot at all, as mentioned earlier.

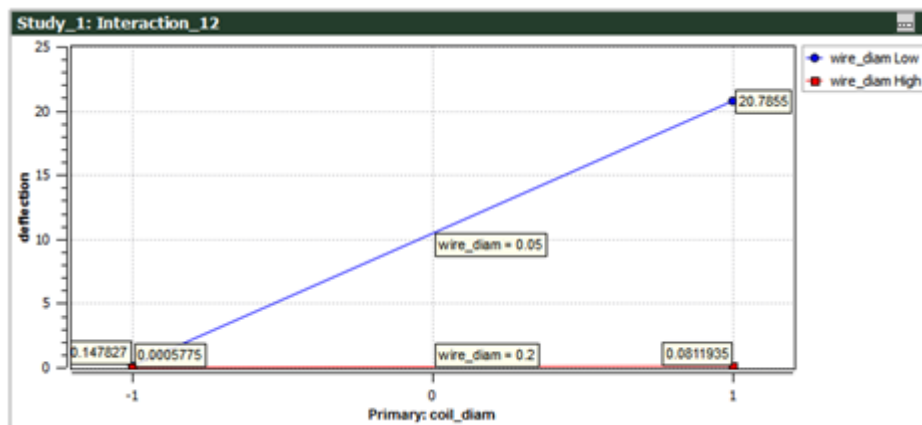
### Variable interaction plots

The *Variable Interaction* plots show the influence of the interaction between the various factors on each response. From these plots, you can tell how each factor combination affects the response. You can also tell which factors most heavily influence the others and which have little, if any, effect. As with *Variable Effects* plots, the slope of the line shows the relative impact of the influence and corresponds to the height of the bar on the *Pareto chart*.

1. On the *Ribbon*, click the variable **Interaction** plot tool from the *DOE Plots* group:

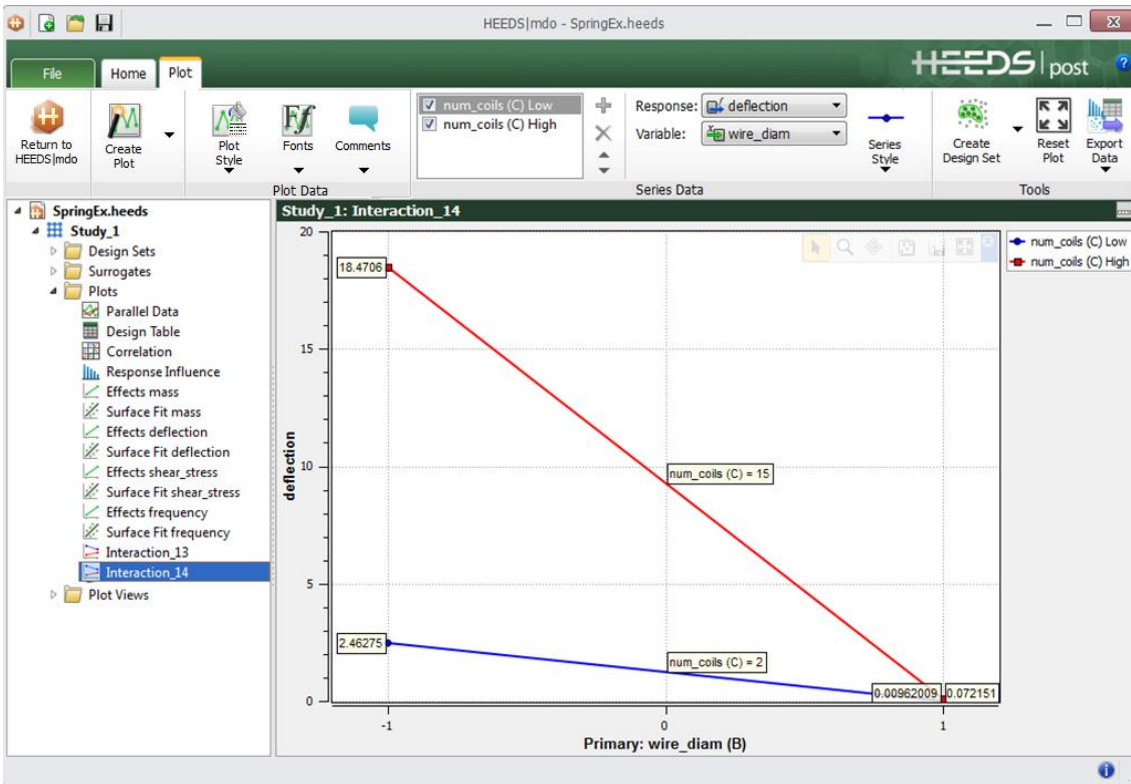


2. For the **Y-axis Parameter**, choose **deflection**.
3. For **Variable pairs**, choose **AB (coil\_diam and wire\_diam)** and click **Finish**:



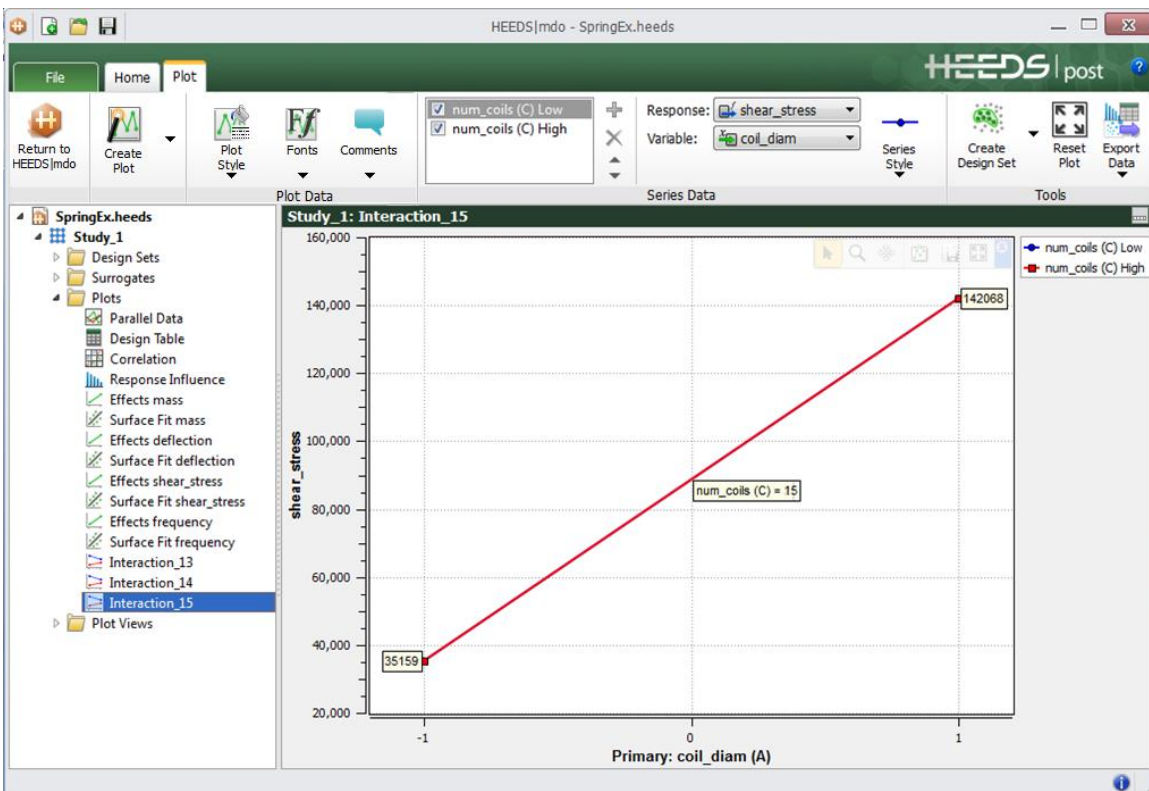
- The y-axis is **deflection**.
  - The x-axis is **A (coil\_diam)**.
  - The two lines represent the high and low values of **B (wire\_diam)** and show changes to **deflection** when **A** and **B** interact.
4. Now let's interpret the plot:
    - When **wire\_diam** is at its high value (the red line), changes to **coil\_diam** have very little (if any) effect on the deflection of the spring.
    - When **wire\_diam** is at its low value (the blue line), changes to **coil\_diam** affect deflection. Deflection increases as **coil\_diam** increases.

5. Create a new **Variable Interaction** plot, which shows the interaction effects of **B (wire\_diam)** and **C (num\_coils)** on deflection. In this case, **B (wire\_diam)** is the x-axis:



When **num\_coils** is at its high value (the red line), the interaction has a greater effect on deflection than it does at the low value (the blue line).

6. Now create a **Variable Interaction** plot for **shear\_stress** and the **AC** interaction:



- Notice that only one line is visible (actually, both lines are plotted on top of each other). This is because the high and low values are the same and the second line is plotted on top of the first. This tells us that the effect of **A (coil\_diam)** on **shear\_stress** is not dependent on **C (num\_coils)** and that **C (num\_coils)** has no effect on the **shear\_stress**.
- Display the **Response Influence** and note that for **shear\_stress** the only factor interaction that appears is AB. The number of coils has no influence on **shear\_stress**, either alone or in combination with the other factors.

### Response surface fit plot

During post-processing, HEEDS MDO calculates mathematical approximations of the design space using one or more of the following models:

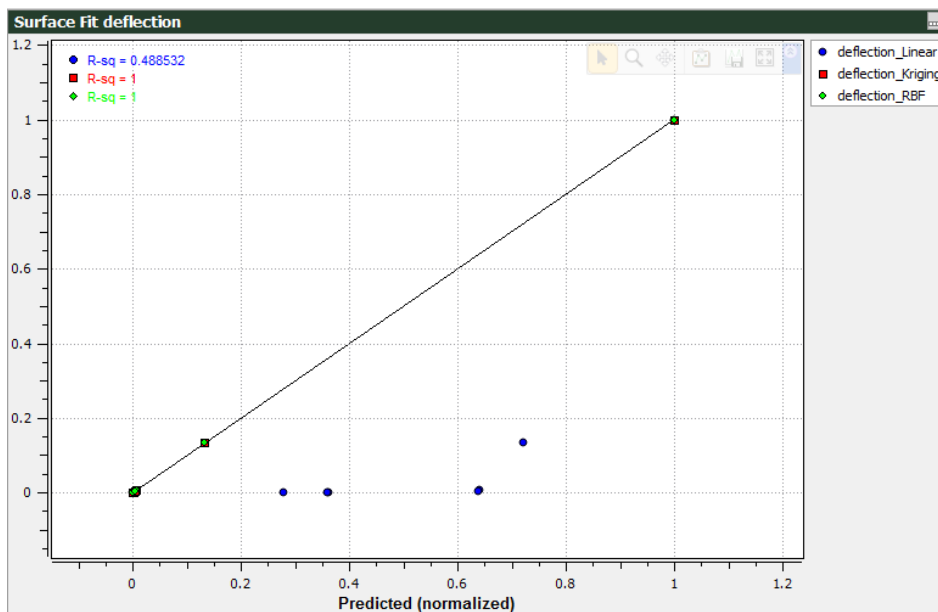
- Quadratic Response Surface* (including linear interaction terms)
- Linear Response Surface*
- Kriging Response Surface* (using a Gaussian correlation function)
- Radial Basis Function Response Surface* (using a Thin Plate Spline function)

Kriging and radial basis function response surfaces interpolate the response values using blending functions. Typically, they exactly match the response values. Kriging with a Gaussian function will usually give a rounder surface, while radial basis function with a thin plate spline function will give a somewhat flatter surface between data points.

Linear and quadratic response surfaces are least-squares-fit polynomial models. Typically, they only approximate the response values.

The *Response Surface Fit* plot shows the quality of response surfaces that were fit.

- Select the **Surface Fit deflection** plot:



- The line shows an ideal case, where the values extracted for each evaluation exactly match the values that would be predicted using the response surface. The dots show the actual plotted values:
  - If the actual values line up close to the line, the response surface is a reliable fit.
  - If most of the actual values do not line up alongside the line, the response surface is not a good representation of the design space.
  - The  $R^2$  value is also listed on the plot; a value of **1.0** is a perfect fit.
  - Kriging and radial basis function surfaces usually fall exactly on the ideal line and have a  $R^2$  value of 1.

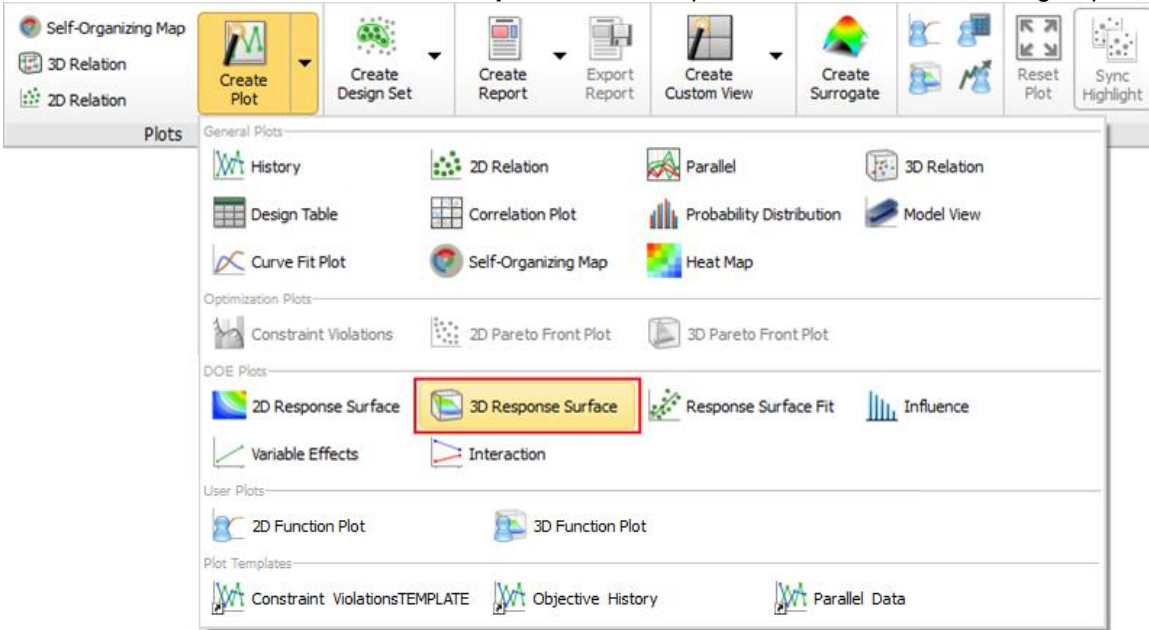
## Response surface plots

These plots let you see three-dimensional “slices” of your design space and let you experiment with the effects of variable changes on your design. The plots available for a given DOE depend on the complexity of the DOE. For two-level DOEs, quadratic response surfaces are not available.

The advantage of these plots is that you can quickly see the effect of changes to your variables. The alternative, which is much more time-consuming, would be to keep changing the variables in your model and running the study until you see your desired results.

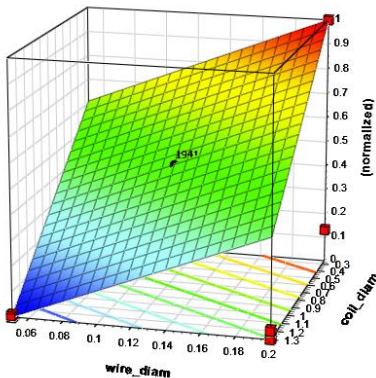
### ► To create response surface plots

1. On the *Ribbon*, click the variable **3D Response Surface** plot tool from the *DOE Plots* group:

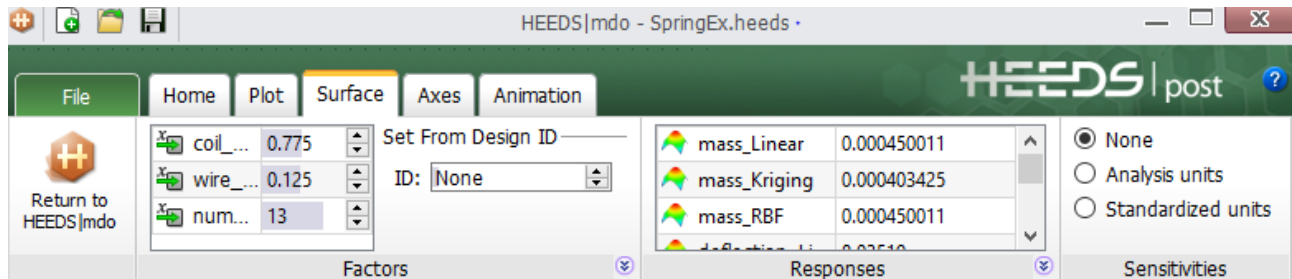


- For *X-axis parameter*, select: **coil\_diam**.
- For *Y-axis parameter*, select: **wire\_diam**.
- For *Available surrogates*, check **frequency\_Linear**.

