Altair

HyperWorks

Altair HyperStudy 2019 Tutorials

HS-1040: Minimization of Internal Rosenbrock Function

altairhyperworks.com

HS-1040: Minimization of Internal Rosenbrock Function

In this tutorial you will learn how to register a Compose/OML or python function in HyperStudy using the Preference file (`.mvw`), and then use the registered function for output response evaluation in the study. The Rosenbrock function is defined as a python or OML script and registered in HyperStudy.

The example defines two input variables labeled x and y, respectively. The objective of the optimization is to minimize f(x,y)= 100*(y-x^2)^2 + (1-x)^2. The range for x and y is set to [-2 ; -2] , and the start point is [-1 ; -1].

Step 1: Define the Rosenbrock Function

Define the Rosenbrock Function with:

Compose:

1. Start Compose.
2. From the menu bar, click ***File*** > ***New*** > ***OML File***.
3. In the editor, enter the following OML commands:

```
function f = ros_eval(x,y)

    f = 100.0*(y-x^2)^2 + (1-x)^2
```

4. Form the menu bar, click ***File*** > ***Save As***.
5. In the **Save As** dialog, navigate to your working directory and save the file as `rosenbrock_function.oml`.
6. Quit Compose by clicking ***File*** > ***Exit*** from the menu bar.

Tip: Avoid the subsequent preference file steps when using the **Register Function** … feature in Compose with OML functions. Highlight the function name, then right click to access this feature on the context menu.

Or

Python:

1. In your favorite text editor, enter the following python commands:

```
def ros_eval(x, y):
    return 100*(y-x*x)*(y-x*x) + (1-x)*(1-x)
```

2. Form the menu bar, click ***File*** > ***Save As***.
3. In the **Save As** dialog, navigate to your working directory and save the file as `rosenbrock_function.py`.

Step 2: Add the Function to the Preference File

Add the OML function to the Preference file:

1. In a text editor, enter the following OML registration function:

```
   *Id("HyperGraph v11.0")
*BeginDefaults()
*BeginPlotDefaults()
*SetOMLRootDir("C:/Program Files/Altair/2019/Compose2019")
*RegisterOMLFunction("ros_eval",”<path>/rosenbrock_function.oml",2)
*EndPlotDefaults()
*EndDefaults()
```

Note: Replace `<path>` with the actual location of the file, and modify the *SetOMLRootDir statement to point to the appropriate location.

2. Save the file as a preference file with the name `rosenbrock_prefs.mvw` file.
3. Close the text editor.

or

Add the python function to a Preference file:

1. In a text editor, enter the following pyton registration function:

```
*Id("HyperStudy v14.0")
   *BeginDefaults()
      *BeginPlotDefaults()
      *RegisterPythonFunction("ros_eval","<path>/rosenbrock_function.py",
      2)
   *EndPlotDefaults()
*EndDefaults()
```

2. Replace `<path>` with the actual location of the .py file.
3. Save the file as a preference file with the name `rosenbrock_prefs.mvw` file.
4. Close the text editor.

Step 3: Perform the Study Setup

1. Start HyperStudy.

2. From the menu bar, click ***File*** > ***Use Preferences File***.
3. In the **HyperStudy - Set Preference File** dialog, open the `rosenbrock_prefs.mvw` file.
4. To start a new study, click ***File*** > ***New*** from the menu bar, or click on the toolbar.
5. In the **HyperStudy – Add** dialog, enter a study name, select a location for the study, and click ***OK***.
6. Go to the **Define Models** step.
7. Add an Internal Math model.
   a. Click ***Add Model***.
   b. In the **Add - HyperStudy** dialog, select **Internal Math** and click ***OK***.
8. Go to the **Define Input Variables** step.
9. Add input variables.
   a. Click ***Add Input Variable*** twice.
   b. In the work area, label the input variables X and Y.
   c. Change both input variable's lower, initial and upper bounds to the values indicated in the image below.

	Active	Label	Varname	Lower Bound	Nominal	Upper Bound
1	☑	X	var_1	-2.0000000 ...	-1.0000000 ...	2.0000000 ...
2	☑	Y	var_2	-2.0000000 ...	-1.0000000 ...	2.0000000 ...

10. Go to the **Specifications** step.

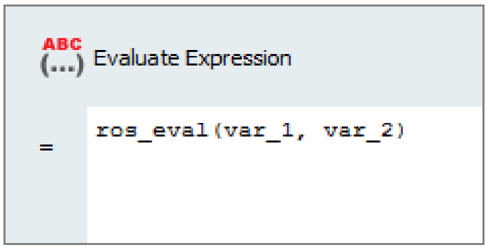
Step 4: Perform the Nominal Run

1. In the work area, set the **Mode** to ***Nominal Run***.
2. Click ***Apply***.
3. Go to the **Evaluate** step.
4. Click ***Evaluate Tasks***. An `approach/nom_1/` directory is created inside the study directory.
5. Go to the **Define Output Responses** step.

