Minjung Ryu, Matti Hietala, Jarkko Mansikka-aho, Jaana Tommiska, Tony Garcia, Lauri Mäenpää, Antti Roine

September 6, 2023

95. Data Module

95.1. Introduction

The HSC Data module is made for visualizing, analyzing, and transforming raw data, as well as converting this data into models. The new HSC Data module features make it possible to train a neural network model which could predict unknown results based on the experimental data. There are two types of neural networks that can be trained by this module: FFNN (Feedforward Neural Network) and RNN (Random Neural Network). Additionally, Charts and PSO (Particle Swarm Optimizer) options are provided.

95.2. Basic use

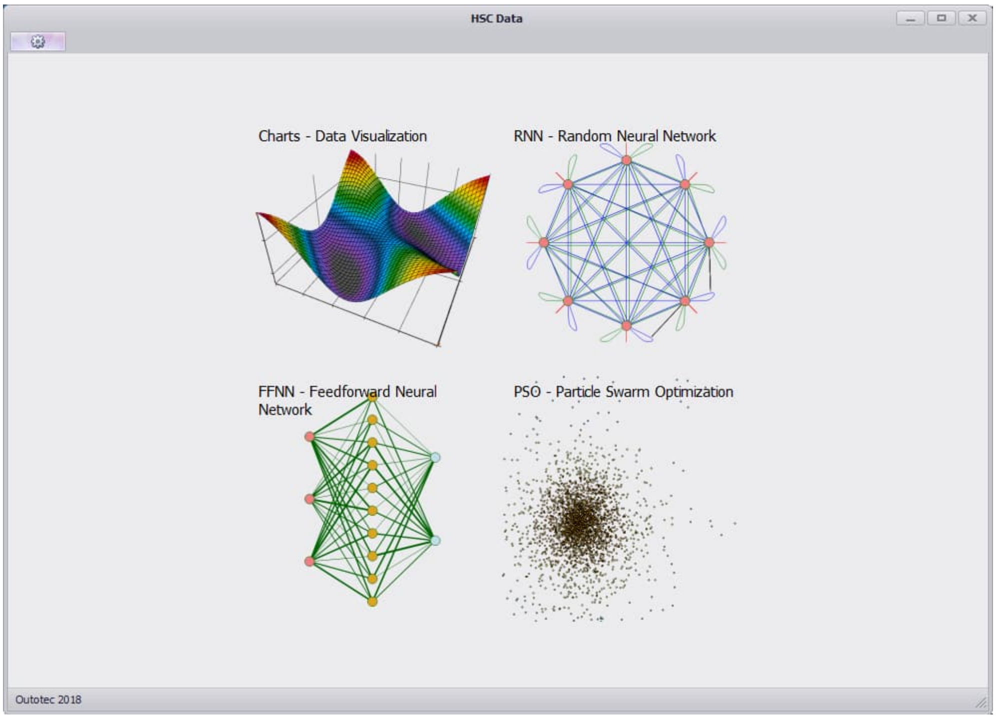

Fig. 1. Data processing module

To start, click the button (top left) and select the menu shown in **Fig. 1**.

1. Charts – Draw a chart
2. FFNN (Feedforward Neural Network) – Train a simple artificial neural network without feedback loops.
3. RNN (Random Neural Network) – Train an artificial neural network with one or more feedback loops.
4. PSO (Particle Swarm Optimization) – Search for a solution of an optimization problem.

Each tool must be closed by clicking the 'Close' button when you need to use another tool. For example, if you are using the Charts tool, then the menu buttons for other

Minjung Ryu, Matti Hietala, Jarkko Mansikka-aho, Jaana Tommiska, Tony Garcia, Lauri Mäenpää, Antti Roine

tools are not available, as shown in **Fig. 2,** until the Charts tool is closed. Once you close the Charts tool, all menus will be enabled again as shown in **Fig. 1**.