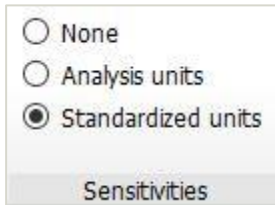
Your plot should look like this:



2. Select the **Surface** tab, and change the value of **coil\_diam** by moving the handle inside the value cell. You will see the black dot on the surface indicating the predicted value move. The value for all responses is also indicated in the table on the right:



3. Change the value of **num\_coils**. Now the response surface itself will move up and down, since **num\_coils** is not plotted. When you change the value of variables that are not plotted, you are changing the "slice" of the hypersurface displayed.
4. Cycle through the values available in the **Set From Design ID** area. The predicted response value for each evaluation can be displayed on the surface.
5. From the **Sensitivities** group, select one of the following options:



- **None** - A bar chart of the response sensitivities is not plotted. **None** is the default value.
- **Analysis units** - The sensitivity values in the bar chart are displayed in the units of the response and variables. Use this option to get actual slope values.
- **Standardized units** - The response and variable values are standardized based on the range of values in the surrogate. Use this option to easily compare the effect of different variables.

The bar chart is displayed in the upper top corner of the 3D surface plot.

5. Select the **Plot** tab and click the "+" button in the **Plot Data** area of the tool ribbon to add a second series:
  - For *Type*, select **Design data**.
  - For *Z*, select **frequency**.
  - For *Design Set*, select **All Designs**.

Now the actual points evaluated are plotted along with the surface. This plot is useful for visualizing the regions of the response surface that might have a better or worse fit.



**Note:** The response surface is only an approximation of the design space, so the information shown in the response surface plots is only as good as the surface it represents.

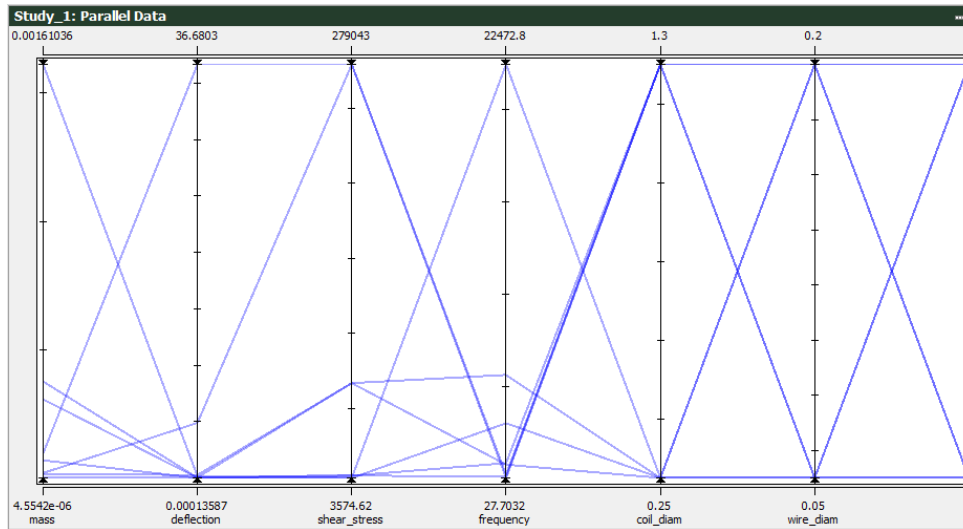
6. Animate the plot:
  - a. Select the Response Surface plot from the tree view.
  - b. Select the **Animation** tab.
  - c. Select the **num\_coils** parameter and click the **Play** button.

The response surface updates to reflect the changes in the selected parameter.

## Parallel plot

The parallel plot shows the variable and response values for all evaluations. It works the same way as the parallel plot for parameter optimization.

Click **Parallel Data**. Your plot should look something like this:



## Example 5: Robustness and Reliability

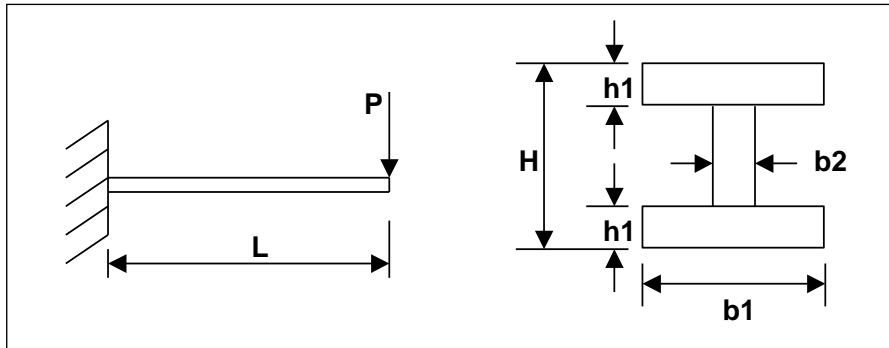


Fig. 1. Cantilevered beam

In Example 2, we set up a parameter optimization project to minimize the mass of a cantilevered beam (fig. 1). In this example, we are going to assume that we have run the parameter optimization and now want to test our best design for robustness and reliability. Instead of creating a new project from scratch, we will adapt our old one.

### The problem

We want to see how robust our design is when variations are applied to our original variables:

<b><i>b1</i></b>	Flange width
<b><i>h1</i></b>	Flange thickness
<b><i>b2</i></b>	Web width
<b><i>H</i></b>	Total beam depth

and also to the Young's modulus of the material.

### The baseline design

For this example, we will use the same *cbeam.exe* that we used for the optimization project and the same input and output files:

File Name	Description
<b>cbeam.inp</b>	The input file.
<b>cbeam.out</b>	The output file.
<b>cbeam.exe</b>	The file that contains the analysis code.

## Step 1: Copying the Project

Our first step is to create a copy of the cantilevered beam optimization project.

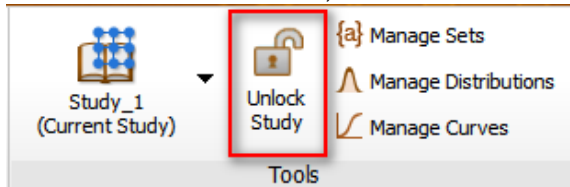
1. Create a directory on your system for your new project.
2. Select **Open** from the **File** tab, and open the **CBeam** project.
3. Open the **File** tab again, and select **Save As**. The Save As dialog box appears.
4. Browse to the directory that you have set up for this project, and then save your project as **CBeamRR**. Your project copy will be loaded into MDO as **CBeamRR**. You can make your changes to this copy without affecting the original **CBeam** project.

## Step 2: Editing the Parameters

We do not need to change our analysis definition because we are using the same tool and input and output files as for **CBeamOpt**. Since our tasks at the *Process* level are complete, we can move on to the *Parameters* level. Here our responses are complete, but our variables require changes. We need to add one variable, and then redefine all of the variables as stochastic parameter variables. After we define the variables, we will create the required stochastic distributions for them.

### Defining the project variables

1. Select the **Parameters** tab, and click **Unlock Study** > **Unlock** in the **Tools** group:



This allows you to make changes to the file.

2. Add a new variable to your project and give it the name **mat\_E**.
3. Define the project variables as follows. The baseline value will be used as the mean for your stochastic variations.

Variable Name	Type	Baseline	Comment
<b>b1</b>	Constant	5.1	Flange width
<b>h1</b>	Constant	0.25	Flange thickness
<b>b2</b>	Constant	0.1	Web width
<b>H</b>	Constant	5.8	Total beam depth
<b>mat_E</b>	Constant	1.0e7	Material property

Your variables should look like this:

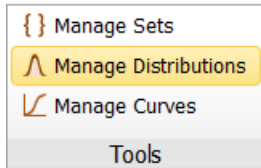
Variable Name	Type	Baseline	Distribution	Comment
1 <input checked="" type="checkbox"/> b1	Constant	5.1		Flange width
2 <input checked="" type="checkbox"/> h1	Constant	0.25		Flange thickness
3 <input checked="" type="checkbox"/> b2	Constant	0.1		Web width
4 <input checked="" type="checkbox"/> H	Constant	5.8		Total beam depth
5 <input checked="" type="checkbox"/> mat_E	Constant	1e+07		Material property

4. Save your project.

## Defining stochastic distributions

All of our variables need stochastic distributions or no variations will be applied. Once you create a stochastic distribution, it may be applied to as many variables as needed. You'll create two stochastic distributions next.

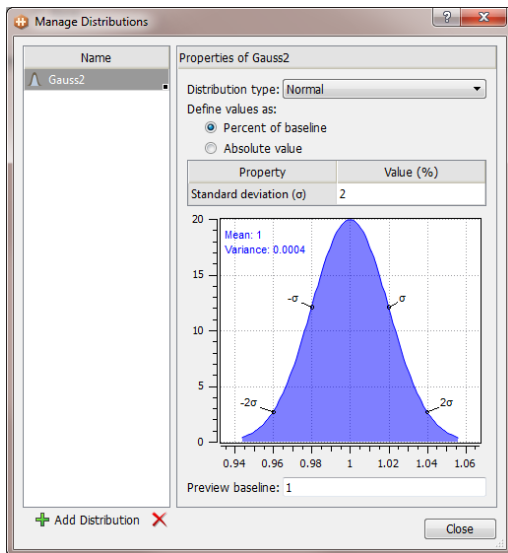
1. On the *Ribbon*, click: **Manage Distributions**:



2. On the *Distributions Manager*, add a **Distribution**, rename it **Gauss2**, and then set the *Properties of Gauss2* as follows:
  - For **Distribution type**, select **Normal**.
  - Select **Percent of baseline**.
  - Enter **2** for **Standard deviation**.
  - Enter a preview baseline value, if desired. The preview plot will center the distribution on the value you enter.

HEEDS plots the resulting probability curve and shows the computed mean and variance values for the distribution.

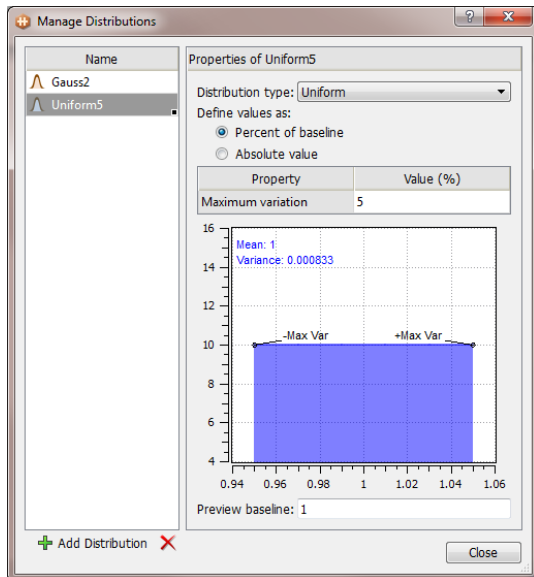
The definition for **Gauss2** should look like this:



3. Add a second **Distribution**, rename it **Uniform5**, and then define the *Properties of Uniform5* as follows:
  - For **Distribution type**, select **Uniform**.
  - Select **Percent of baseline**.
  - Enter **5** for **Maximum variation**.
  - Enter a preview baseline value, if desired. The preview plot will center the distribution on the value you enter.

HEEDS plots the resulting probability curve and shows the computed mean and variance values for the distribution.

The definition for **Uniform5** should look like this:



4. Close the *Distributions Manager*.

## Applying the distributions

Now that we've created our distributions, we can apply them to the variables.

1. On the *Variables Manager*, locate the **Distribution** column. For variables **b1**, **h1**, **b2**, and **H**, select **Gauss2**.
2. For variable **mat\_E**, select **Uniform5**.

Your variables should look like this:

Variables		Responses		
Variable Name	Type	Baseline	Distribution	Comment
1   b1	Constant	5.1	Gauss2	Flange width
2   h1	Constant	0.25	Gauss2	Flange thickness
3   b2	Constant	0.1	Gauss2	Web width
4   H	Constant	5.8	Gauss2	Total beam depth
5   mat_E	Constant	1e+07	Uniform5	Material property

3. Save your project.

## Step 3: Tagging the Input and Output Files

Our output file does not need additional tagging because we made no changes or additions to the responses. The four original variables are already tagged in the input file because we tagged them in the parameter optimization project. The variables will remain tagged, even if we change the variable type. Our only task is to tag our newly-added variable, **mat\_E**.

1. Select the **Tagging** tab, and then open **Cbeam.inp**. The parsed file is displayed.
2. On the *Ribbon*, open the drop-down for **Variable** and select **mat\_E**.

3. Tag the value at cell **A4**, for variable **mat\_E**:

File: C:/HEEDS/Ex2\_CBeam/cbeam.inp

	A	B	C
1	HEEDS	Example	Problem
2	MATERIAL	PROPERTIES:	
3	mat_E		
4	1.0e7		
5	GEOMETRIC	PROPERTIES:	
6	L	H	B1
7	36.0	H	b1
8	TIP	LOAD:	
9	P		
10	1000.0		
11			

Your variables should look like this:

File: C:/HEEDS/Ex2\_CBeam/cbeam.inp Delimiters: ☒ , ☐ : ☒ ; ☒ tab ☒ space

	A	B	C	D	E	F
1	HEEDS	Example	Problem			
2	MATERIAL	PROPERTIES:				
3	mat_E					
4	mat_E					
5	GEOMETRIC	PROPERTIES:				
6	L	H	B1	H1	B2	
7	36.0	H	b1	h1	b2	
8	TIP	LOAD:				
9	P					
10	1000.0					
11						
12						

4. Save your project.

## Step 4: Defining the Reliability Study

Assembling a robustness and reliability project is very simple. The first step is to determine which type of study to perform. If you are looking for information on whether a constraint will be violated, then a reliability study should be performed. If you want to know how much a response will vary due to stochastic variations of the input parameters, then a robustness study should be performed. We will perform both studies for this example, starting with reliability.

### Selecting the search method

1. Select the **Study** tab.
2. On the *Study Manager*, open the list box for **Study type**, and then select **Robustness and Reliability Analysis**.
3. On the **Methods** tab of the *Study Manager*, define the following:
  - For **Study Method**, select **Reliability**.
  - For **Method Properties**, select **Latin hypercube sampling**.
  - For **Number of evaluations**, enter **200**.
4. Save your project.