Step 5: Create and Define Output Responses

1. Click ***Add Output Response***.
2. In the **Expression** column of **Response 1**, click ...

3. In the **Expression Builder**, click the ***Functions*** tab.
4. From the list of available functions, select ***ros_eval***.
5. Click ***Insert Varname***. The function `ros_eval()` appears in the **Evaluate Expression** field.
6. Click the ***Input Variables*** tab.
7. In the work area, select the input variables ***X*** and ***Y***.
8. Click ***Insert Varname***. The input variables appear in the expression as `ros_eval(var_1, var_2)`.



9. Click ***Evaluate Expression.*** The expression `ros_eval(var_1, var_2)` changes to `404`.
10. Click ***OK.*** This completes the study setup.

Step 6: Run an Optimization Study

1. In the **Explorer**, right-click and select ***Add*** from the context menu.
2. In the **Add - HyperStudy** dialog, select ***Optimization*** and click ***OK***.
3. Go to the **Select Output Responses** step.
4. Add an objective to Response 1.
   a. In the **Objectives** column of Response 1, click .
   b. In the pop-up window, set **Type** to ***Minimize*** and click ***OK***.

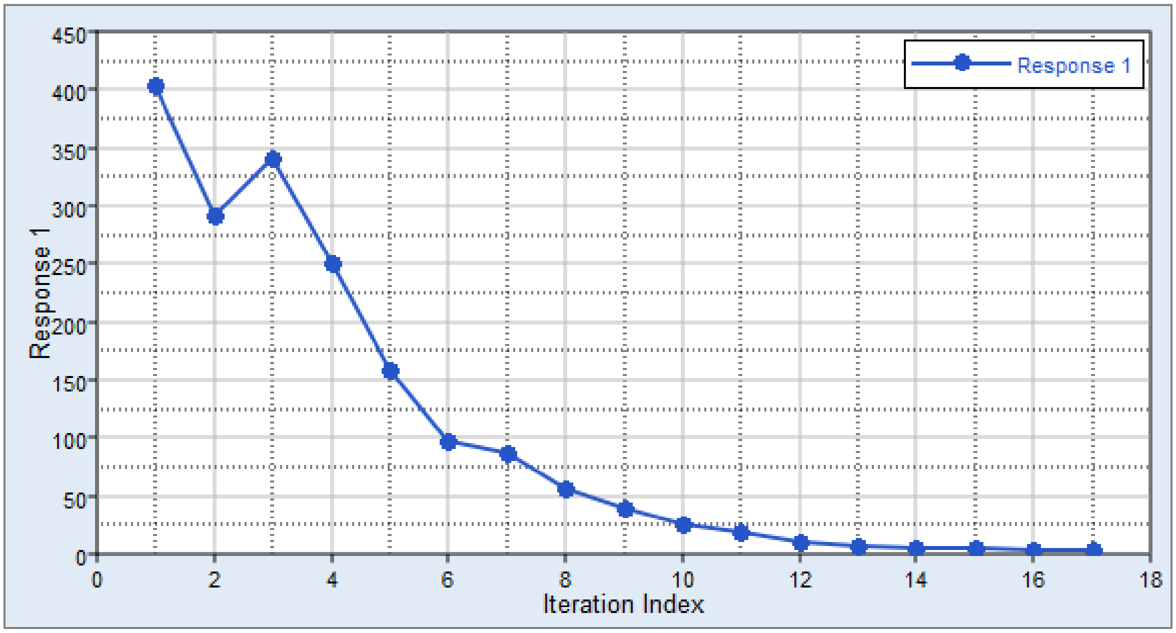
	Active	Label	Varname	Objectives	Constraints	Evaluate From	Expression
1	☑	Response 1	r_1	Minimize ...		Solver	ros_eval(var_1, var_2)

5. Click ***Apply***.
6. Go to the **Specifications** step.
7. In the work area, set the **Mode** to ***Adaptive Response Surface Method (ARSM)***.

   Note: Only the methods that are valid for the problem formulation are enabled.
8. Click ***Apply***.
9. Go to the **Evaluate** step.
10. Click ***Evaluate Tasks***.

11. **Optional**. Click the ***Iteration Plot*** tab to monitor the progress of the optimization.

    The iteration history shows a significant reduction in the objective value. The Rosenbrock function has a global minimum that is difficult for any optimizer to find due to its flatness in the area of the true optimum, and ARSM has not found the theoretical solution at (x,y)=(1,1).



Last modified: v2017.2 (12.1156684)