Created in COMSOL Multiphysics 6.2

A Tubular Reactor Surrogate Model

Introduction

This example demonstrates how to increase the computational speed of a tubular reactor app by using a surrogate model, as opposed to a finite element model. A surrogate model is a simpler, usually computationally cheaper model, which is used to approximate the behavior of a more complex, and often more computationally expensive, model. The faster model evaluation offered by the surrogate model provides the user of the app a more interactive user experience and makes it easier to spread the use of simulations in an organization.

This document shows how to create the surrogate model by training a deep neural network (DNN). The design of the app that uses this surrogate model is described in a separate document.

The surrogate model training is based on output data from a large parametric sweep of the embedded tubular reactor model. In order to make this parametric sweep efficient, a design of experiments method is used.

The surrogate model presented here extends on the model Tubular Reactor with Nonisothermal Cooling Jacket available in the Application Libraries at

```
COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor
```

The corresponding app extends on the tubular reactor app available in the Application Libraries at

```
COMSOL_Multiphysics/Applications/tubular_reactor
```

The extended app that uses the surrogate model is available at

```
COMSOL_Multiphysics/Applications/tubular_reactor_surrogate
```

Model Definition

The app is based on a model of a tubular reactor, which is used to analyze an elementary, exothermic, and irreversible reaction. The reactions take place in a liquid phase and the reactor is operating in the laminar flow regime. Temperature control is maintained via a cooling jacket and the model focuses on the steady-state behavior of the reactor.

The reaction is a conversion of chemical species A, B, and C in a liquid:

$$A + B \rightarrow C \tag{1}$$

Here, A represents propylene oxide, B represents water, and C represents propylene glycol. The reaction kinetics are first order in regard to the concentration of species A.

Figure 1 describes the reactor model.

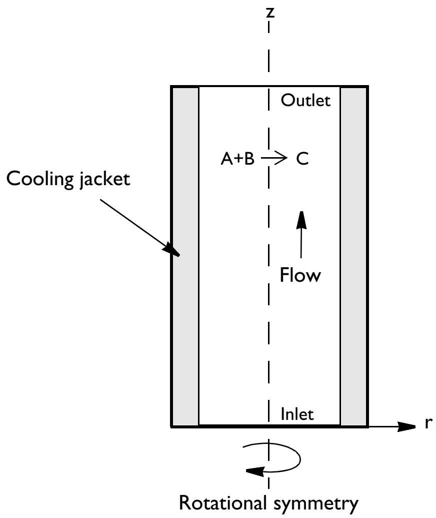


Figure 1: Schematic of the reactor model.

INPUTS AND OUTPUTS

The simulation is used to analyze the compositional variations in both the radial and axial directions. The results include the spatial distribution of the temperature and the chemical species concentrations, as well as various quantities that can be derived from these. In the app, the results are limited to the temperature and the conversion factor of species A. The conversion factor ranges from 0 to 1, indicating the extent of the conversion of species A into species C. A factor of 0 signifies no conversion, while a factor of 1 represents complete conversion into Species C.

The app has three scalar inputs:

- Activation energy, E
- Thermal conductivity, k
- Heat of reaction, dH_{rx}

In order to train the surrogate model, two additional inputs are needed: the radial, r, and axial, z, coordinates in a cylindrical coordinate system.

The two outputs, temperature, T, and conversion factor, x_A, are then viewed as functions of the five inputs:

- $T = f(r, z, E, k, dH_{\mathrm{rx}})$
- $x_A = g(r, z, E, k, dH_{\mathrm{rx}})$

In a traditional model, the finite element solution would provide us with the functions f and g. However, here we replace the finite element model with a surrogate model consisting of two new functions f_{s} and g_{s} (using the subscript "s" for surrogate), which will approximate the finite element solution and be much faster to evaluate:

- $T = f_{\mathrm{s}}(r, z, E, k, dH_{\mathrm{rx}})$
- $x_A = g_{\mathrm{s}}(r, z, E, k, dH_{\mathrm{rx}})$

Note: In the user interface of the software, the surrogate model functions f_{s} and g_{s} are called `dnn1_1` and `dnn1_2` (see below).

DESIGN OF EXPERIMENTS

In order to train the surrogate model, consisting of the functions f_{s} and g_{s}, a large number of data points are needed to fully reveal how the inputs ($r, z, E, k, dH_{\mathrm{rx}}$) map to the outputs T and x_A. The full finite element model of the tubular reactor will need to be solved for each such data point in the five-dimensional input space defined by the input parameters. Although a straightforward parametric sweep could densely and uniformly distribute input points throughout a five-dimensional grid, such an approach would be inefficient. While random sampling may serve as an alternative, it has inherent drawbacks, such as nonuniform sampling and potential failure to cover the entire input space.

Instead, a more strategic approach is to use a design of experiments method, carefully sampling within the parameter space. In COMSOL Multiphysics, a dedicated **Surrogate Model Training** study uses Latin hypercube sampling (LHS), which is a design of experiments method that generates a dataset that uniformly covers the input space without requiring an excessive number of finite element computations. This is a clear advantage over uniform grid or random sampling. LHS is a more efficient method for data generation intended for the purpose of training a surrogate model.