## Step 5: Running the Reliability Study

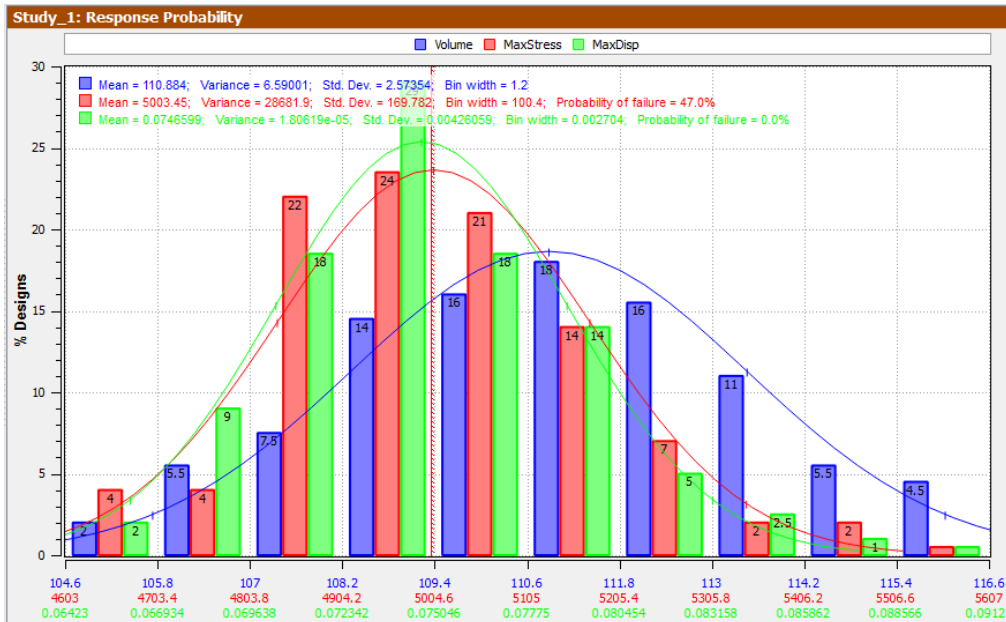
1. Select the **Run** tab.
2. On the *Ribbon*, click **Run** to start the study. Your study summary and running status appear:



Like DOE studies, robustness and reliability studies do not display performance information. However, you may find the variable and response plots useful for tracking the progress of your study.

## Step 6: Viewing the Results of the Reliability Study

The most important information that this study provides is the probability of failure. This information is available on the *Probability Distribution* plot as you see below:

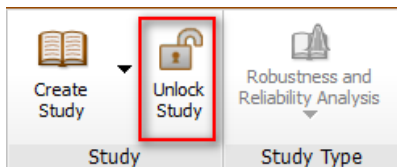


1. See [Step 9: Viewing the Results of the Robustness Study](#) for details on how a correlation plot gives more insight on this problem.
2. Animate the *Probability* plot.

## Step 7: Defining the Robustness Study

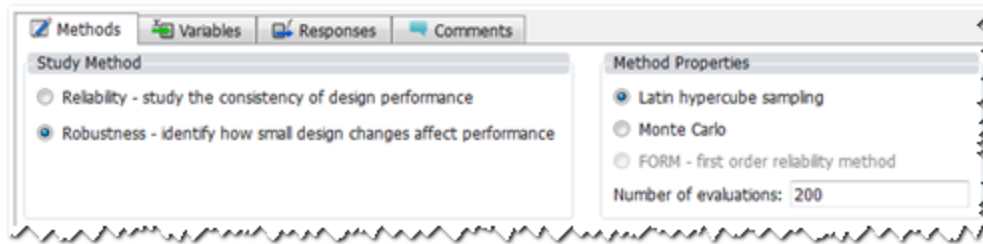
Next, we will define the study as a *Robustness* study. *Robustness* studies provide information about how much variation a response will have under specified variation of the input parameters. The input parameters do not require constraints to be defined.

1. Click the **Study** tab and return to the *Study Manager*.
2. From the *Ribbon*, click **Unlock Study > Unlock**.



3. You have only one small change to make: Locate the **Study Method** and select **Robustness**. The values in the other fields should be the same as for the *Reliability* study.

4. The *Methods View* of your *Study* definition should look like this:



5. Save your project.

## Step 8: Running the Robustness Study

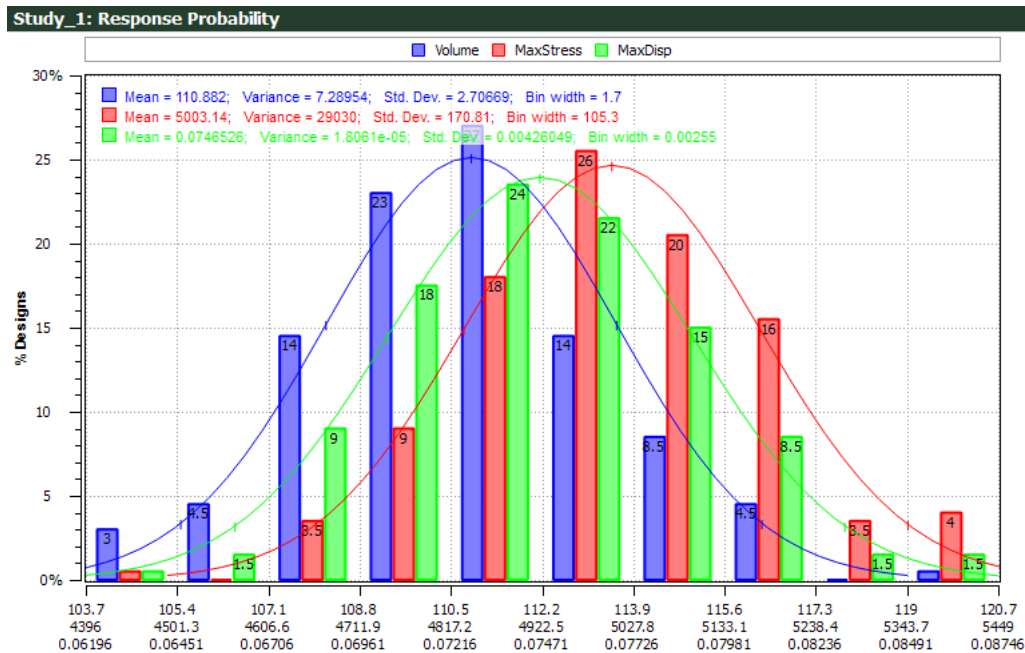
On the *Ribbon*, click **Run** to start the study. Your study summary and running status appear.

## Step 9: Viewing the Results of the Robustness Study

### Response Probability Plot

The Response Probability plot shows the response values distributions.

1. On the *Ribbon*, select **Open HEEDS|post**.
2. From the **Plots** node on the tree, select **Response Probability**. Notice that results for *Volume*, *MaxStress*, and *MaxDisp* are displayed on the same plot. The *Mean*, *Variance*, and *Standard Deviation* are shown at the top of the plot.

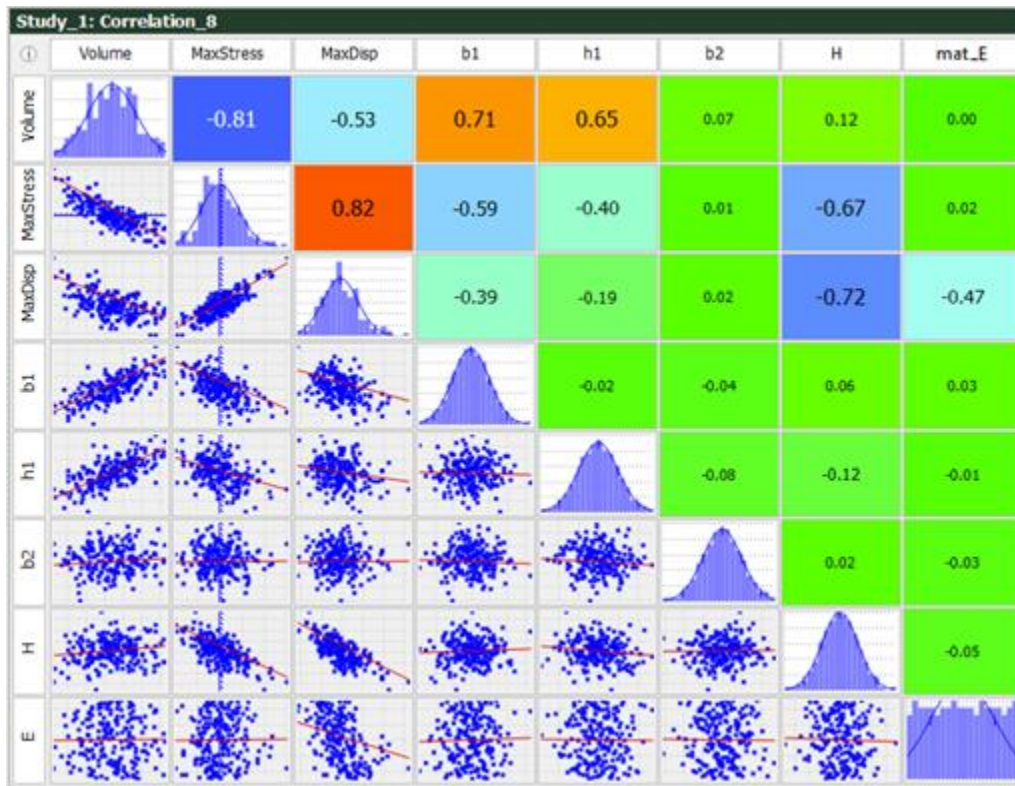


### Correlation Plot

The Correlation Plots shows the correlation values for the variables and responses.

1. On the **Home** ribbon, select **Correlation Plot** from the **General Plots** section.
2. Select **All Designs** for **Design Set**.

3. Plot all variables and responses except performance.
4. Select **Finish**.



The correlation plot has three types of information.

- The small plots on the diagonal are histograms of the variable and response values. For comparison, a normal distribution is superimposed.
- The numbers in the upper right area are the correlations between the different variable and response combinations. Values close to 1 or -1 indicate high correlations, meaning variations tend to occur together. Values close to zero indicate low correlations, meaning variations tend to occur independently.
- The small plots in the lower left area are scatter plots of the different variable and response combinations. The slope of the superimposed line is the correlation value.

Different combinations of variable and response comparisons show different aspects of the system.

The variable vs. variable comparisons show we have good evaluation samples. The variable values clearly follow the normal and uniform distributions we specified. They also show very little correlation among themselves. If the variable values did not follow the chosen distributions well, or any pairs of variables showed high correlations, then we would re-run with more evaluations or with a different random seed.

The response vs. response comparisons show the trade-offs between responses. MaxStress and MaxDisp strongly rise and fall together (high positive correlation). Volume, on the other hand, moves strongly opposite to both MaxStress and MaxDisp (high negative correlations). This suggests that MaxDisp and MaxStress can be improved together, but that any improvement will come at the expense of Volume.

The response vs. variable comparisons show that responses MaxStress and MaxDisp are most highly correlated with variable H and least correlated with b2. This suggests that modifying H will help robustness the most, while b2 is relatively unimportant.

## Example 6: Multi-Objective Optimization

A compound gear train is to be designed to meet a desired gear ratio between the driver and the gears (fig. 1). The number of teeth on each of the four gears ( $x_1$ ,  $x_2$ ,  $x_3$ ,  $x_4$ ) can be varied. The objectives are to minimize (1) the error between the designed gear ratio and the desired gear ratio of 1/6.931 and (2) the maximum number of teeth on any of the four gears.

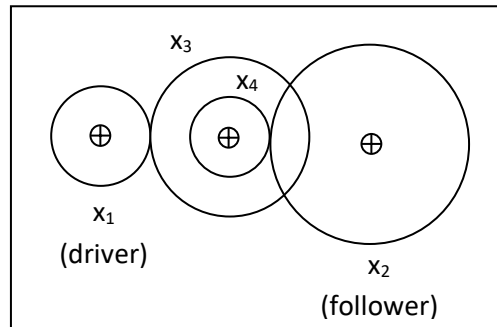


Fig. 1. An example of a compound gear train.

There are two basic approaches to solving a problem with multiple objectives:

- **If the objectives do not conflict** (i.e., they get better or worse together), a weighted sum parameter optimization, which takes all objectives into consideration, is the best approach to an optimal solution.
- **If the objectives conflict** (i.e., the conditions that cause improvement in one may affect the others adversely), a Pareto optimization can be used to study the tradeoffs between the conflicting objectives. However, Pareto optimization generally requires a larger number of evaluations to obtain an optimal design set (the Pareto front).

Since this example falls into the second category and we want to examine the tradeoffs between the two objectives, we will use the SHERPA – Multiple objective tradeoff study (Pareto front) method to perform a Pareto optimization.

### The problem

The current optimization problem is defined as:

**Minimize:** error between designed and desired gear ratio  
maximum number of teeth on any gear

**By varying:**  $12 \leq x_1, x_2, x_3, x_4 \leq 60$   
 $x_i$  represents an integer

### The baseline design

The files for this example are found in the **Geartrain** folder in the **Examples** directory:

- The default path for Windows is:  
**C:\HEEDS\MDO\Version\Examples\Geartrain**
- The default path for Linux and UNIX is:  
**/opt/HEEDS/MDO/Version/Examples/Geartrain**

If you installed HEEDS MDO in a different directory, look for the files in that directory instead. This project uses the following three files:

File Name	Description
<b>gears.inp</b>	The input file.
<b>gears.out</b>	The output file.
<b>gears.exe</b> (Windows) <b>gears</b> (Linux)	The file that contains the analysis code.

## Step 1: Starting a New Project

Before you create your project, create a directory on your system specifically for your HEEDS project. Be sure to choose a location where you have write permissions.

Start a new project, and save it in your project directory as **Geartrain**. Your project is saved as **Geartrain.heeds**.

## Step 2: Defining the Process

For this problem we will identify one analysis for our process.

### Identifying the analyses

1. Select the **Process** tab. The *Process* page appears.
2. Click the **Analysis\_1** tool from the *Process View* to display the *Analysis Manager*:

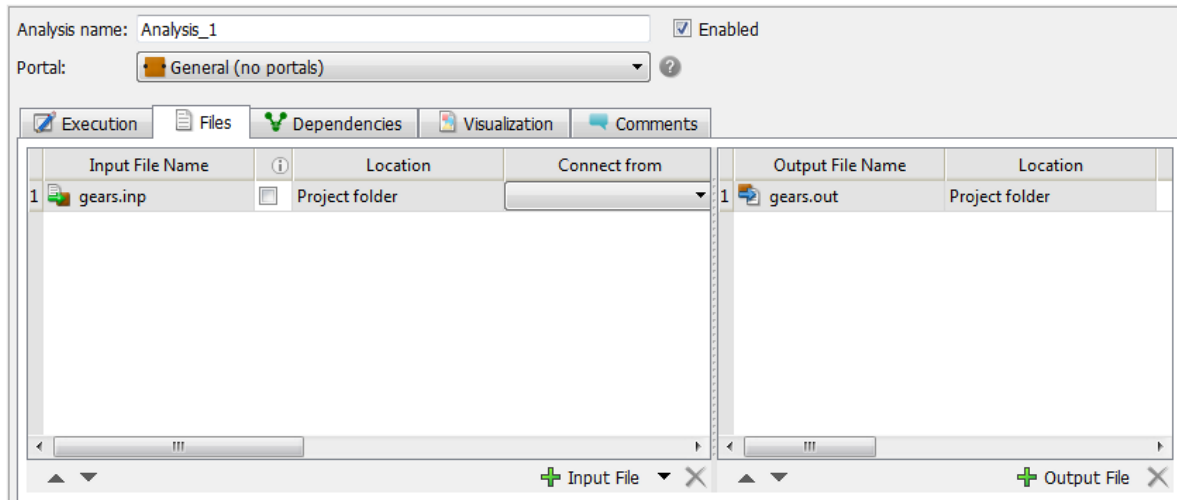


The *Analysis Manager* opens with the **Execution** tab selected:

3. Locate the **Execution command** field and click to browse for a file.
4. Browse to the **Gears** example directory and add **gears.exe** to the analysis definition.
5. Next add the input file and output file. Click the **Files** tab and then click **+ Input File**.
6. Navigate to the **Geartrain** folder, select **gears.inp**, and click **Open**. Your *Analysis* definition includes **gears.inp** as an input file.

- Click **+ Output File**, navigate to the **Geartrain** folder, select **gears.out**, and click **Open**. Your *Analysis* definition includes **gears.out** as an output file.

Your *Analysis* definition should look like this:

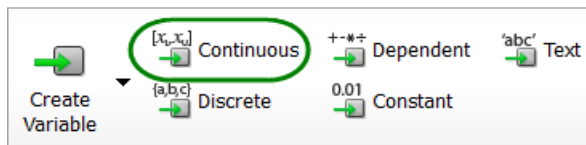


- Save your project.

## Step 3: Defining the Parameters

### Defining the project variables

- Click the **Parameters** tab. The *Parameters* page opens to the *Variables Manager*. You can also define the parameters on the **Study** tab.
- On the *Ribbon*, use the **Continuous** variable tool to add **four** continuous variables to your project.