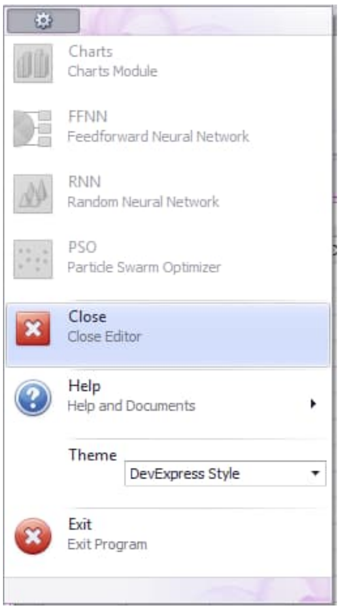

Fig. 2. Selecting other menus is disabled when the user is already using a tool. Click ‘Close’ to return to the main view.

95.3. Charts

Charts can be plotted using this tool.

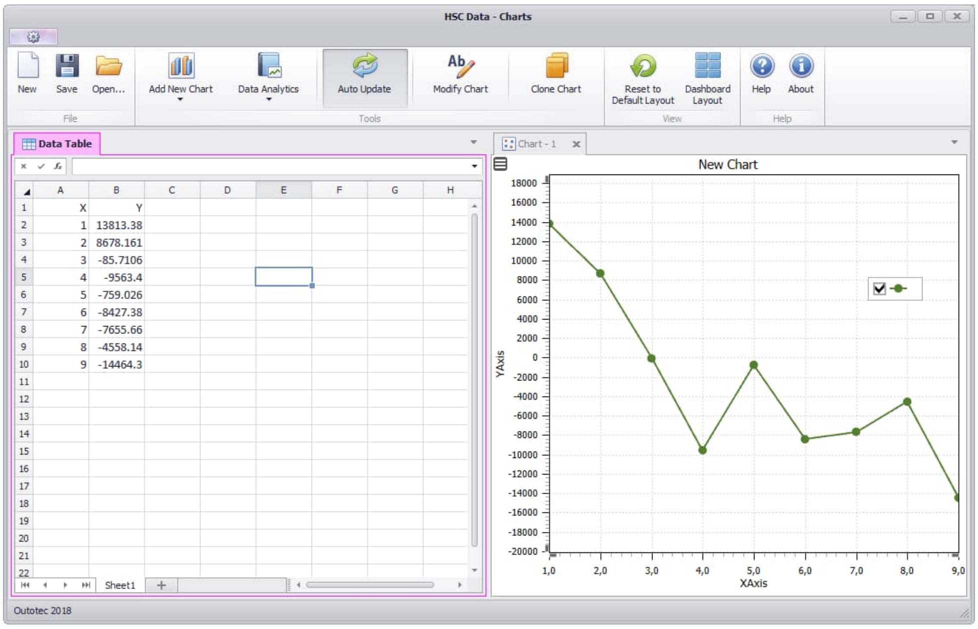

Fig. 3. Chart tool

1. **Import data**
   Data can be imported by copy-and-paste to an empty table or by opening an .HSCChart file.

Minjung Ryu, Matti Hietala, Jarkko Mansikka-aho, Jaana Tommiska, Tony Garcia, Lauri Mäenpää, Antti Roine

2. **Create a chart**

   Click ‘Add New Chart’ to create a new chart.

95.4. FFNN (Feedforward Neural Network)

Simple feedforward neural network models can be trained using this tool.

1. **Import raw data**

   Data can be imported by Copy-and-Paste to an empty table or by opening an .xlsNN file with data. NB! The data must start from Row 4.

   Please note that the .xlsNN file format is the same as the Excel xlsx file format. This also makes it possible to edit the raw data in Excel, but please do not change the format and layout in Excel.

2. **Specify variables**

   Variable types X, Y, YNN, and RULE can be specified on the first row of the data table. X columns are for inputs, of which there can be as many as the number of input variables, and Y column is for the actual output. YNN can remain empty until the model is trained and ready to predict the values. Marking ‘E’ or ‘e’ on the RULE column is used to indicate the data rows that will be excluded during the training phase.

   On Row 2 and Row 3, the name of variables or additional information (e.g., measure units) can be written.