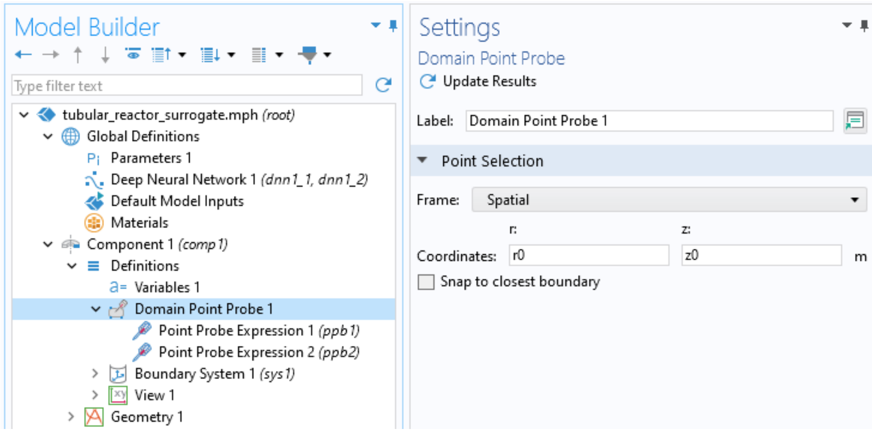
The more data points that you use, the more accurately the surrogate model will be able to represent the actual solution. However, generating a large number of data points require a large number of simulations to be run, and there is a tradeoff between the time it takes to generate all the data points and the desired accuracy of the surrogate model.

The figure below shows the first portion of a table generated by the **Surrogate Model Training** study.

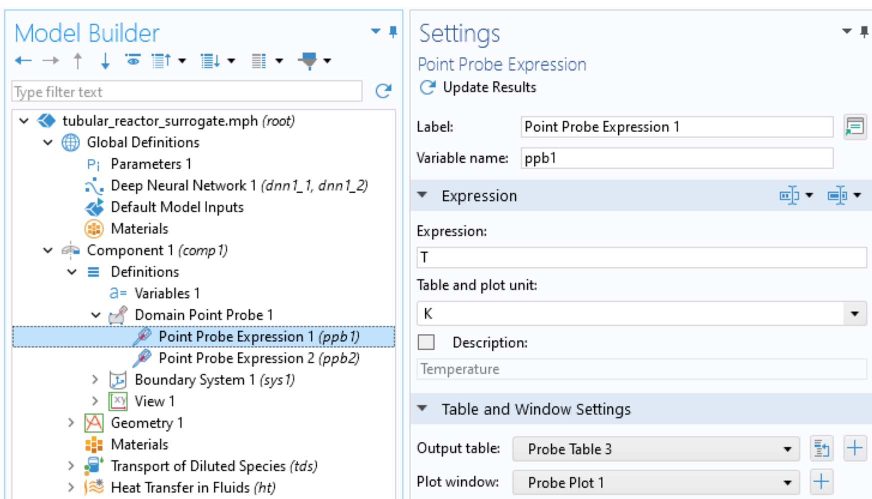
r0	z0	E	ke	dHrx	comp1.ppb1	comp1.ppb2
0.034867	0.10526	78205	1.4771	-95217	312.47	0.011570
0.0036703	0.43447	72593	3.2852	-85193	336.00	0.62326
0.075130	0.087665	74005	2.2266	-70917	314.71	0.10475
0.066744	0.50480	78832	2.7607	-96906	309.06	0.061759
0.056261	0.24118	72033	4.8796	-94083	340.10	0.70428
0.079438	0.59099	75657	3.5169	-68933	300.36	0.19454
0.059902	0.46475	71828	3.7092	-75968	338.64	0.98661
0.016618	0.67873	78108	0.98348	-93440	315.10	0.076463
0.093237	0.80800	77312	4.1000	-86099	287.82	0.089776
0.086699	0.14976	75048	4.3877	-84598	299.43	0.095224
0.038944	0.34464	74558	0.34555	-98067	320.76	0.20614
0.036991	0.55241	71728	2.4084	-99975	357.62	1.0000
0.040883	0.69070	79128	2.0557	-76746	313.86	0.058797
0.0049117	0.82853	76095	2.8845	-80016	320.14	0.22578
0.047550	0.13495	77880	3.0951	-78306	312.65	0.019162
0.033420	0.030419	73749	4.3640	-79867	312.63	0.018462
0.088304	0.45213	76636	5.3503	-99017	292.27	0.080251
0.083740	0.22368	74662	2.3689	-1.0073E5	307.35	0.20911
0.021889	0.95200	72992	3.9495	-90330	352.85	0.99994
0.015000	0.62403	75693	3.4049	-1.0103E5	321.91	0.21799
0.013696	0.73910	73260	5.1106	-74168	340.55	0.82452
0.062650	0.88584	72767	0.61902	-91789	352.30	1.0000
0.054614	0.60228	74852	5.5226	-85476	317.89	0.34961
0.073566	0.84372	73371	4.1960	-98784	324.95	0.94695
0.065779	0.40804	78968	5.4007	-83895	306.01	0.040212
0.099631	0.21912	78526	3.8388	-73505	282.42	0.052959
0.049637	0.97335	76372	2.6027	-92785	322.60	0.35026
0.069048	0.98136	73640	3.7242	-79239	321.42	0.86497
0.052674	0.041144	77516	4.5190	-95723	312.29	0.0071179
0.081911	0.75166	72637	1.2182	-71665	318.33	0.98694

The two last columns labeled `comp1.ppb1` and `comp1.ppb2` are the values from two domain point probe expressions used to sample the temperature and conversion rate, respectively, in the cylindrical (r, z) coordinate system.