- Define the variables as follows:

Variable Name	Type	Min	Baseline	Max	Resolution
x1	Continuous	12	40	60	49
x2	Continuous	12	40	60	49
x3	Continuous	12	40	60	49
x4	Continuous	12	40	60	49

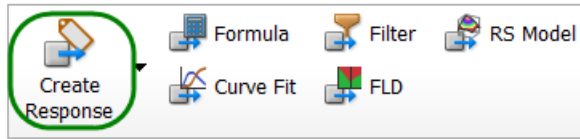
Your variables should look like this:

Variable Name	Type	Min	Baseline	Max	Resolution	Distribution
x1	Continuous	12	40	60	49	
x2	Continuous	12	40	60	49	
x3	Continuous	12	40	60	49	
x4	Continuous	12	40	60	49	

4. Save your project.

### Identifying the project responses

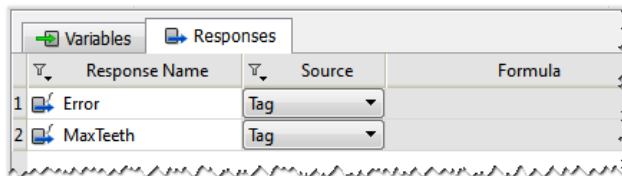
1. On the *Ribbon*, use the **Create Response** tool to add **two** responses to your project:



2. Define the responses as follows:

Response Name	Source
Error	Tag
MaxTeeth	Tag

Your responses should look like this:



3. Save your project.

## Step 4: Tagging the Input and Output Files

Next, we'll tag the variables and responses in our input file and output file.

### Tagging the input file

1. Select the **Tagging** tab. At the prompt in the *Process View*, select **gears.inp** from the dropdown. The parsed file is displayed.
2. Tag all of the variables, **x1**, **x2**, **x3**, and **x4**, to the values at location **B1**, **B2**, **B3**, and **B4**:

	A	B	C
1	x1	10	
2	x2	12.1	
3	x3	18.3	
4	x4	7.4	
5			

Your tagged **gears.inp** should look like this:

File: C:/HEEDS/Ex6\_Geartrain/gears.inp

	A	B	C
1	x1	x1	
2	x2	x2	
3	x3	x3	
4	x4	x4	
5			

### Tagging the output file

1. On the *Ribbon*, open the dropdown for **File** and select **gears.out**. The parsed file is displayed.
2. Tag the responses, **Error** and **MaxTeeth**, with the values at location **E3** and **J4**:

File: \_GettingStarted\Geartrain\gears.out Delimiters: ☒ , ☐ : ☒ ; ☒ tab ☒ space "="()

	A	B	C	D	E	F	G	H	I	J
1	RESULTS	DATA:								
2										
3	The	error	squared	is:	0.653028					
4	The	maximum	number	of	teeth	on	any	gear	is:	18
5										
6	INPUT	DATA:								
7	x1	10								
8	x2	12								
9	x3	18								
10	x4	7								
11										

The tagged **gears.out** should look like this:

File: \Geartrain\Geartrain\gears.out Delimiters: ☒ , ☐ : ☒ ; ☒ tab ☒ space "="()

	B	C	D	E	F	G	H	I	J
1									
2									
3		squared	is:	Error					
4	um	number	of	teeth	on	any	gear	is:	MaxTeeth
5									
6									
7									
8									
9									
10									
11									

## Step 5: Defining the Study

### Defining the search method

We will use the Multiple objective tradeoff method for this study. Parameter optimization methods either determine the optimal design, which is based on the weighted sum performance of all objectives, or they present a Pareto front of optimal designs, which is based on tradeoffs between the objectives. Because we want our optimization study to do the latter, we will use the Multiple objective tradeoff method. The attributes that are required for this method also control the size of the Pareto front and the number of evaluations that are performed.

1. Select the **Study** tab. The *Study Manager* appears.
2. Locate the field, **Study type**, open the list box, and select **Parameter Optimization**.
3. Next, click the **Methods** tab and define the following:

- For **Hybrid Adaptive Methods**, select **SHERPA – Multiple objective tradeoff study (Pareto front)**.
  - For **Max number of evaluations**, enter **1000**.
  - For **Archive size**, enter **16**.
4. Use the defaults for the remaining fields on the *Study Manager*.

Your completed *Study* definition should look like this:

The screenshot shows the 'Study Manager' window for 'Study\_1'. The 'Process' is 'Process\_1' and the 'Study type' is 'Parameter Optimization'. The 'Hybrid Adaptive Method' section has 'SHERPA' selected, with 'Multiple objective tradeoff study (Pareto front)' chosen. The 'Method Properties' section shows 'Max number of evaluations' set to 1000 and 'Archive size' set to 16. The 'Run Options' section has 'Save restart data after each evaluation' checked and 'Pre-set options' selected. The 'Saved Designs' section shows 'Last best design' as the success design and 'Rename' as the error design.

5. Save your project.

### Completing the response definition

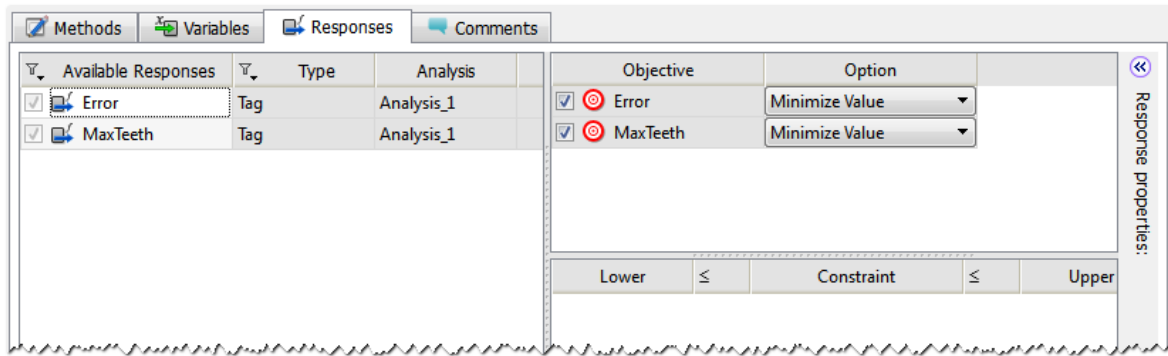
In this procedure, we will define two objectives for our design study.

1. On the *Study Manager*, select the **Responses** tab.
2. Select **Error** and **MaxTeeth** from the *Responses View*, and then drag and drop them into the *Objective View*:

The screenshot shows the 'Responses' tab in the 'Study Manager'. The 'Available Responses' list on the left contains 'Error' and 'MaxTeeth', both with checkboxes. The 'Type' column shows 'Tag' for both, and the 'Analysis' column shows 'Analysis\_1'. An orange arrow points from the 'MaxTeeth' row to the 'Objective' field in the 'Response properties' table on the right. The table has columns for 'Objective', 'Option', and 'Constraint'. The 'Option' field is currently empty, and the 'Constraint' field has a dropdown menu with 'Lower', '≤', 'Upper', and '≥' options.

3. For the **Option** field, use the default, **Minimize Value**.

Your completed *Responses* definition should look like this:



4. Save your project.

## Step 6: Running the Study

You are ready to execute your study. Select **Run** from the tab bar. The *Run* page appears with a summary of your *Study Details* and *Optimization Statement*.

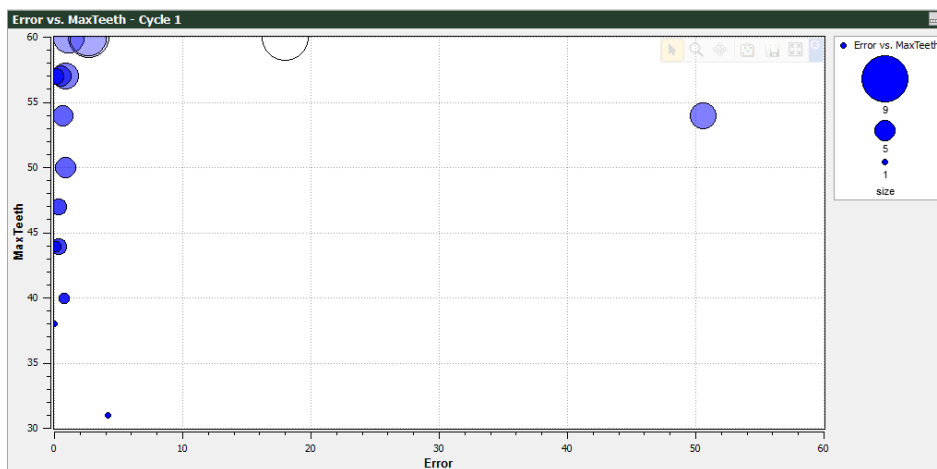
### To run the study

1. On the *Ribbon*, click **Run**. The *Run* page shows you the progress of your running study along with summary information.



**Note:** Your results may vary during a run. If your study does not run, make sure you **Save** it, and then try again.

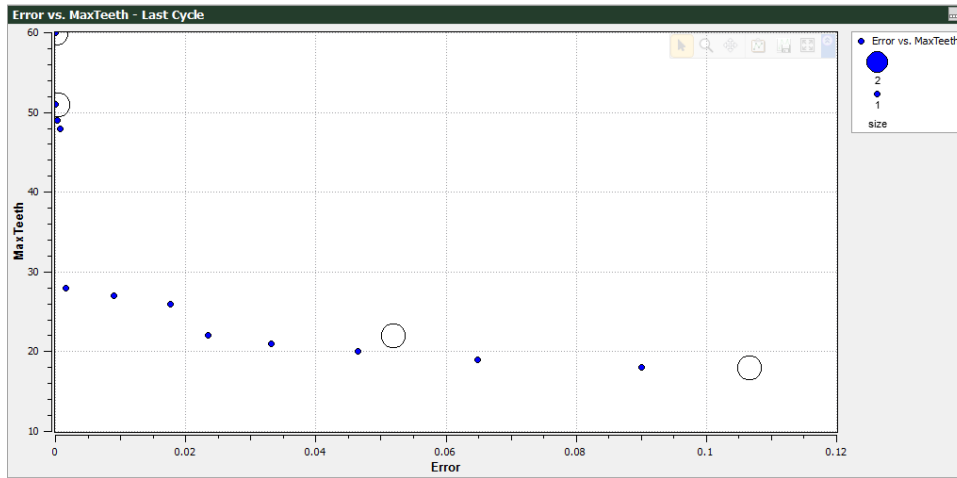
2. From the **Plots** node on the tree, select **Error vs. MaxTeeth**. This is where the results of your Pareto optimization will be displayed. The Pareto front is archived at the end of each cycle.
3. Open **HEEDS POST** and view **Error vs. MaxTeeth**. Locate the controls in the *Plot* tool ribbon tab. Locate the **Cycle** field and enter **1**. For **Rank**, select **All**. At this early stage of the run, the front will be scattered. Your plot should look similar to the one below:



Each dot on the plot is a design that has been included in the Pareto front.

4. Look at the Pareto front for several of the later cycles. Do this by increasing the cycle number (start at about **10**) in the *Cycle* field. You can use your up/down arrow keys or use the slider next to the *Cycle* field box. Notice that the designs in the front are less scattered and are falling closer to a Pareto front curve as the run progresses.
5. Look at the plot once the study has finished running. You will see a dot on the plot for every design that was included for a given cycle during the run.

6. Select the last cycle that ran, and you will see a typical Pareto front curve:



## Step 7: Viewing the Results

In addition to the *Pareto front* plot, all other parameter optimization plots are available for a Multiple objective tradeoff (*Pareto optimization*) run.

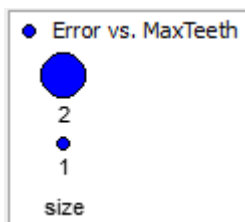
### The Pareto front plot

Before we examine the *Pareto front* plot further, we need to have a better understanding about how designs are selected for a *Pareto front*. A *Pareto* optimization selects the optimal designs based on whether they dominate other designs. A design is said to dominate another when it is better in at least one objective and not worse in all other objectives. During the run, feasible designs that are not currently dominated by any other designs are given the rank of 1. Then the remaining designs are re-ranked and those that are not dominated by any other design in that group are given the rank of 2. The procedure is repeated, re-ranking the remaining designs to establish ranks 3, 4, etc.

The set of designs that is not dominated by any other designs yet found is called the *Pareto set*. The set of designs that is not dominated by any designs in the entire search space is known as the *Pareto front*. As the run progresses, the *Pareto sets* typically continue to approach the *Pareto front* (the set of ideal solutions). For two-objective studies, the results are reported in the *Pareto front* plot. There is one *Pareto front* plot for each cycle.

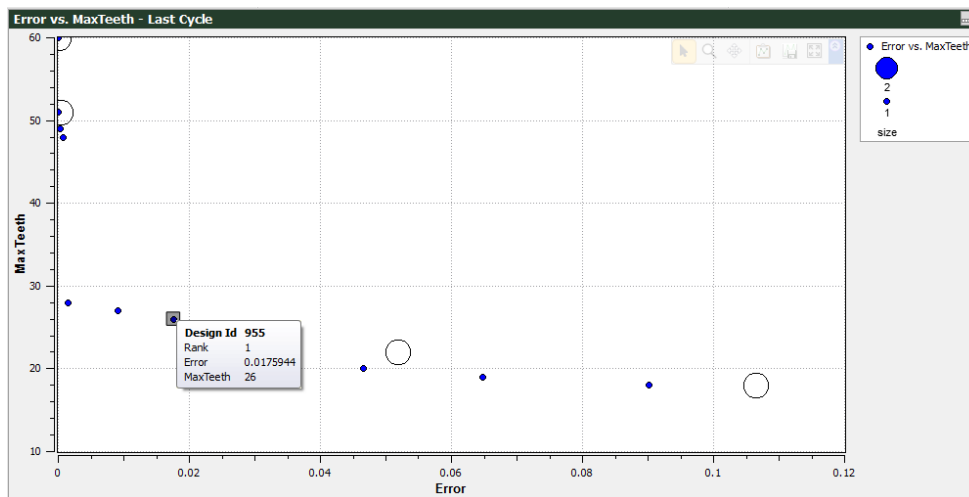
The *Pareto front* plot is a plot of the values of the two objectives. In our plot, **Error** is represented on the horizontal axis and **MaxTeeth** on the vertical axis. Usually, the designs with the best tradeoffs are found around the “knee” of the curve. The designs at the ends of the curve are more optimal for one objective but are only adequate for the other.

During the early stages of a run, the *front* will have a more random pattern. Usually dots from rank 2 or below, as well as the dots from rank 1, will appear. If there are not enough non-dominated designs from rank 1, rank 2 designs will be added. If the number of non-dominated designs in rank 2 does not fill the archive size, then rank 3 will be used, etc. Each rank has a corresponding size, which is shown in the plot legend:



When you select a cycle, the designs that were included in the *Pareto front* for that cycle will appear on the plot.