A14

	A	B	C	D	E	F	G	H	I	J	K
1	RULE	X	X	X	X	X	X	X	X	Y	YNN
2		Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6	Variable 7	Variable 8	Output	Prediction
3											
4		-814,2718	789,95574	-909,6783	-619,8832	181,66895	217,46993	-776,2931	-134,2362	-8737,705	
5	E	310,72289	184,31778	-378,2826	867,05577	697,7672	446,1703	607,06593	389,1524	16541,272	
6		-253,5516	-475,1565	-673,2303	808,08084	593,94161	823,2823	36,886301	512,00009	12272,375	
7		-601,5752	552,85545	-77,71463	-53,34304	-489,0354	-342,2334	24,225353	-736,8163	-10165,91	
8	E	529,05619	-37,94643	-230,2115	-912,2926	-714,1582	546,07828	-286,5546	453,37864	-2559,816	
9		-136,8372	66,442694	-6,061481	291,32099	-316,8077	187,63231	-319,0755	109,59526	-671,8633	
10		878,65056	-467,4567	711,63312	-584,0268	-987,7586	-848,6445	-424,3272	-490,2518	-17180,44	
11		-273,0786	-526,5216	-598,1683	-802,0934	317,06132	-497,5222	177,29824	-12,81387	-6590,251	

Fig. 4. Variable assignment for FFNN

3. **Set the model properties**

   Hyperparameters are set to specify the structure of the neural network model. To see a detailed explanation, click the name of the parameter.

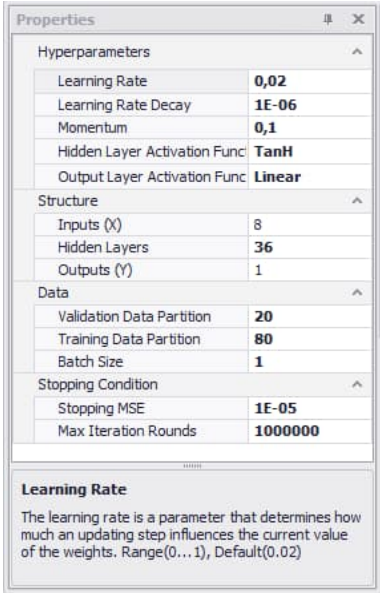

Fig. 5. Hyperparameters setting for FFNN

Click File – Neural Network – 'View NN Picture' to visualize the structure of the neural network.

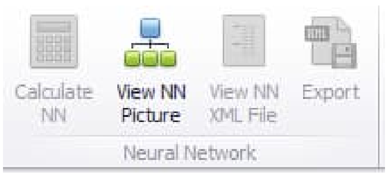

Fig. 6. Neural Network menu for FFNN

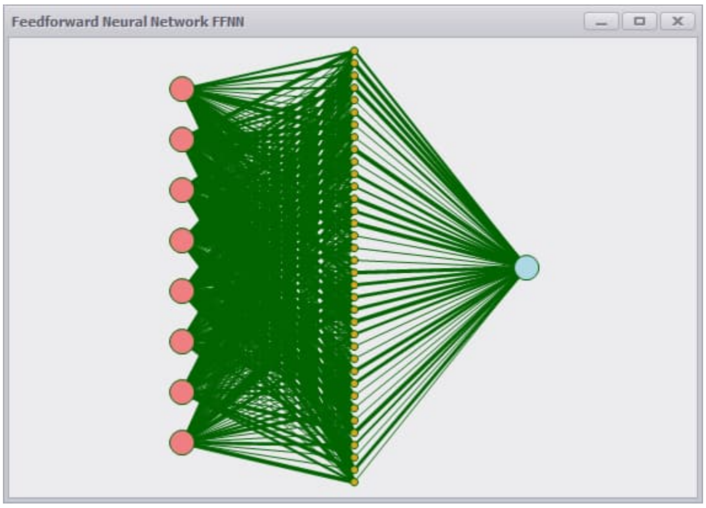

Fig. 7. Visualization of the FFNN model structure

4. **Apply data**
   Click the button File – Training – 'Apply Data' under File to check the validity of the model specification.

Fig. 8. Training menu for FFNN

5. **Train the model**

   Click the button File – Training – ‘Learn’ to start training. If you want to stop the training before it satisfies the stopping condition, click the ‘Stop’ button. Otherwise it will continue learning.

6. **Review the result**
   1) <u>Parity chart</u>: The parity chart compares the prediction against the given output of training data. The X-axis is for the values that a user has given as a Y variable, and the Y-axis is for the values that the neural network has calculated. The values on the plot are scaled between -1 and 1. The points lie on the gray line when the predictions are close to the original value.
   2) <u>Compare values</u>: In the compare values chart, the axes are the same but the values are arranged in ascending order.
   3) <u>Performance MSE</u>: This shows the Mean Squared Error at each iteration. The dashed line indicates the lowest error value.