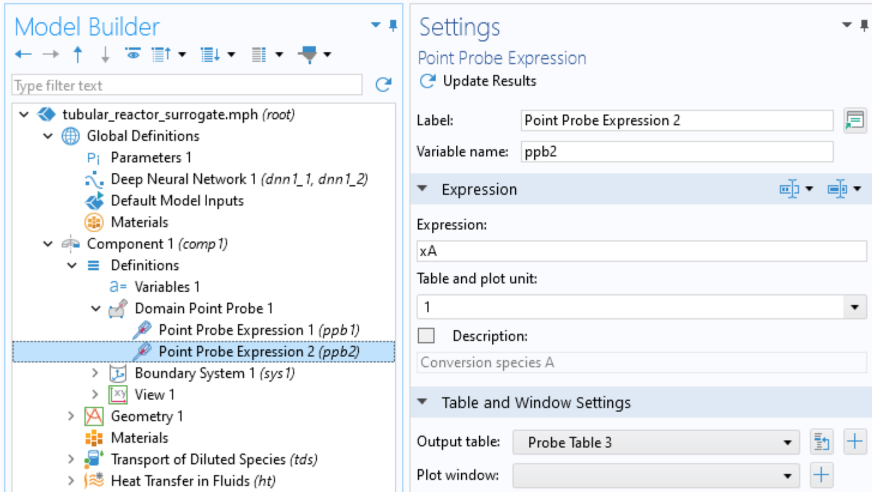
The figure below shows the coordinate settings for the **Domain Point Probe**.



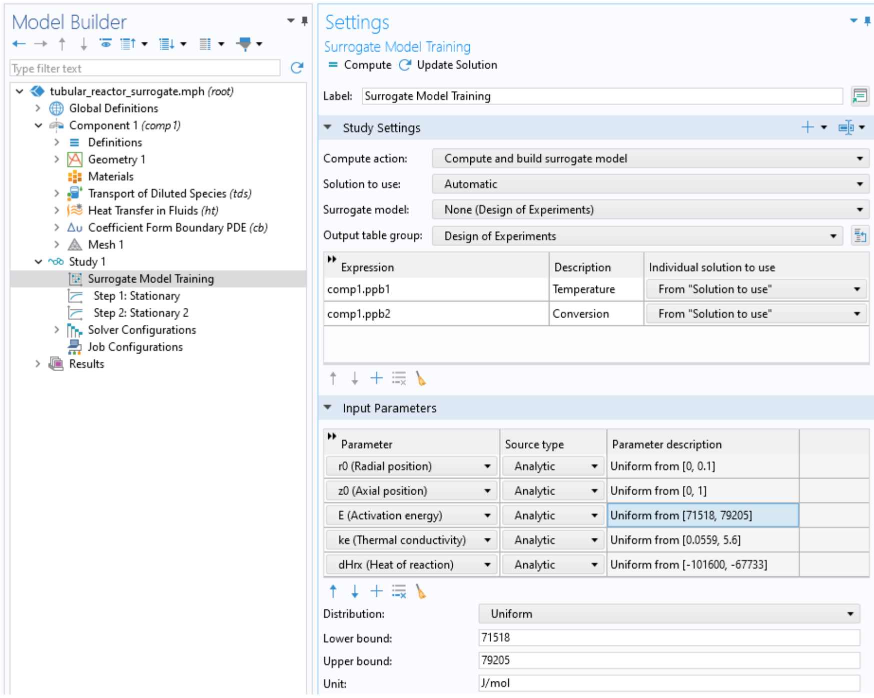
The figure below shows the settings for probing the temperature, T, in the first **Point Probe Expression (ppb1)**.



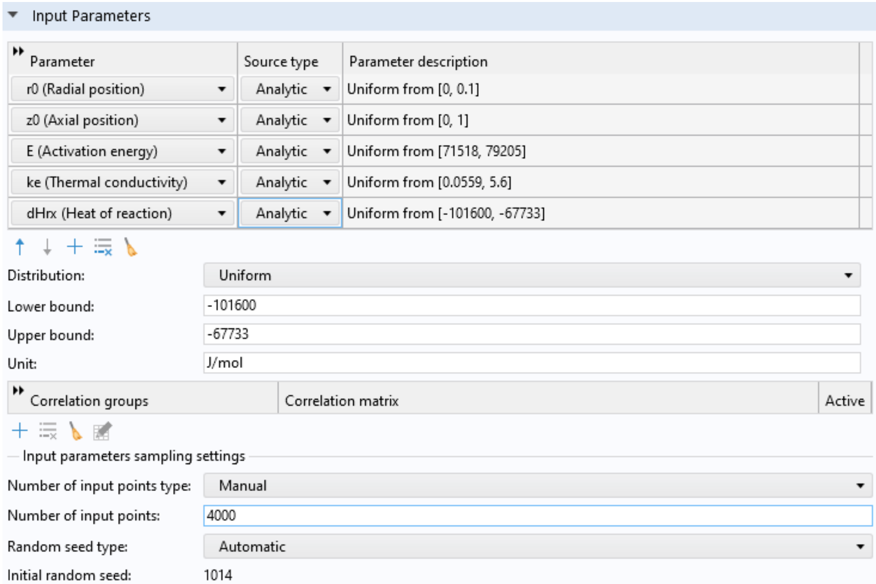
The second **Point Probe Expression (ppb2)** similarly includes an expression for the conversion rate xA, as shown below.