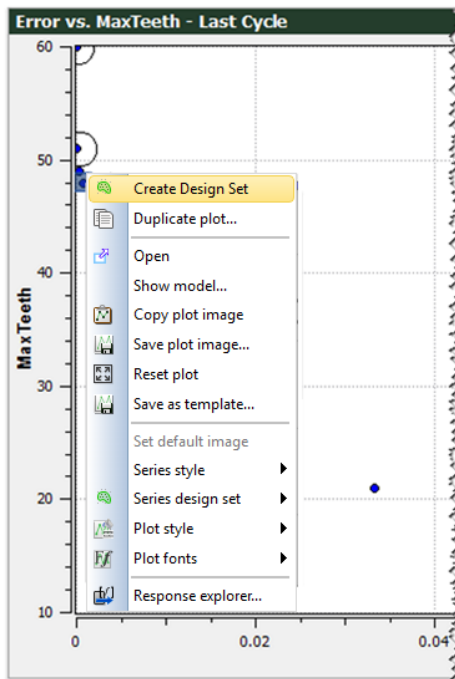
1. On the **Error vs. MaxTeeth** plot, hover over an evaluation design point from the last cycle. A popup displays the details of the design selected:



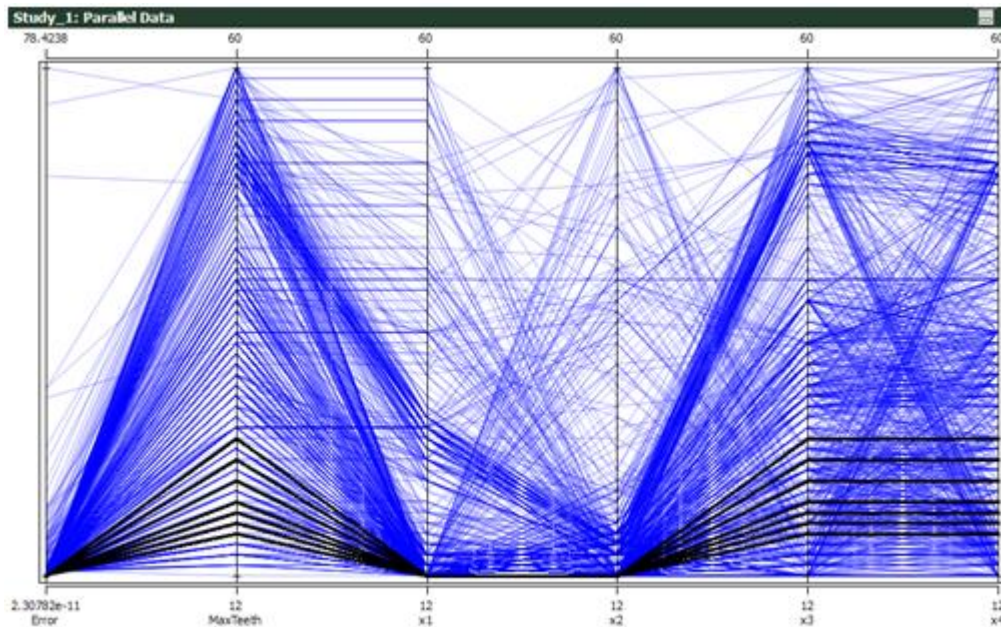
**Note:** The scale on the horizontal and vertical axes changes to accommodate the results of each cycle. To compare how the Pareto front changes between cycles, you can use the **Freeze Plot Range** tool to keep the scale the same for all cycles.

2. You can select one or more designs on the *Pareto front* plot and right-click to create a design set from the selected points. Try this now.
3. Select the points shown on the following plot. These points fall at the knee of the curve where the tradeoff between *Error* versus *MaxTeeth* is acceptable for our problem.

- Next, right-click and select **Create Design Set**: A new design set is created. (The design set shown below was saved as **DesignSet\_5**.) You can use the new design set to create additional plots for more detailed post processing of the designs of interest:

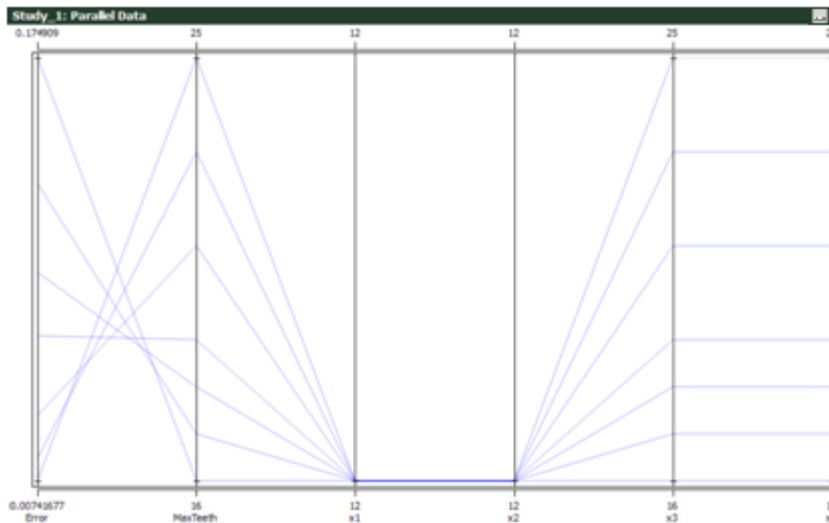


- Under the **Plots** node on the tree, select **Parallel Data**. The *Study\_1: Parallel Data* plot appears displaying the default, **Feasible Designs**, for the *Design Set*:



- Next, try creating a parallel plot using your new set: **DesignSet\_5**.

- On the *Study\_1: Parallel Data* plot, from the *Series Data* area of the tool ribbon, locate the **Design Set** dropdown and select **DesignSet\_5**. A *Parallel Data* plot for *DesignSet\_5* appears:



- Next, create a *Design Table* using **DesignSet\_5**. Under the **Plots** node on the tree, select **Design Table**. The *Study\_1: Design Table* appears.
- On the *Study\_1: Design Table*, from the *Series Data* area of the tool ribbon, open the **Design Set** dropdown and select **DesignSet\_5**. The *Design Table* updates for **DesignSet\_5**.
- Animate the Pareto plot.
- Save your project.

Because the study specifies an archive size of 16, the Pareto plot shows up to 16 rank-1 designs. But, the study contains 23 rank-1 designs. To see all the rank-1 designs, use the Non-dominated Sort tool.

### Performing a non-dominated sort

Before using the non-dominated sort feature, ensure that the design set you choose contains two or more designs.

- Create a performance criteria by opening the **Home** tab and clicking **Create Performance Criteria** in the **User Data** area. The performance criteria will match your study criteria. The performance criteria that you create are displayed under the User Data folder in the tree view.
- In the tree view, right-click the design set or the performance criteria and choose **Non-dominated sort**. The **Perform Non-dominated Sort** dialog box appears.
- The **Design set** drop-down list shows the design that you selected for sorting and the **Performance criteria** drop-down list shows the performance criteria that you defined for this design set.
- Specify a value in Number of ranks. HEEDS POST creates a new design set for each rank.
- Select **Create plot** to create a plot of the design sets.

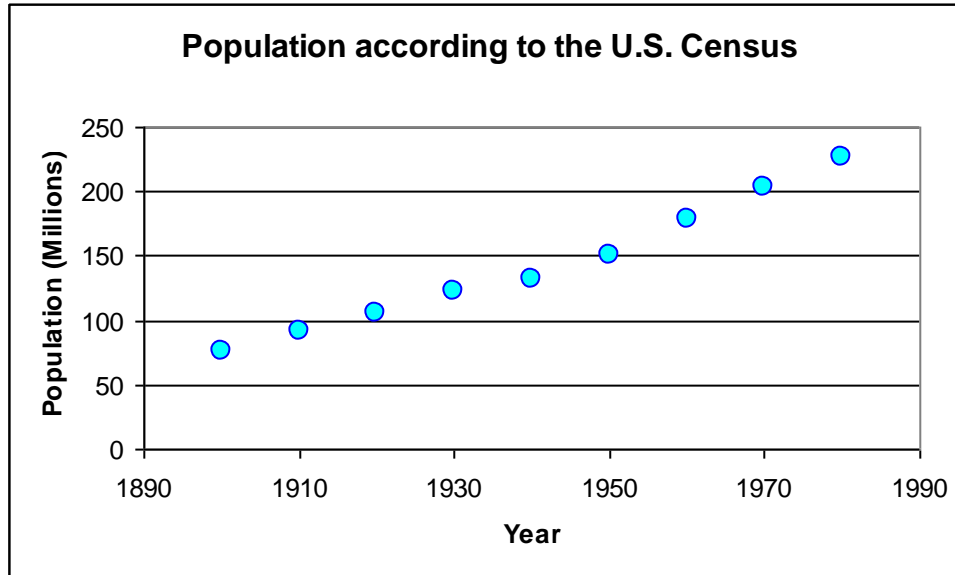
HEEDS POST creates a 2D relation plot with a series for each rank.



## Example 7: Population Modeling Using Curve Fitting

In this parameter optimization problem, the United States population will be modeled using the curve fitting option within HEEDS MDO. Variables defining a function that models the population will be determined through optimization with HEEDS MDO. This example shows you how to set up a project that utilizes the curves utility within HEEDS MDO.

Fig. 1. Population of the United States from 1900-1980 according to the U.S. Census.



### The problem

The current optimization problem is defined as:

**Minimize:** Difference between predicted population curve and actual population curve

**By varying:**

1000	≤	<i>L</i>	≤	1500
10	≤	<i>c</i>	≤	20
-0.1	≤	<i>r</i>	≤	0.1

### The baseline design

For this example, an executable named *Population\_modeler.exe* was created to solve for the population of the United States for the years 1900, 1910, 1920, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, 2010, 2020, 2030, 2040, and 2050, using the following equation:

$$P = \frac{L}{1 + ce^{r(t-t_0)}}$$

where: *P* is the population for a given year,  
*L* is a variable,  
*c* is a variable,  
*r* is a variable,

$t$  is the year the population is being predicted for,  
 $t_0$  is the year 1900.

The input file used for this executable contains values for  $L$ ,  $c$ , and  $r$ .

The files for our example are found in the **Population\_Modeling** folder in the **Examples** directory:

- The default path for Windows is: **C:\HEEDS\MDO\Version\Examples\Population\_Modeling**
- The default path for Linux and UNIX is: **/opt/HEEDS/MDO/Version/Examples/Population\_Modeling**



**Note:** If you installed HEEDS MDO in a different directory, look for the files in that directory instead.

This example uses the following four files:

File name	Description
<b>Input_data.in</b>	The input file
<b>Predicted_pop.out</b>	The output file
<b>curve.csv</b>	The file that contains the population data to be matched
<b>Population_modeler.exe</b> (Windows) <b>Population_modeler</b> (Linux)	The file that contains the analysis code

## Step 1: Starting a New Project

Start a new project, and save it in your project directory as **Predicting\_Population**. Your project is saved as **Predicting\_Population.heeds**.

## Step 2: Defining the Process

We will define our *Process* with one analysis, one input file, and one output file.

### Identifying the analyses

1. Select the **Process** tab and then click the **Variables** tab under **Study\_1**, if necessary. The *Process* page appears.

- Click the **Analysis\_1** tool from the *Process View* to display the *Analysis Manager*:



The *Analysis Manager* opens with the **Execution** tab selected:

Analysis name: Analysis\_1 ☒ Enabled

Portal: General (no portals) ?

Execution Files Dependencies Visualization Comments

Compute resource: Local

Execution command:

Command options:

Num. designs to execute simultaneously: 1

Analysis Execution Options

Run in: Analysis folder

Run condition: Always Run

Finished condition: None

Success condition: None

Advanced: Default

- In the **Execution command** field, browse to **Population\_modeler.exe** and add it to the analysis definition.
- Click the **Files** tab and then click **+ Input File**. Add **Input\_data.in**.
- Click **+ Output File**, and then add **Predicted\_pop.out**.

Your *Analysis* definition should look like this:

Analysis name: Analysis\_1 ☒ Enabled

Portal: General (no portals) ?

Execution Files Dependencies Visualization Comments

Input File Name	Location	Connect from	Output File Name	Location
1 Input_data.in	Project folder		1 Predicted_pop.out	Project folder

+ Input File + Output File

- Save your project.

## Step 3: Defining the Parameters

This example utilizes a *Target Curve* to define a *Curve Fit* response. The *RMS Curve Fitting* type is used to compare curves.

### Defining the project variables

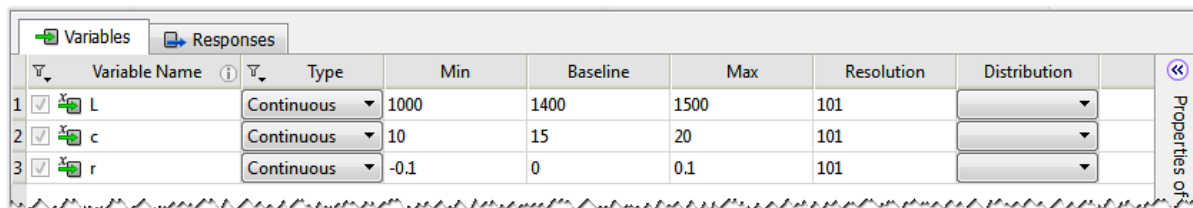
1. Click the **Parameters** tab. You can also define the parameters on the **Study** tab.

The *Parameters* page opens to the *Variables Manager*:

2. Add **three** continuous variables to your project and define them as follows:

Variable Name	Type	Min	Baseline	Max	Resolution
L	Continuous	1000	1400	1500	101
c	Continuous	10	15	20	101
r	Continuous	-0.1	0.0	0.1	101

3. Use defaults for the remaining fields. Your variables should look like this:



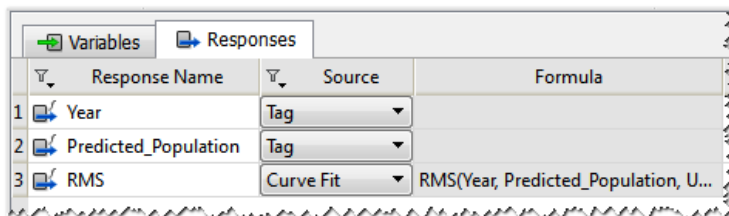
Variable Name	Type	Min	Baseline	Max	Resolution	Distribution
1 L	Continuous	1000	1400	1500	101	
2 c	Continuous	10	15	20	101	
3 r	Continuous	-0.1	0	0.1	101	

### Identifying the project responses

Add three responses to your project and define them as follows:

Response Name	Source
Year	Tag
Predicted_Population	Tag
RMS	Curve Fit

The *Responses* definition should look like the one below. When we finish adding the *Curve Fit* definition to the RMS response, we will come back to the RMS response and define it further in the *Properties* panel.

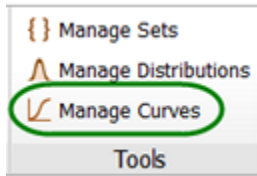


Response Name	Source	Formula
1 Year	Tag	
2 Predicted_Population	Tag	
3 RMS	Curve Fit	RMS(Year, Predicted_Population, U...

#### ► Defining a Curve Fit

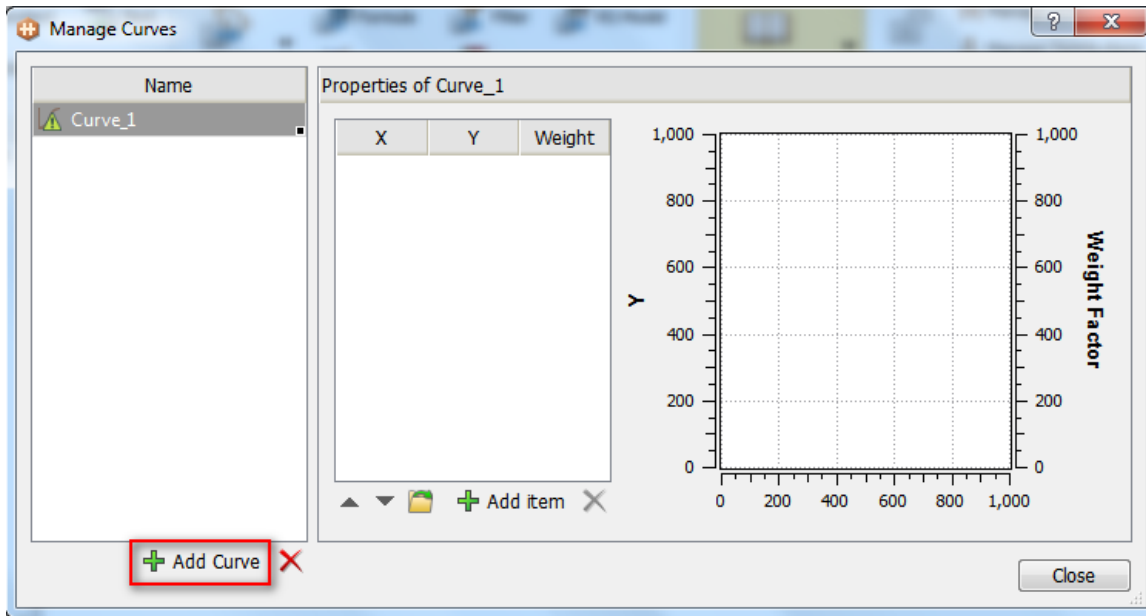
The RMS response has a *Curve Fit* as its source. The *Curve Fit* requires that we assign a *Curve Fit* definition to the RMS response.