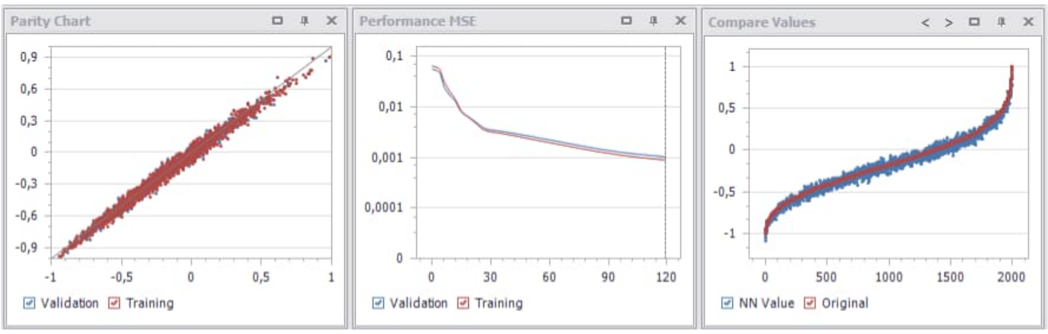

Fig. 9. Training result of FFNN

7. **Apply the model**

   Click ‘Calculate NN’ to apply the trained neural network to new inputs. Predicted outputs are calculated and shown briefly on the YNN column.

8. **Export the model**

   Click the button File – Neural Network – ‘View XML File’ to see the model. This file contains all the information about the model.

   Click the button File – Neural Network – ‘Export’ to save the XML file. This file can be used as a model in the HSC Sim module later.

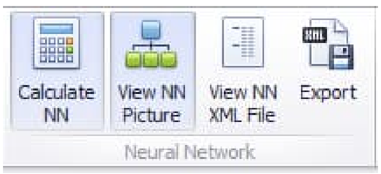

Fig. 10. Export button for FFNN

Neural Network XML file


```
<ANN z:Id="1"
xmlns="http://schemas.datacontract.org/2004/07/Outotec.HSC.NeuralNetwork"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns:z="http://schemas.microsoft.com/2003/10/Serialization/"><Layers z:Id="2"
z:Size="2"><ANN.Layer z:Id="3"><Neurons z:Id="4" z:Size="36"><ANN.Neuron
z:Id="5"><Weights z:Id="6" z:Size="9"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"><a:doubl
e>0.087863238078767184</a:double><a:double>0.10051625202477402</a:double
><a:double>-
0.14024925410291</a:double><a:double>0.053815296469839229</a:double><a:d
ouble>-0.057377539863653106</a:double><a:double>-
0.14372930139161377</a:double><a:double>-
0.15549887696409495</a:double><a:double>0.21509732237125964</a:double><a
:double>-0.093407151064921229</a:double></Weights><DeltaWeights z:Id="7"
z:Size="9"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"><a:doubl
e>0.0004656373328349088</a:double><a:double>1.3570674647441076E-
05</a:double><a:double>-2.2335438011419661E-05</a:double><a:double>-
0.00053280778874902155</a:double><a:double>-
```


Fig. 11. Generated XML file example

95.5. RNN (Random Neural Network)

Neural networks with loops can be trained using this tool.

1. **Import raw data**

   Data can be imported by Copy-and-Paste to the empty table or by opening an .xlsNN file with raw data. NB! The data must start from Row 4.

   Please note that the .xlsNN file format is the same as the Excel xlsx file format. This also makes it possible to edit the raw data in Excel, but please do not change the format and layout in Excel.

2. **Specify variables**

   Variable types X, Y, YNN, and RULE can be specified on the first row of the data table. The X columns are for inputs of which there can be as many as the number of data inputs, and the Y column is for the actual output. YNN can remain empty until the model is trained and ready to predict the values. Marking ‘E’ or ‘e’ on the RULE column is used to indicate the data rows that will be excluded during the training phase.

   On Row 2 and Row 3, the name of variables or additional information (e.g. measure units) can be written.