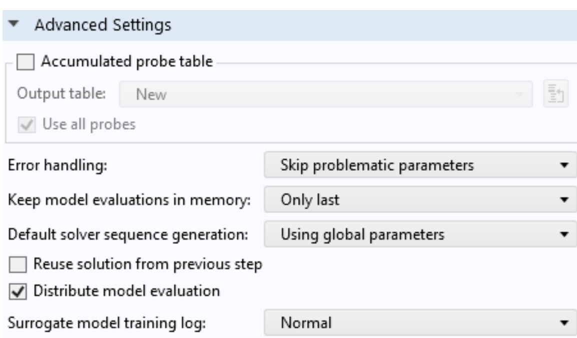
The figure below shows the corresponding **Surrogate Model Training** study settings in the Model Builder.



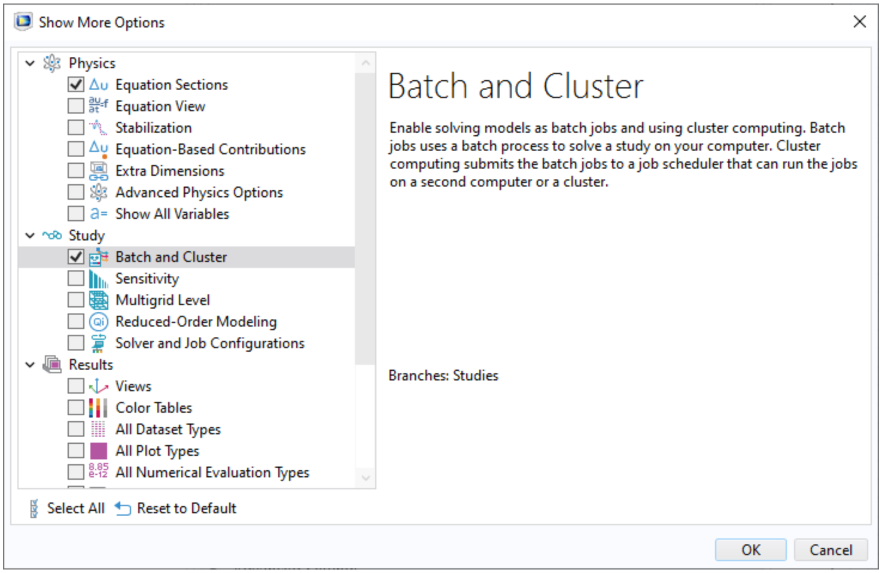
In this example, the number of finite element solutions, or **Number of input points**, used for the data generation is set to 4000. This number is chosen empirically to train the surrogate model to a sufficient degree of accuracy. The figure below shows the portion of the **Surrogate Model Training** study that includes the setting for the **Number of input points**.



If you have access to a cluster, then you can utilize parallelization to speed up the surrogate model data generation. This option is called **Distribute model evaluation** and is available in the **Advanced Settings** section of the **Surrogate Model Training** study window, as shown below.



To enable this option, select the **Batch and Cluster** option in the **Show More Options** dialog box, as shown below.



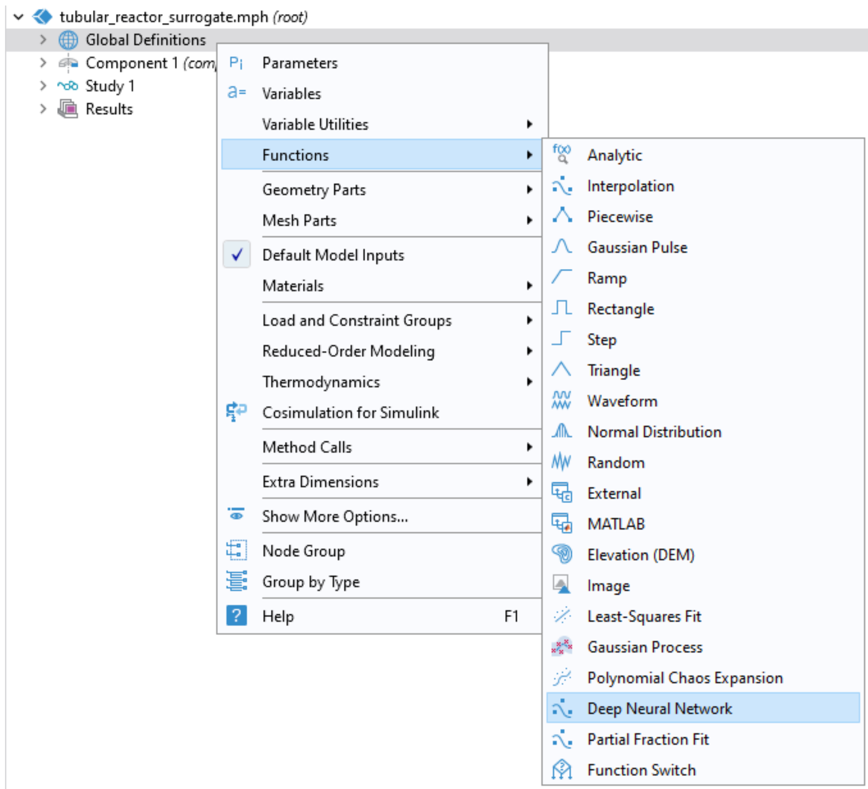
Note also that for more robust training, select the option **Skip problematic parameters** for **Error handling**. This ensures that the training continues even if the model cannot be computed for certain sets of input parameters. For example, if a set of input parameters generates an invalid geometry model or mesh, or if the solver does not converge.