1. On the *Ribbon*, select **Manage Curves**:

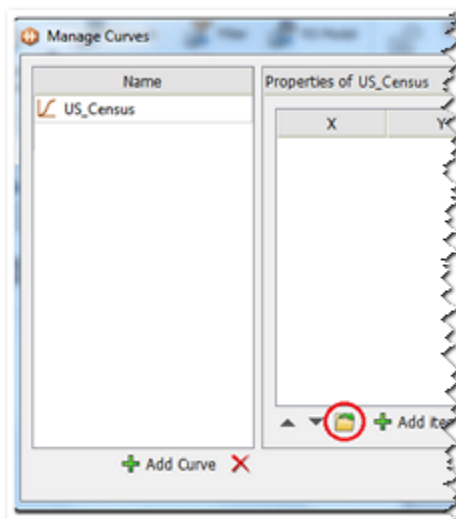


The *Curves Manager* appears.

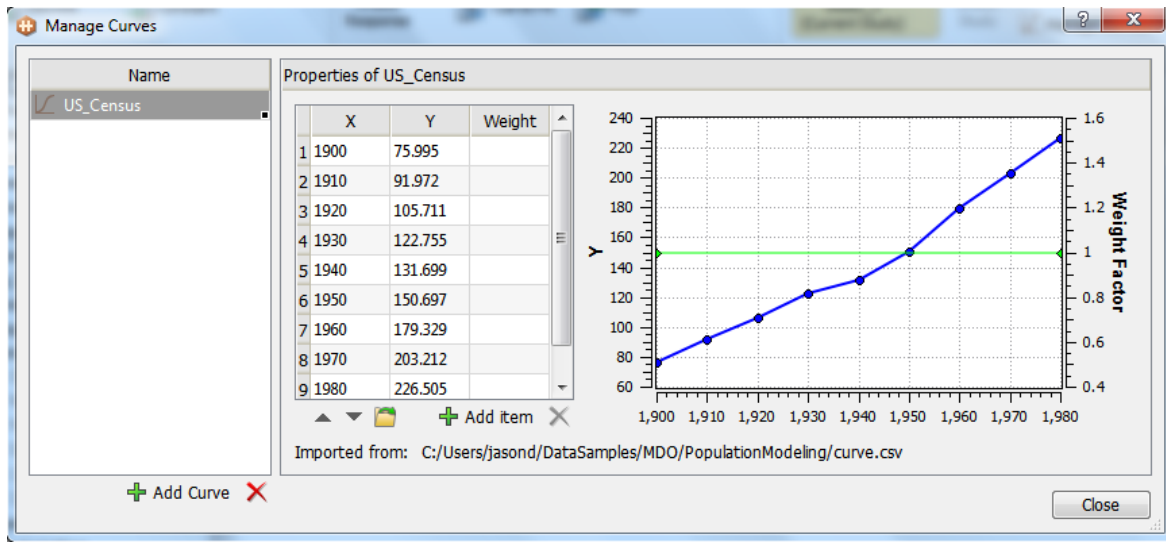
2. On the *Curves Manager*, select **+ Add Curve**. **Curve\_1** is added as a curve:



3. Rename **Curve\_1** to **US\_Census**.
4. Next, we'll import **curve.csv** to populate the values for **US\_Census**. Just below the *Properties* box on the *Curves Manager*, click  to browse for a file:



5. Browse to the **Population\_Modeling** folder and add **curve.csv**. This file provides the values for  $X$  and  $Y$ , where  $X$  is the year, and  $Y$  is the U.S. population, in millions, according to the U.S. Census. The *Properties of US\_Census* are defined as follows:



6. **Close** the *Curves Manager*.
7. Select the **RMS** response.

1. Click open the *Properties* panel and from the dropdowns, add the **Design curve definition**, **Reference curve**, and **RMS calculation method option** properties as follows:

**Properties of RMS**

Specify the source analysis:  
 Analysis\_1

Set design to ERROR if value is outside: ⓘ  
 Minimum:   
 Maximum:

**Design curve definition:**

X values: Year  
 Y values: Predicted\_Population

**Reference curve:**  
 US\_Census

☐ Normalize difference ⓘ

**RMS error calculation option:**

☒ Use all points  
☐ Use only points below reference curve  
☐ Use only points above reference curve

Legend:  
 ● Reference curve  
 ■ Active design points  
 ■ Inactive design points

Use this option if the goal is:  
 - To match the reference curve. Also define an objective to **minimize** this response.

$$\sqrt{\frac{1}{N} \sum_{i=1}^N (F_i * (Y_i - y_i)^2)}$$

**RMS** is now defined as the root-mean square between two curves. One curve is defined by the project responses, **Year** and **Predicted\_Population**, while the other curve is defined by the target curve, **US\_Census**. Specifying **Use all points** will use the response as a minimized objective to match the target curve as closely as possible.

2. Save your project.

## Step 4: Tagging the Input and Output Files

Next, we must tag the variables and responses in our input and output files. For the input file, we will use the marking method. For the output file, we will use scripting.

### Tagging the input file

1. Select the **Tagging** tab. Open the dropdown and select **Input\_data.in**. The parsed input file is displayed.

- Tag variables **L**, **c**, and **r** to the values at location **A2**, **B2**, and **C2**:

File: C:/HEEDS/Ex8\_Population/Input\_data.in

	A	B	C	D
1	L	C	R	
2	1481.53	17.8261	0.0146898	
3				

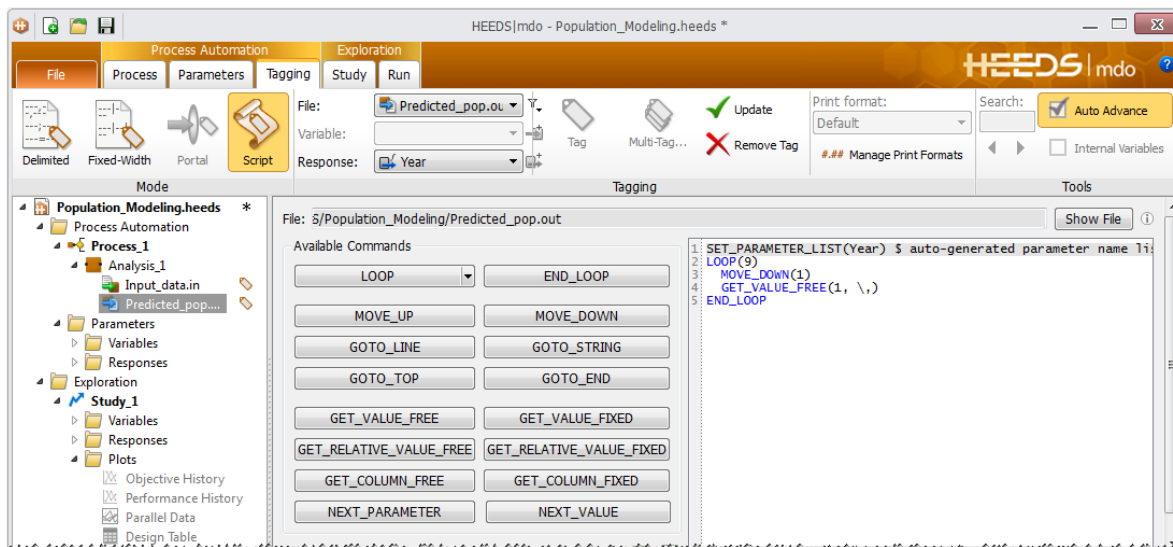
Your tagged variables should look like the following:

File: C:/HEEDS/Ex8\_Population/Input\_data.in

	A	B	C	D
1	L	C	R	
2	L	c	r	
3				

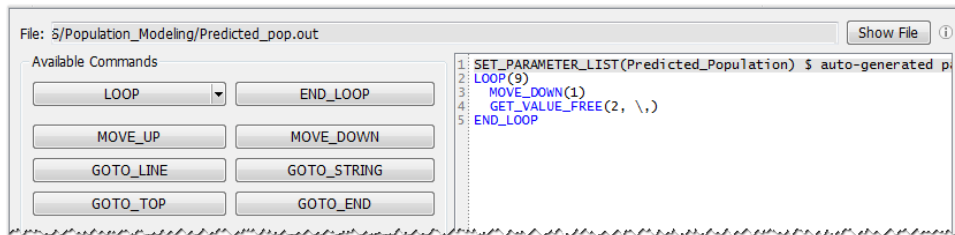
### Tagging the output file

- On the *Ribbon*, open the **File** dropdown and select: **Predicted\_pop.out**. The parsed output file is displayed. We'll tag it using the *Scripting* method.
- On the *Ribbon*, open the dropdown for **Response** and select **Year**.
- On the *Ribbon*, select the **Script** tool, and then enter the script below:



- On the *Ribbon*, click **Tag** and then click **OK**. **Year** is tagged with the script.

- Now in the same manner, tag the response **Predicted\_Population**. Use the following script:



Your tagged responses should look like the following:

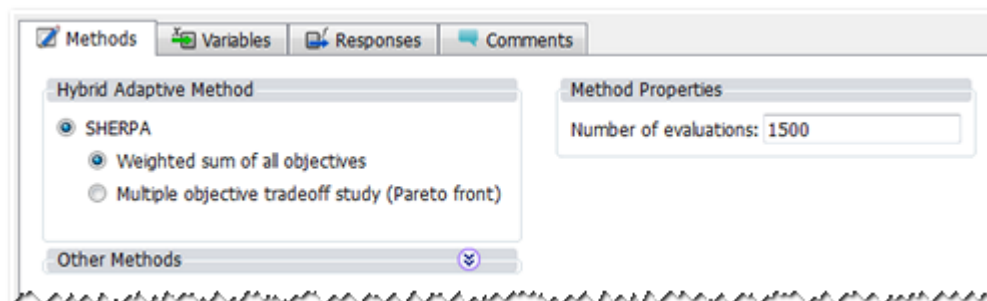
	Parameter	Mode	Data
1	Year	Script	LOOP(9)...
2	Predicted_Population	Script	LOOP(9)...

- Save your project.

## Step 5: Defining the Study

### Selecting the search method

- Select the **Study** tab. The *Study Manager* appears.
- For the search method, use the default, **SHERPA – Weighted sum of all objectives**.
- For **Number of evaluations**, enter **1500**. The completed *Methods* definition for your study should look like this:



- Save your project.

### Completing the response definition

We will define one objective for this study:

- On the *Study Manager*, select the **Responses** tab.
- Select the **RMS Response** and drag it to the *Objective View*. **RMS** is added as an *Objective*.

- Use the defaults for the remaining fields on the *Study Manager*. Your completed *Study* definition should look like this:

✓ The study has passed all validity checks and is ready to [run](#).

Study name:

Process:

Study type:

Saved Designs

Success designs:

Error designs:

Run Options

☒ Save restart data after each evaluation

☐ Do not stop HEEDS for a design-based error ⓘ

Random seed:

☒ More options ▾

Methods Variables Responses Comments

Available Responses	Type	Analysis
<input checked="" type="checkbox"/> Year	Tag	Analysis_1
<input checked="" type="checkbox"/> Predicted Popula...	Tag	Analysis_1
<input checked="" type="checkbox"/> RMS	Curve Fit	Analysis_1

Objective	Option
<input checked="" type="checkbox"/> RMS	Minimize Value ▾

Response properties: ⏪

Lower	≤	Constraint	≤	Upper

## Step 6: Running the Study

On the *Run* page, click **Run**.

## Step 7: Viewing the Results

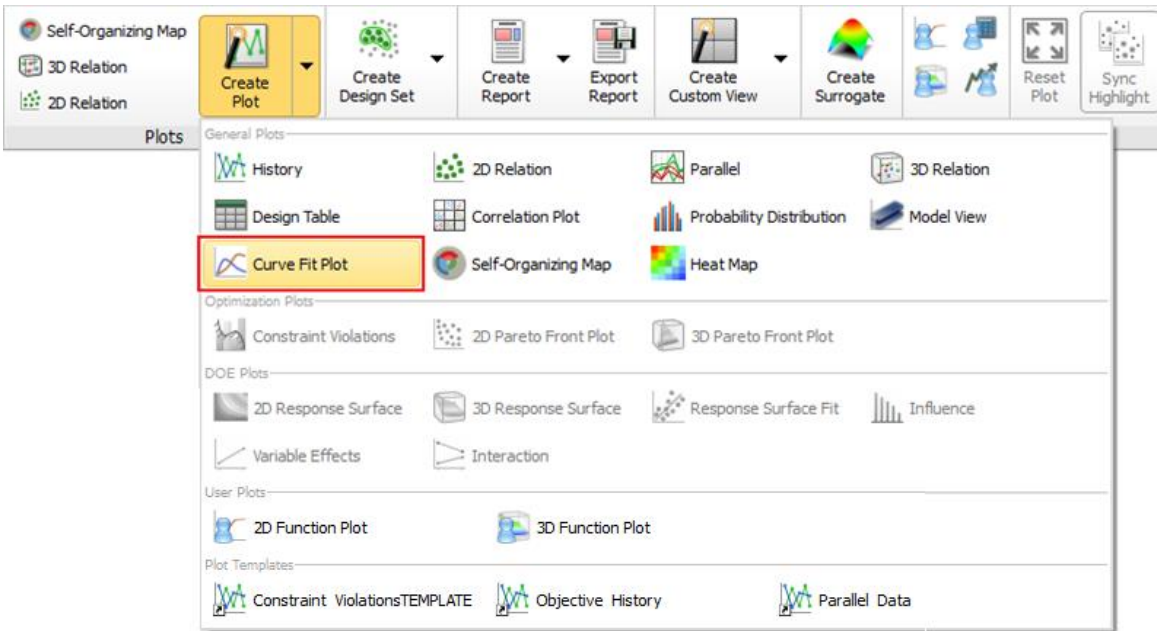
HEEDS MDO will find designs that lower the RMS value from the baseline design of 220.94 to around 8.9. We will create a *Curve Fit* plot to show an optimal design.



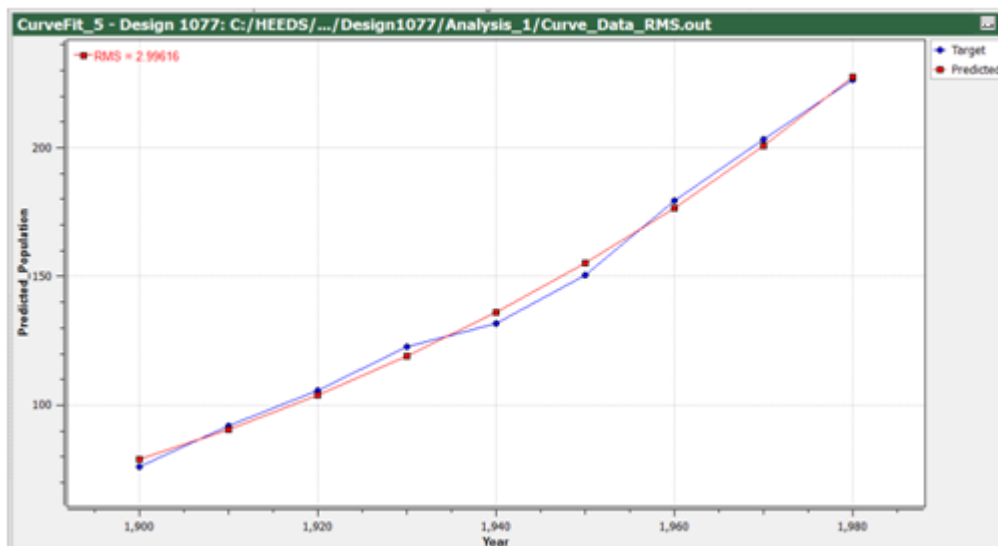
**Note:** Your results may vary based on baseline design values and other factors.

- Open HEEDS POST.

- On the *Ribbon*, select the **Curve Fit** plot tool.