L13

	A	B	C	D	E	F	G	H	I
1	RULE	X	X	X	X			Y	YNN
2									
3									
4		5.1	3.5	1.4	0.2	Iris-setosa		1	1.000003
5		4.9	3	1.4	0.2	Iris-setosa		1	0.99999
6	E	4.7	3.2	1.3	0.2	Iris-setosa		1	0.999994
7	E	4.6	3.1	1.5	0.2	Iris-setosa		1	0.999991
8		5	3.6	1.4	0.2	Iris-setosa		1	1.000004
9		5.4	3.9	1.7	0.4	Iris-setosa		1	1.000012
10		4.6	3.4	1.4	0.3	Iris-setosa		1	0.999999

Fig. 12. Variable assignment for RNN

Minjung Ryu, Matti Hietala, Jarkko Mansikka-aho, Jaana Tommiska, Tony Garcia, Lauri Mäenpää, Antti Roine

3. **Set the model properties**

Hyperparameters are set to specify the structure of the neural network model. To see a detailed explanation, click the name of the parameter.

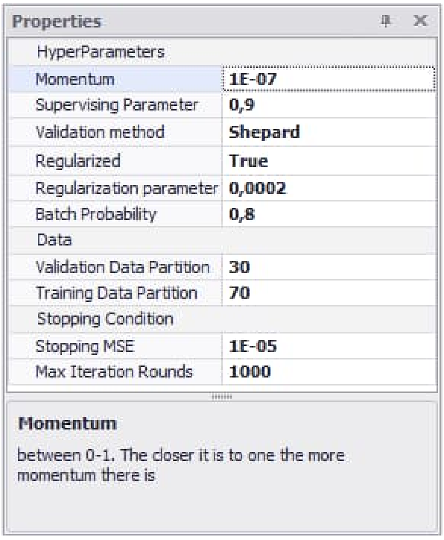

Fig. 13. Parameters setting for RNN

Click File – Neural Network – 'View RNN Picture' to visualize the structure of the neural network. Neurons with a red line are inputs and a neuron that has a black line is an output.

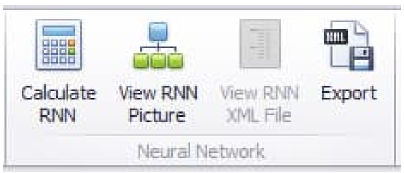

Fig. 14. Neural Network menu for RNN

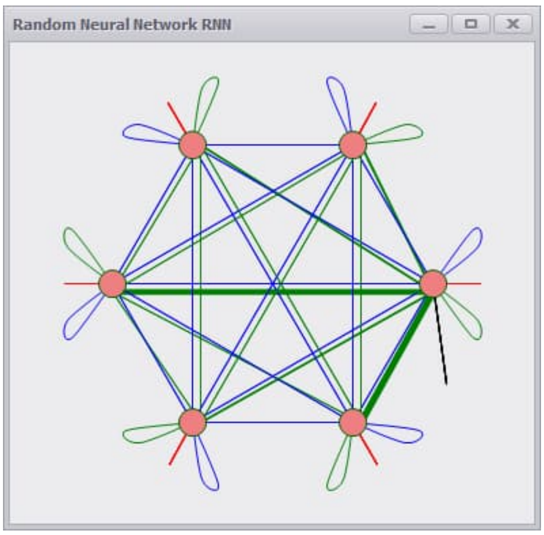

Fig. 15. Visualization of the RNN model structure

Minjung Ryu, Matti Hietala, Jarkko Mansikka-aho, Jaana Tommiska, Tony Garcia, Lauri Mäenpää, Antti Roine

4. **Apply data**

   Click the button File – Training – ‘Apply Data’ under File to check the validity of the model specification.

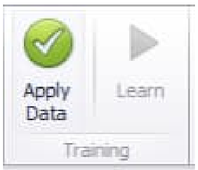

Fig. 16. Training menu for RNN

5. **Train the model**

   Click the button File – Training – ‘Learn’ to start training. If you want to stop the training before it satisfies the stopping condition, click the ‘Abort’ button next to the progress bar.