For more information on cluster settings, see the *COMSOL Multiphysics Reference Manual*.

There is an extension of this model that generates training data more quickly. It uses a technique for efficient geometry sampling by combining design of experiments with coordinate-based sampling of the solution through import, export, and concatenation of text files. The training time is then reduced from hours to minutes. The version with efficient sampling is available for download at https://www.comsol.com/model/tubular-reactor-surrogate-model-application-with-efficient-geometry-sampling-120391.

SURROGATE MODELS

The **Surrogate Model Training** study is used to generate the data needed for the training and includes the option of also setting up the surrogate model automatically after the data generation step. In this example, the latter option is not used but the study is only used for generating the data and, in a separate step, the surrogate model is created and trained based on this data. The surrogate model is available as a function in the **Model Builder** under **Global Definitions**. The figure below shows the **Functions** menu with the **Deep Neural Network** surrogate model highlighted.



Included with COMSOL Multiphysics is the **Deep Neural Network** (DNN) surrogate model. With the Uncertainty Quantification Module you will additionally get the surrogate models **Gaussian Process** (GP) and **Polynomial Chaos Expansion** (PCE). The GP and PCE surrogate models include uncertainty estimates in regards to the quality of the data fit whereas the DNN model does not give any uncertainty estimates. The benefit with the DNN model is that it can handle larger datasets than the GP and PCE models, which are both limited to 2000 input points (data points).

To get an intuitive feeling for the surrogate models you can think of them as a nonlinear generalization of the linear interpolation functions available in COMSOL Multiphysics. However, the linear interpolation functions can only handle three-dimensional inputs; in the case of spatial interpolation this would be the spatial coordinates x, y, and z. The methods used for linear interpolation do not easily generalize to higher-dimensional unstructured input data and it is hard to find efficient methods for linear interpolation in such cases. The surrogate models, on the other hand, can handle an arbitrary number of inputs. In addition, the surrogate models are well suited for handling complex nonlinear relationships in the data.

THE DEEP NEURAL NETWORK SURROGATE MODEL

A DNN model consists of an input layer, a series of hidden layers, and an output layer. Each layer consists of a number of nodes, or neurons. Figure 2 shows a graph for a network with three hidden layers, 5 input nodes, and 2 output nodes.