- Create the plot and observe how the designs of the study compare to the target curve:



- Use the *Slider* arrows next to the **Design** field on the bottom of the plot to track various designs against the target. You can see that **Design 1077** is the best match to the target for this study.
- Animate the *Curve Fit* plot.

## References

[1] Mathews, J.H., "Bounded Population Growth: A Curve Fitting Lesson," *Mathematics and Computer Education*, Spring 1992, Volume 26, No. 2, pp. 169-176.



## Example 8: Designing a Pipe Network with Taguchi RPD

In this Taguchi Robust Parameter Design (RPD) problem, the effect that water flow rate, diameter of pipe, valve opening, and friction factors have on a pipe network is determined by HEEDS MDO. The diameter of pipe, the flow rate of water flowing through a pipe, and the amount a valve is opened are treated as control factors in this problem, while the friction factors are treated as noise factors. These noise and control factors will be used to help design a pipe network. This example shows you how to set up a project that utilizes the Taguchi RPD utility within HEEDS MDO.

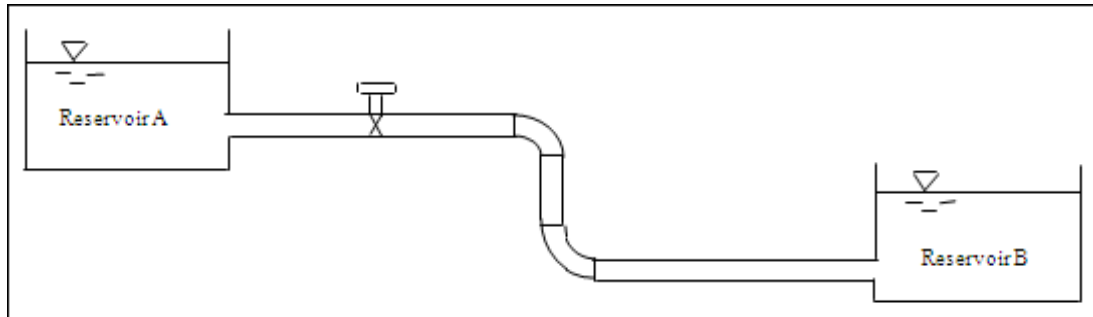


Fig. 1. Pipe network to be studied with Taguchi Robust Parameter Design (RPD).

### The problem

A pipe network is to be designed by determining the values for the pipe diameter, valve opening, and water flow rate through the pipe on water velocity and hydraulic head, such that the variations in the noise factors cause minimum variations in the responses (the hydraulic head and water velocity). The effect of these three control variables will be studied along with the noise variables present to the pipe network (friction factors).

#### Control Variables:

$D$	Diameter of the pipes.
$K\_Valve$	Loss coefficient caused by the valve (based upon how open the valve is).
$Q$	Flow rate of water flowing through the pipes.

#### Noise Variables:

$K\_entrance$	Loss coefficient nominal value caused by the entrance into the pipes from reservoir A.
$K\_valve\_variability$	Variability in the loss coefficient of the valve from the nominal value.
$K\_elbow$	Loss coefficient nominal value caused by the elbows in the pipe network.
$K\_exit$	Loss coefficient nominal value caused by the exit from the pipes into reservoir B.
$f\_variability$	Variability in the Friction factor based upon relative roughness and Reynolds Number.

#### Responses:

$H$	Hydraulic Head.
$V$	Water velocity.

### The baseline design

For this example, an executable named *Pipe\_Network.exe* was created to solve for  $H$ , and  $V$  using the following equations:

$$V = \frac{Q}{\pi \cdot \left(\frac{D}{2}\right)^2}$$

$$R_e = \frac{V \cdot D}{\nu}$$

$$f = \frac{1.325}{\left\{ \ln \left[ \left[ \frac{e}{3.7 \cdot D} \right] + \left[ \frac{5.74}{R_e^{0.9}} \right] \right] \right\}^2}$$

$$H = (K_{entrance} + K_{Valve} \cdot K_{valve\_variability} + 2 \cdot K_{elbow} + K_{exit}) \cdot \frac{V^2}{2 \cdot g} + f \cdot f_{variability} \cdot \frac{L}{D} \cdot \frac{V^2}{2 \cdot g}$$

Where:

- $\nu$  is the kinematic viscosity of water at 20 degrees Celsius ( $1.0 \cdot 10^{-6} \text{ m}^2/\text{s}$ ),
- $R_e$  is the Reynolds number of the flow,
- $e$  is the roughness of wrought iron tubing to be used for the pipes (0.046 mm),
- $f$  is the friction factor based upon the Reynolds number and relative roughness; predicted from a representative equation for the Moody Diagram [2],
- $L$  is the length of pipe in the network (50m).

Fig. 2 displays the pipe network in detail:

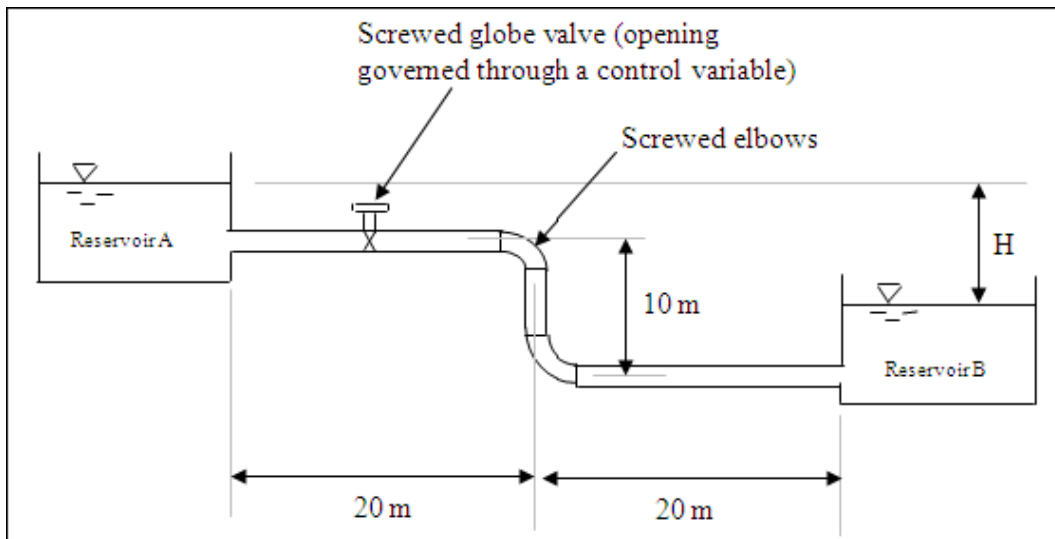


Fig. 2. Pipe network to be studied with Taguchi Robust Parameter Design (RPD). Fluid is water at 20 degrees Celsius.

The files for our example are found in the **Pipe\_Network** folder in the **Examples** directory.

The default path for Windows is: **C:\HEEDS\MDO\Version\Examples\Pipe\_Network**

The default path for Linux and UNIX is: **/opt/HEEDS/MDO/Version/Examples/Pipe\_Network**

If you installed HEEDS MDO in a different directory, look for the files in that directory instead. Three files are included for this example:

File Name	Description
<b>Network.in</b>	The input file
<b>Network.out</b>	The output file
<b>Network.exe</b> (Windows) <b>Network</b> (Linux)	The file that contains the analysis code

## Step 1: Starting a New Project

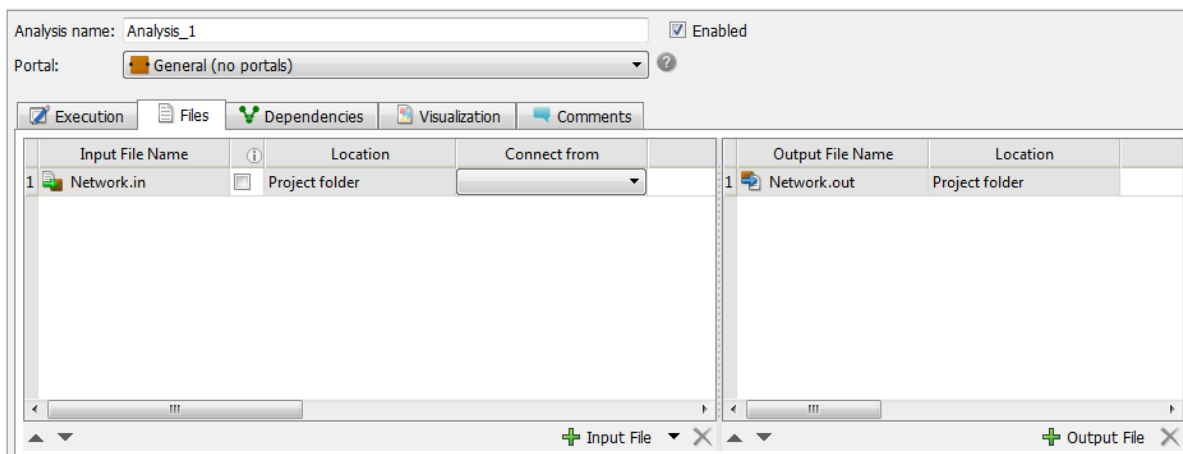
For this problem, we have one analysis. Our only task at the *Process* level is to define the default analysis that was created for us:

Start a new project and save it in your project directory as **PipeNetworkEx**. Your project is saved as **PipeNetworkEx.heeds**.

## Step 2: Defining the Process

We will define our *Process* using one analysis.

1. Select the **Process** tab. The *Process* page appears.
2. Click the **Analysis\_1** tool from the *Process View*. The *Analysis Manager* opens the **Execution** tab.
3. In the **Execution command** field, browse to **Network.exe** and add it to the analysis definition.
4. Click the **Files** tab and then click **+ Input File**. Add **Network.in**.
5. Click **+ Output File**, and then add **Network.out**.
6. Use defaults for the remaining fields. Your *Process* definition should look like this:



7. Save your project.

## Step 3: Defining the Parameters

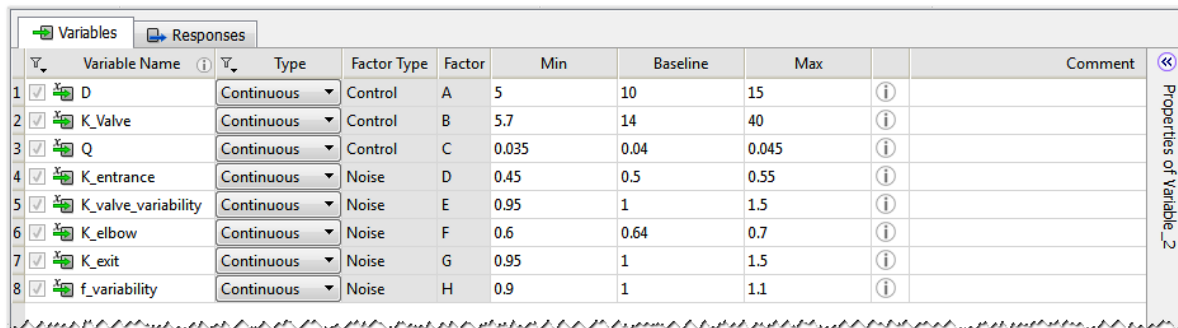
This example uses Taguchi DOE as the study method. As in a standard DOE study, only continuous and discrete variables can be used as factors, although you can have other types of variables in the project. In our project, all eight variables are continuous.

### Defining the project variables

1. Click the **Parameters** tab. The *Parameters* page opens to the *Variables Manager*. You can also define the parameters on the **Study** tab.
2. Add **eight** continuous variables to your project and define them as follows:

Variable Name	Type	Min	Baseline	Max
D	Continuous	5	10	15
K_Valve	Continuous	5.7	14	40
Q	Continuous	0.035	0.04	0.045
K_entrance	Continuous	.45	.5	.55
K_valve_variability	Continuous	.95	1.0	1.5
K_elbow	Continuous	.6	.64	.7
K_exit	Continuous	.95	1.0	1.5
f_variability	Continuous	.9	1.0	1.1

3. Use defaults for the remaining fields. Your variables should look like this:



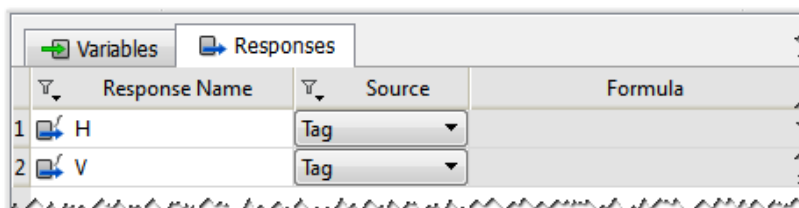
Variables									
	Variable Name	Type	Factor Type	Factor	Min	Baseline	Max		Comment
1	D	Continuous	Control	A	5	10	15		
2	K_Valve	Continuous	Control	B	5.7	14	40		
3	Q	Continuous	Control	C	0.035	0.04	0.045		
4	K_entrance	Continuous	Noise	D	0.45	0.5	0.55		
5	K_valve_variability	Continuous	Noise	E	0.95	1	1.5		
6	K_elbow	Continuous	Noise	F	0.6	0.64	0.7		
7	K_exit	Continuous	Noise	G	0.95	1	1.5		
8	f_variability	Continuous	Noise	H	0.9	1	1.1		

### Identifying the project responses

1. Add two responses to your project, and define them as follows:

Name	Source
H	Tag
V	Tag

Your completed *Responses* definition should look like this:



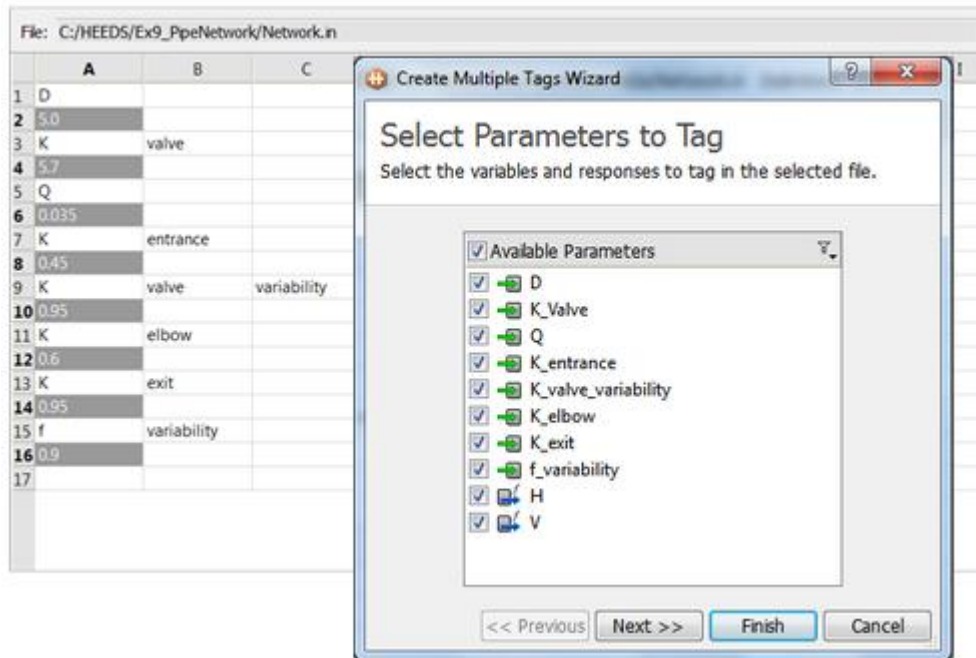
Responses		
	Response Name	Source
1	H	Tag
2	V	Tag

2. Save your project.

## Step 4: Tagging the Input and Output Files

### Tagging the input file

1. Select the **Tagging** tab. At the prompt in the *Process View*, open the dropdown and select **Network.in**. The parsed input file is displayed.
2. **Multi-Tag** all eight variables to the values you see below:



Your tagged variables should look like this:

File: C:/HEEDS/Ex9_PipeNetwork/Network.in			
	A	B	C
1	D		
2	D		
3	K	valve	
4	K_Valve		
5	Q		
6	Q		
7	K	entrance	
8	K_entrance		
9	K	valve	variability
10	K_valve_vari...		
11	K	elbow	
12	K_elbow		
13	K	exit	
14	K_exit		
15	f	variability	
16	f_variability		
17			

3. Save your project.

### Tagging the output file

1. On the *Ribbon*, open the **File** dropdown for **Response** and select **Network.out**. The parsed file is displayed.
2. Tag the responses **H** and **V** with the values at location **B1** and **B2**:

File: \\Pipe\_network\\Network.out

	A	B	C	D	E
1 H		416.849846			
2 V		17.825369			
3 Re		891268.434...	0.019694Pey...		