Fig. 17. Training progress bar for RNN

6. **Review the result**

   1) <u>Parity chart</u>: The parity chart compares the prediction against the given output of training data. The X-axis is for the values that a user has given as the Y variable, and the Y-axis is for the values that the neural network has calculated. The values on the plot are scaled between -1 and 1. The points lie on the gray line when the prediction is close to the original value.
   2) <u>Compare values</u>: In the compare values chart, the axes are the same but the values are arranged in ascending order.
   3) <u>Performance MSE</u>: This shows the Mean Squared Error at each iteration. The dashed line indicates the lowest error value.

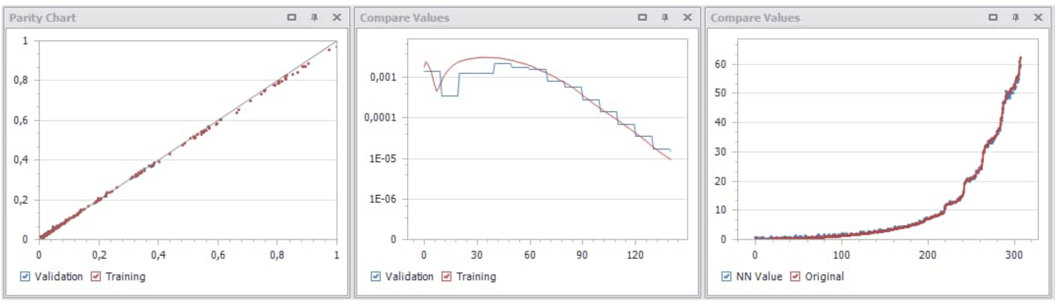

Fig. 18. Training result of RNN

7. **Apply the model**

   Click ‘Calculate RNN’ to apply the trained random neural network to new inputs. Predicted outputs are calculated and shown on the YNN column.

8. **Export the model**

Click the button File – Neural Network – ‘View RNN XML File’ to see the model. This file contains all the information about the model.

Click the button File – Neural Network – ‘Export’ to save the XML file. This file can be used as a model in the HSC Sim module later.

Fig. 19. Export button for RNN

95.6. PSO (Particle Swarm Optimization)

PSO searches for a solution for optimization problems.

1. **Specify variables and a function**
   Variables can be set by clicking the ‘Set X’ button or by adding the comment ‘PSO_X//’. If there are constraints on the range of variables, the comment should include them. For example, the comment PSO_X/-10/10 means that the search domain for variable X is bigger or equal to -10 and smaller or equal to 10 ($-10 \leqslant X \leqslant 10$). Search domains are stated in each field of the variables.
   Once the variables are specified, the objective function that will be minimized over the variables is specified. Use one or more fields for this in the same way as variable assignments. This can be done by clicking ‘Set M’ or by adding the comment ‘PSO_M’. For the value of this field, the formula function can be written using usual Excel functions, such as ABS, SIN, or EXP.

Fig. 20. PSO menu

Minjung Ryu, Matti Hietala, Jarkko Mansikka-aho, Jaana Tommiska, Tony Garcia, Lauri Mäenpää, Antti Roine

September 6, 2023

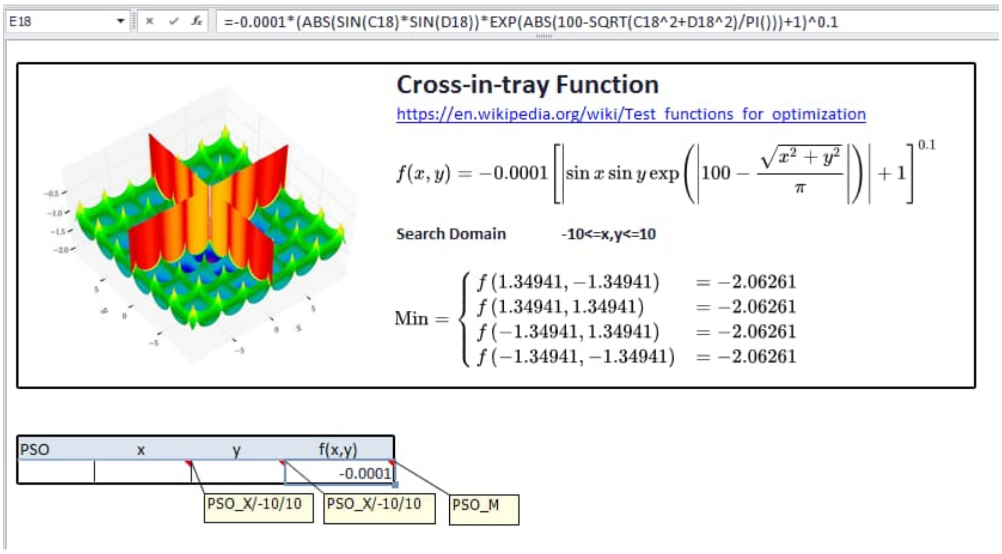

Fig. 21. Variable assignment for PSO

2. **Set the properties**
   Parameters can be tuned to yield better performance. For a detailed explanation, click the name of the parameter.