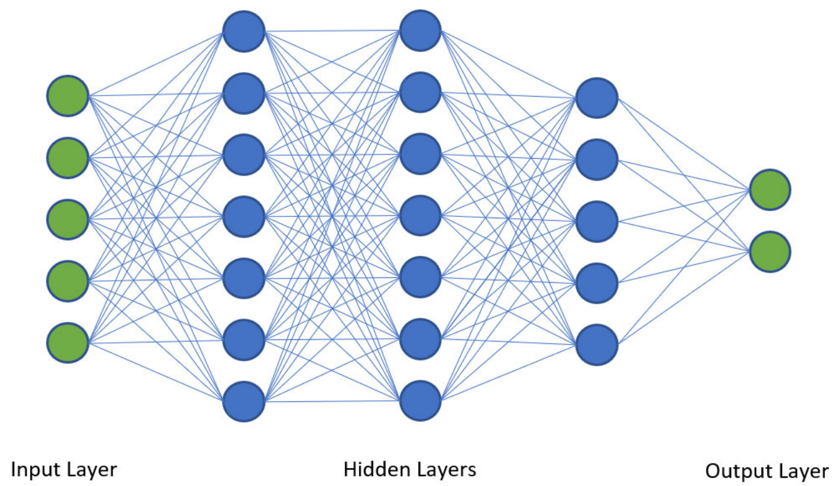
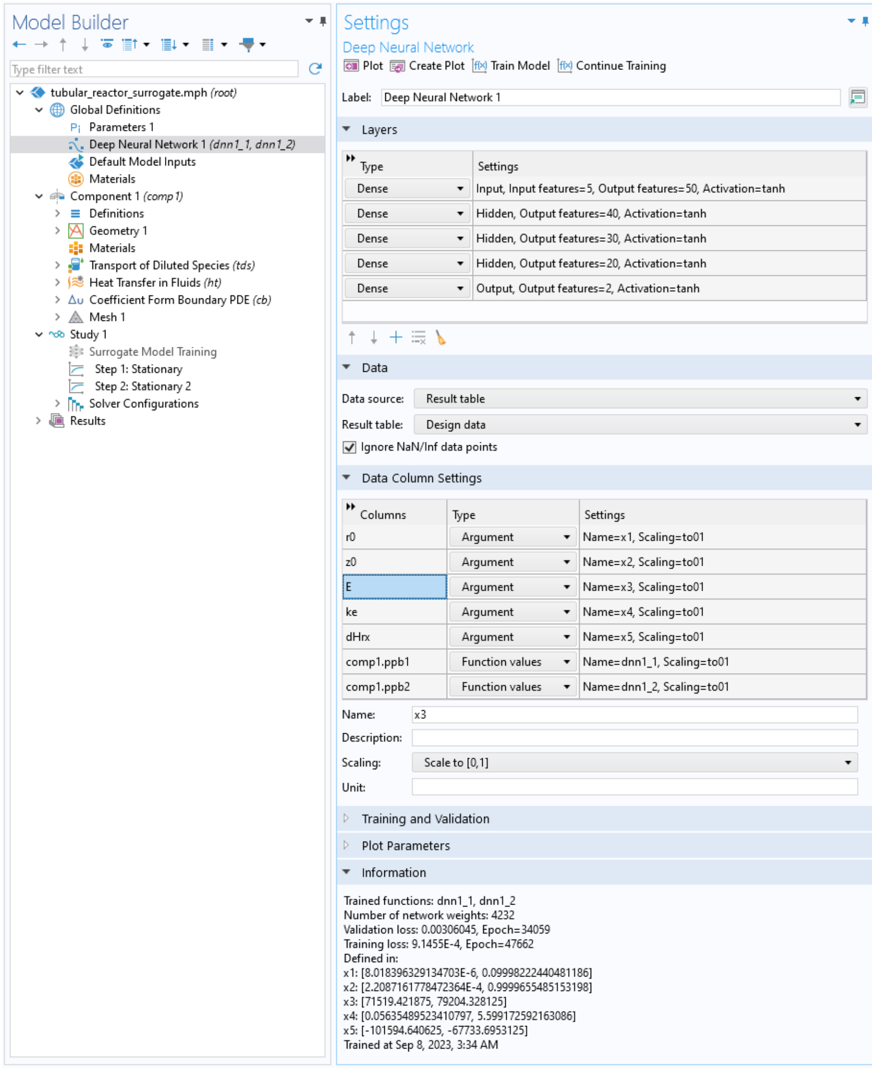


Figure 2: A neural network with five layers.

The network shown in the figure above could in principle be used for the tubular reactor model; however, it is a bit too simple to represent the model with enough accuracy. Instead, in the tubular reactor example, there are four hidden layers with 50, 40, 30, and 20 nodes, respectively.

The figure below shows the settings window for the surrogate model used in the tubular reactor example.



The **Data source** option supports referencing a **Results table** or a **File**. The **File** option supports data on the COMSOL spreadsheet format. In this example, the surrogate model functions for the temperature and conversion are called `dnn1_1` and `dnn1_2`, respectively.

Note: You can differentiate these functions with respect to any of their input arguments. For example, `d(dnn1_1(x1,x2,x3,x4,x5),x2)` is the partial derivative of `dnn1_1` with respect to the second input argument. This implies, among other things, that you can use the surrogate model functions for gradient-based optimization.

Choosing the number of layers and nodes in a neural network is often an iterative process that involves a combination of knowledge about the specific problem and data, empirical testing, and a bit of trial and error. A network with too few hidden layers or nodes may not be complex enough to serve as an accurate surrogate model. A network with too many hidden layers or nodes may suffer from so-called overfitting, where the network performs well on the training data but fails to generalize to a dataset that it was not trained on. A model with a large number of hidden layers or nodes will also be slower to evaluate than a lighter model.

Figure 3 shows a few additional configurations of neural networks that can serve as inspiration for your own modeling projects:

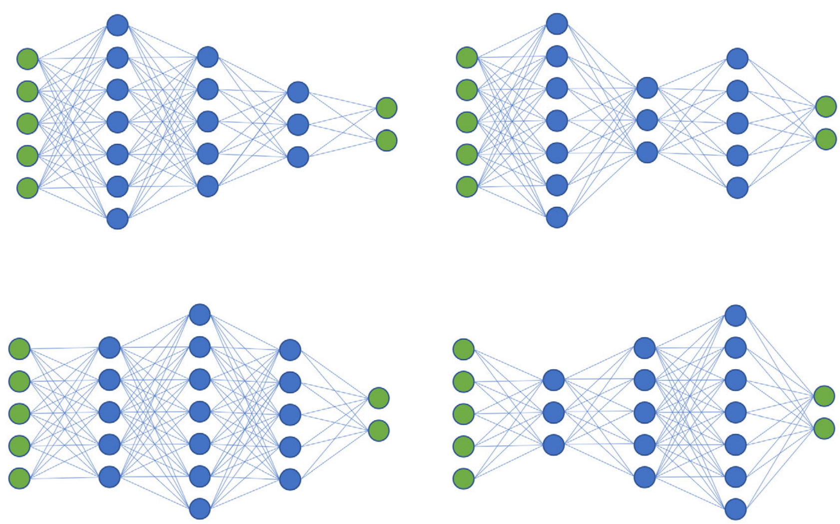


Figure 3: Examples of deep neural network configurations.