Your tagged output file should look like this:

File: \\Pipe\_network\\Network.out

	A	B	C	D	E
1 H		H			
2 V		V			
3 Re		891268.434...	0.019694		

3. Save your project.

### Step 5: Defining the Study

On the top part of the *Study Manager* we'll define our study type. On the lower tabbed section, we'll define the factors for our study.

1. Select the **Study** tab. On the *Study Manager*, locate **Study type** and select **DOE – Taguchi Robust Parameter Design** from the list box:

Study name: Study\_1

Process: Process\_1

Study type: DOE - Taguchi Robust Parameter Design

Surrogates: All

Saved Designs

Success designs: All designs

Error designs: Rename

Run Options

☒ Save restart data after each evaluation

☐ Do not stop HEEDS for a design-based error

Random seed: 0.1

☒ More options

2. Just below the **Study type**, click the **Methods** tab. The *Methods View* lists all of the variables in the project.

3. For this study, we want to identify all of the listed variables as factors. To do this, open the list box next to each variable, and then select a **Factor Type** as indicated below. HEEDS automatically assigns a name, **A**, **B**, **C**, etc., to each factor:

Variable	Factor Type	Fact	Levels	Baseline	Mode	Low
<input checked="" type="checkbox"/> D	Control	A	3	10	Value	5
<input checked="" type="checkbox"/> K_Valve	Control	B	3	14	Value	5.7
<input checked="" type="checkbox"/> Q	Control	C	3	0.04	Value	0.035
<input checked="" type="checkbox"/> K_entrance	Noise	D	3	0.5	Value	0.45
<input checked="" type="checkbox"/> K_valve_variability	Noise	E	3	1	Value	0.95
<input checked="" type="checkbox"/> K_elbow	Noise	F	3	0.64	Value	0.6
<input checked="" type="checkbox"/> K_exit	Noise	G	3	1	Value	0.95
<input checked="" type="checkbox"/> f_variability	Noise	H	3	1	Value	0.9

4. After you identify the factors, move to the panel on the right and define the properties for the **Control Sampling Method** and **Noise Sampling Method** as indicated below. As you enter the information, the values in the table update automatically. Your completed *Study* definition should look like this:

Variable	Factor Type	Fact	Levels	Baseline	Mode	Low
<input checked="" type="checkbox"/> D	Control	A	3	10	Value	5
<input checked="" type="checkbox"/> K_Valve	Control	B	3	14	Value	5.7
<input checked="" type="checkbox"/> Q	Control	C	3	0.04	Value	0.035
<input checked="" type="checkbox"/> K_entrance	Noise	D	3	0.5	Value	0.45
<input checked="" type="checkbox"/> K_valve_var...	Noise	E	3	1	Value	0.95
<input checked="" type="checkbox"/> K_elbow	Noise	F	3	0.64	Value	0.6
<input checked="" type="checkbox"/> K_exit	Noise	G	3	1	Value	0.95
<input checked="" type="checkbox"/> f_variability	Noise	H	3	1	Value	0.9

Control Sampling Method	
Number of factors:	3
Number of evaluations:	18
<b>2 Level</b> <input type="radio"/> Full factorial 8 <input type="radio"/> Partial factorial 4 <input type="radio"/> Taguchi 4	
<b>3 Level</b> <input type="radio"/> Full factorial 27 <input type="radio"/> Partial factorial 9 <input type="radio"/> Taguchi 9,27	
<b>Mixed Level</b> <input checked="" type="radio"/> Taguchi 18	

Noise Sampling Method	
Number of factors:	5
Number of evaluations:	18
<b>2 Level</b> <input type="radio"/> Full factorial 32 <input type="radio"/> Partial factorial 8,16 <input type="radio"/> Taguchi 8	
<b>3 Level</b> <input type="radio"/> Full factorial 243 <input type="radio"/> Partial factorial 27,81 <input type="radio"/> Taguchi 27,81	
<b>Mixed Level</b> <input checked="" type="radio"/> Taguchi 18,36	

Total evaluations: 324

5. Your *Methods* definition is complete. Now select the **Responses** tab. Your project responses **H** and **V** are listed here.
6. We would like both the hydraulic head (**H**) and the water velocity (**V**) to be as large as possible for this problem, meaning faster movement of water and greater ability for movement of water. For **H** and **V**, locate the **Goal** list box, and select **Larger is better**:

Available Responses	Type	Analysis	Goal	Value
<input checked="" type="checkbox"/> H	Tag	Analysis_1	Larger is better	
<input checked="" type="checkbox"/> V	Tag	Analysis_1	Larger is better	

**Note:** Other goal options include: *Smaller is better*, *Target is best*, *Nominal is best*, and *No goal*.

7. Save your project.

## Step 6: Running the Study

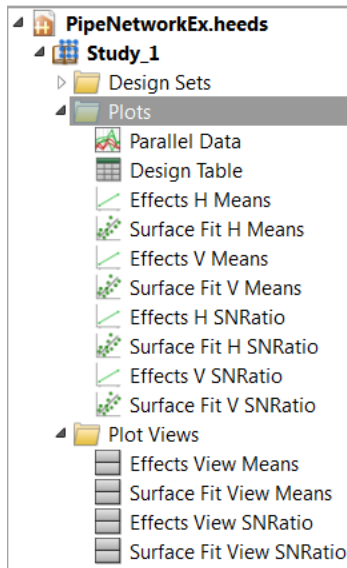
On the *Run* page, click **Run**.

## Step 7: Viewing the Results

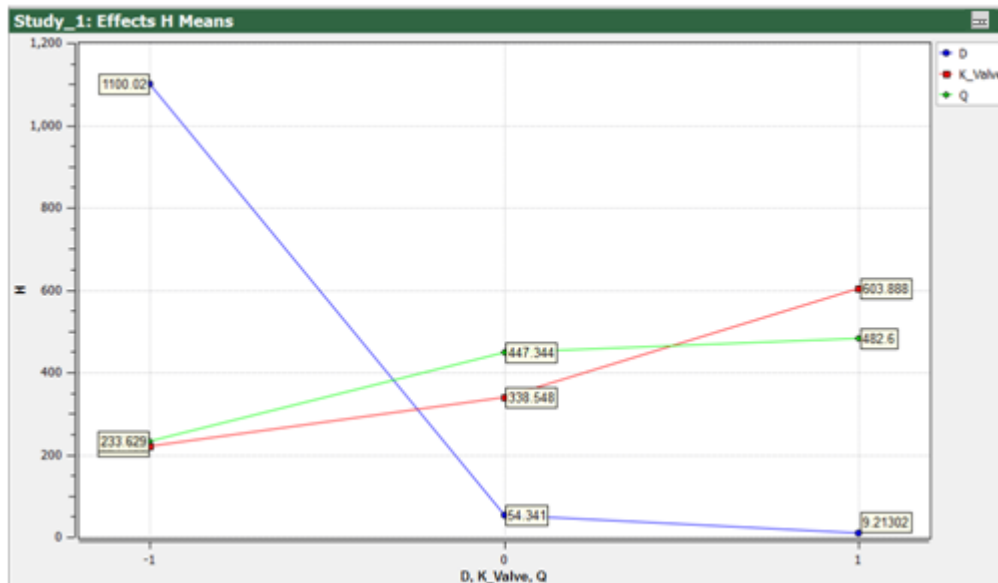
The results for Taguchi RPD analyses are presented in the same manner as traditional DOE methods. For this example, we will focus on identifying an optimal design using the Taguchi RPD method.

1. On the *Ribbon*, click **Open HEEDS|post**.

Several plot options are available for you to review your design evaluations:



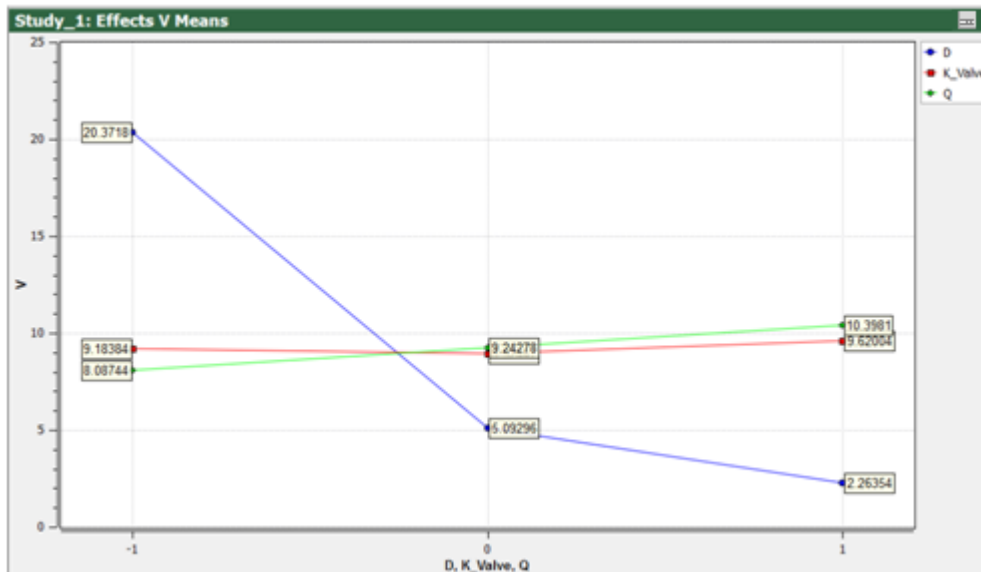
Let's look at the plot, **Effects H Means**:



Considering the response **H**, one can infer that the following is true:

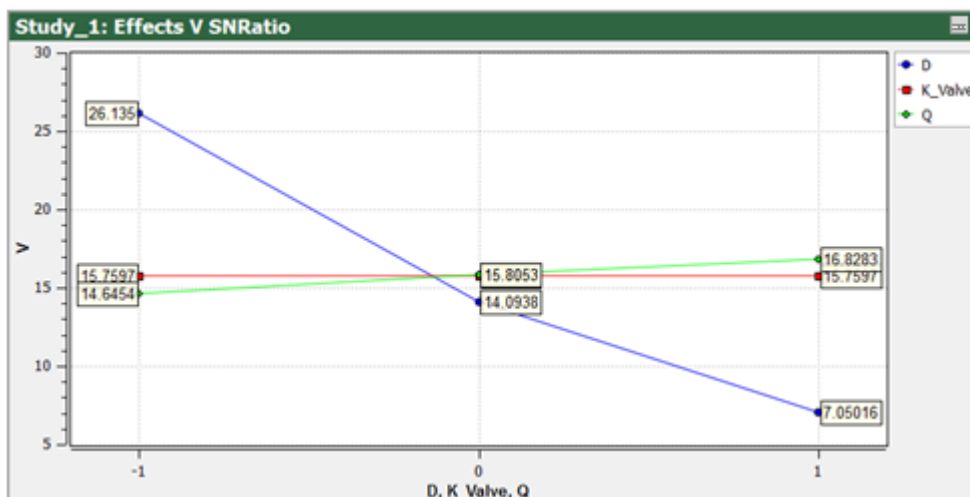
- Largest **H Means** corresponds to smaller **D**.

- Largest **H Means** corresponds to larger **K\_Valve**.
  - Largest **H Means** corresponds to larger **Q**.
2. Click **Effects V Means**. The plot appears as follows:



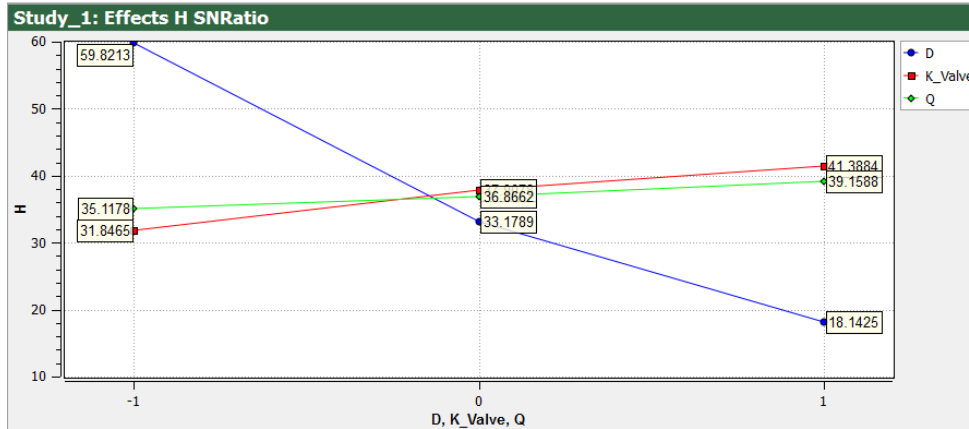
Considering the response **V**, one can infer that the following is true:

- Largest **V Means** corresponds to smaller **D**.
  - Largest **V Means** is mostly independent of **K\_Valve**.
  - Largest **V Means** corresponds to larger **Q**.
3. Click **Effects V SNRatio**. This plot takes the noise factors into account:



The response **V** has a similar trend for **SNR**, where smaller **D** and larger **Q** are desired for larger **SNR(V)**. The factor **K\_Valve** has no effect on **SNR(V)**.

4. Click **Effects H SNRatio**. The plot appears:



Again, the response **H** has a similar trend for **SNR**, where smaller **D** and larger **Q** are desired for larger **SNR(V)**. Now, the factor **K\_Valve** should be higher for larger **SNR(V)**.

Since the **Effects Means** and **Effects SNRatio** for both responses are very similar, the following conclusions can be made from the analysis about the optimal design for the pipe network:

- **D** should be smaller.
- **K\_Valve** should be larger.
- **Q** should be larger.

## Step 8: Identifying the Best Design

Since the goal for both the responses is **Larger is better**, the objective is to identify the control factor settings that maximize the SNR for both the responses.

Based on the results, the best settings for the control factors for **H** should be:

**D** = 5 (the lowest value)  
**K\_Valve** = 40 (the largest value)  
**Q** = 0.045 (the largest value)

Based on the results, the best settings for the control factors for **V** should be:

**D** = 5 (the lowest value)  
**K\_Valve** = any value (almost no change in the SNR)  
**Q** = 0.045 (the largest value)

Based on the settings above, the best design for both the responses is:

**D** = 5 (the lowest value)  
**K\_Valve** = 40 (the largest value)  
**Q** = 0.045 (the largest value)

## References

- [1] Potter, M.C. and Wiggert, D.C., *Mechanics of Fluids*, Third Edition, 2002, pp. 318.
- [2] <http://www.lmnoeng.com/moody.htm>, 1998-2001 LMNO Engineering, Research, and Software, Ltd.

## **Siemens Industry Software**

### **Headquarters**

Granite Park One 5800  
Granite Parkway  
Suite 600  
Plano, TX 75024  
USA  
+1 972 987 3000

### **Americas**

Granite Park One  
5800 Granite Parkway  
Suite 600  
Plano, TX 75024  
USA  
+1 314 264 8499

### **Europe**

Stephenson House  
Sir William Siemens Square  
Frimley, Camberley  
Surrey, GU16 8QD  
+44 (0) 1276 413200

### **Asia-Pacific**

Suites 4301-4302, 43/F  
AIA Kowloon Tower, Landmark East  
100 How Ming Street  
Kwun Tong, Kowloon  
Hong Kong  
+852 2230 3308

## **About Siemens PLM Software**

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit [www.siemens.com/plm](http://www.siemens.com/plm).

© 2019 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Geolus, GO PLM, I-deas, Insight, JT, NX, Parasolid, Solid Edge, Teamcenter, Tecnomatix and Velocity Series are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other trademarks, registered trademarks or service marks belong to their respective holders.