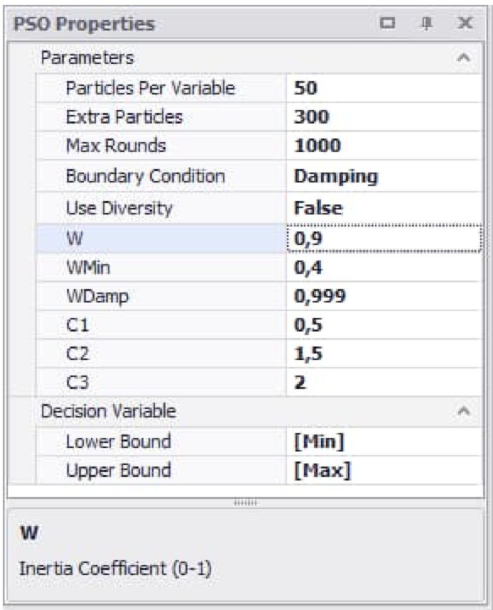

Fig. 22. Parameter settings for PSO

3. **Optimize**
   Click the button File – PSO – 'Optimize' to start computing. Press 'Stop' to terminate the calculation.

4. **Review the result**
   The performance graph shows the solution value at each iteration.

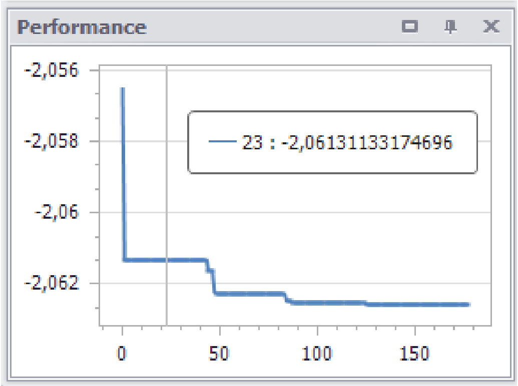

Fig. 23. Performance review for PSO

Also, all fields that were empty before optimization are now filled with a result.

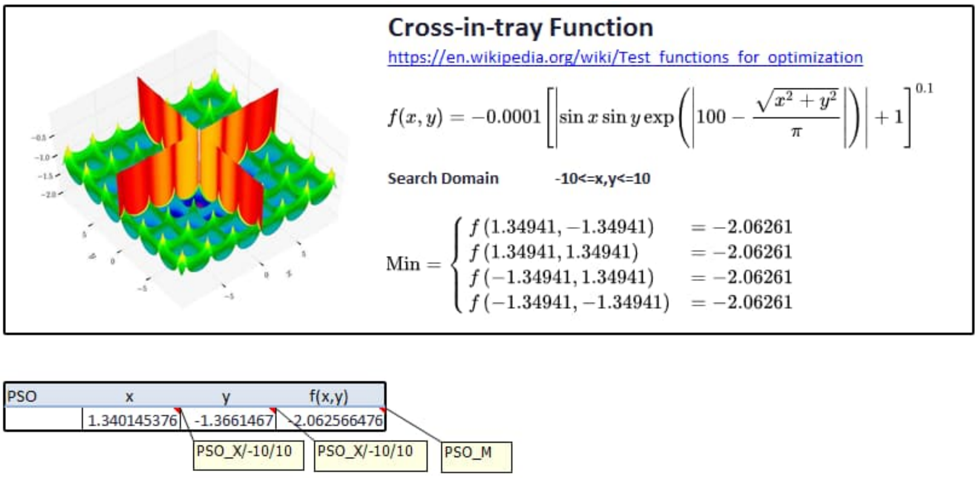

Fig. 24. Result review for PSO