In COMSOL Multiphysics, you define the number of output features from each layer, instead of defining the number of nodes. This means that, for the tubular reactor example, you specify 50,40, 30, and 20 for the hidden layers. The number of input and output features are automatically determined from the list of arguments and function values in the **Column Settings** section in the **Settings** window of the **Deep Neural Network** surrogate model.

TRAINING THE SURROGATE MODEL

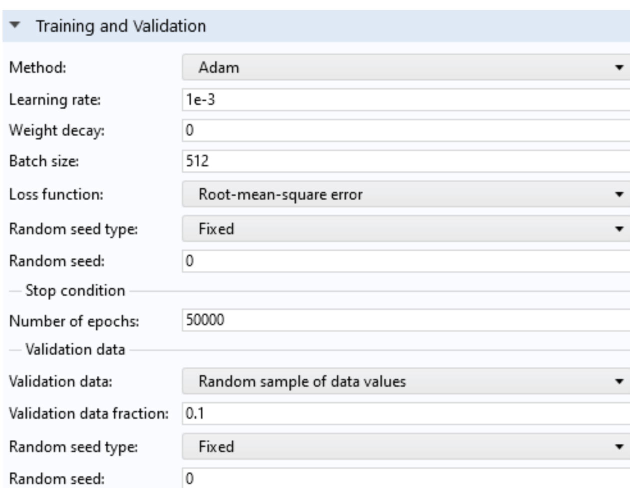
Associated with each edge in the neural network graph is a weight. When the neural network is trained on the data generated by the **Surrogate Model Training** study, the weights, together with other network parameters, called biases, are optimized so as to minimize the error between the surrogate model and the finite element model. This error is known as the loss function, L. The loss function can be of different kinds and the default setting is a root-mean-squared error (RMSE) loss function.

Based on the previously defined functions f, g, f_s, and g_s, we can write the RMSE loss function schematically as

$$L = \sqrt{\frac{1}{N}\sum((f-f_s)^2+(g-g_s)^2)}$$

where N is the number of training data points and validation data points corresponding to the training loss and validation loss, respectively.

In the **Deep Neural Network Settings** window, in the **Training and Validation** section, you can find the optimization solver parameters, also known as hyperparameters shown in the figure below.



Some of the most important parameters are the **Learning rate**, **Batch size**, and **Number of epochs**. The **Learning rate**, which can be likened in some ways to numerical damping in a nonlinear Newton solver, determines the step size during the optimization process. A learning rate that is too small can lead to the model getting stuck in a local minimum, while a learning rate that is too large can result in overshooting the minimum and poor convergence. The **Batch size** denotes how the training data is divided up into subsets during the optimization process. A too small batch size can lead to noisy gradient updates and longer training times, while a too large batch size might lead to poor generalization and inefficient utilization of computational resources. The **Number of epochs**, which defines the number of complete passes through the entire dataset, plays an important role in the learning process. Too few epochs can result in underfitting, where the model has not adequately learned from the training data, while too many epochs can lead to overfitting, where the model learns the noise in the training data and performs poorly on new, unseen data. In this example, the **Number of epochs** is empirically set to 50000 and the other parameters are left at their default values.

To train the surrogate model you click the **Train Model** button at the top of the **Deep Neural Network Settings** window. During the training process, you can monitor the progress in terms of the loss function versus epoch in the **Convergence Plot** window, as shown in Figure 4 below after 3000 epochs.

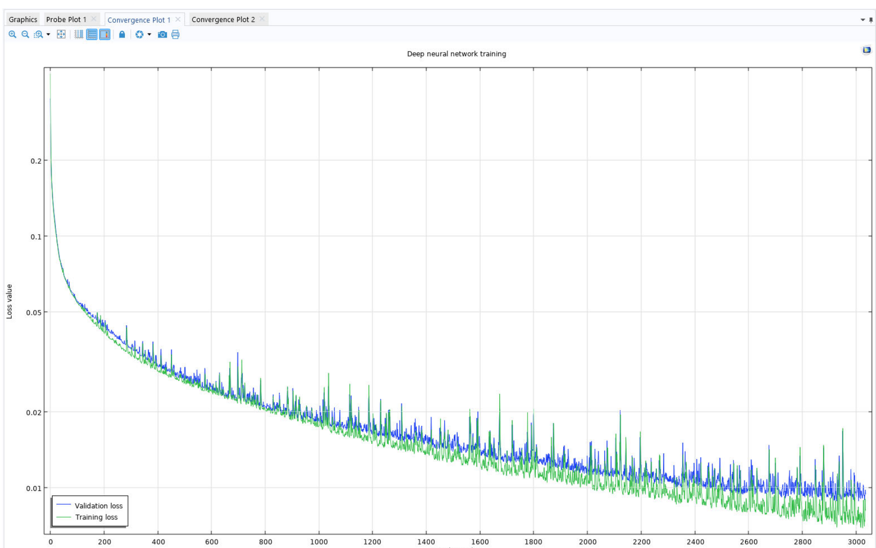


Figure 4: Training and validation losses after 3000 epochs.

The convergence plot shows two versions of the loss function: the **Training loss** and the **Validation loss**. The **Training loss** shows the loss function with respect to the main part of the training data created by the **Surrogate Model Training** study. A random portion of the data is set aside and treated as unseen data for validation purposes and this is what the **Validation loss** represents.

In more detail, the training loss measures how well the neural network fits the training data. It decreases as the model learns during training. However, if it becomes too low, the model might be overfitting, meaning it is learning the training data too closely and may perform poorly on unseen data.

The validation loss measures the model's performance on the validation data. It gives an estimate of how well the model will generalize to new, unseen data. If validation loss starts increasing while training loss is still decreasing, it usually indicates overfitting.

The hyperparameters are tuned to find a balance where both training and validation loss are minimized, indicating the model has learned well and can also generalize well to new data.

Note that if you have very few data points, you will not see the convergence graph but only the contents of the **Information** section in the **Deep Neural Network** settings window, as shown below.

```
▼ Information

Trained functions: dnn1_1, dnn1_2
Number of network weights: 4232
Validation loss: 0.00306045, Epoch=34059
Training loss: 9.1455E-4, Epoch=47662
Defined in:
x1: [8.018396329134703E-6, 0.09998222440481186]
x2: [2.2087161778472364E-4, 0.9999655485153198]
x3: [71519.421875, 79204.328125]
x4: [0.05635489523410797, 5.599172592163086]
x5: [-101594.640625, -67733.6953125]
Trained at Sep 8, 2023, 3:34 AM
```

Results and Discussion

The results plots are similar to the model Tubular Reactor with Nonisothermal Cooling Jacket. However, instead of 2D plots this model includes the temperature and conversion as revolved 3D plots. In addition, the corresponding revolved plots for the surrogate model are included (not shown here).

The revolved plots Figure 5 and Figure 6 show the computed temperature and conversion, respectively. These plots show that where the temperature is low, little conversion takes place and vice versa. This is because the rate of the reaction is temperature dependent. The low temperature closest to the wall is due to the coolant. The surrogate model plots are virtually identical in appearance (not shown here).

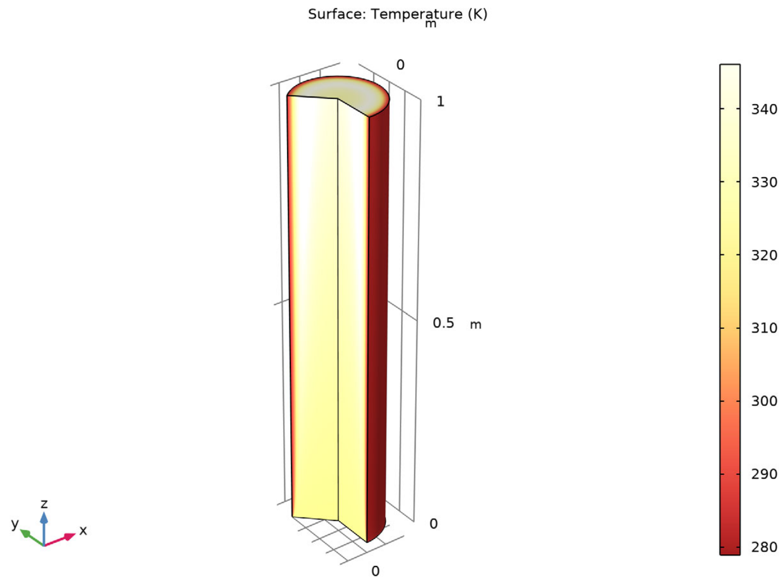


Figure 5: Computed temperature.

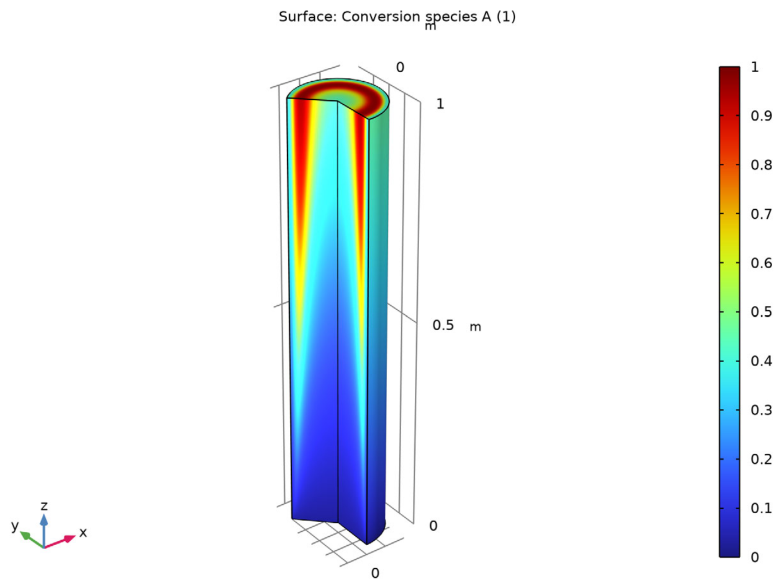


Figure 6: Computed conversion.

The computed temperature and conversion surface profiles at the outlet are shown below together with the profiles predicted by the surrogate model. The plots corresponding to the finite element solution is labeled “computed” and the plots corresponding to the surrogate model is labeled “preview”. The term preview comes from the fact that the surrogate model plots are used as preview plots in the corresponding app where this model is used as the embedded model.

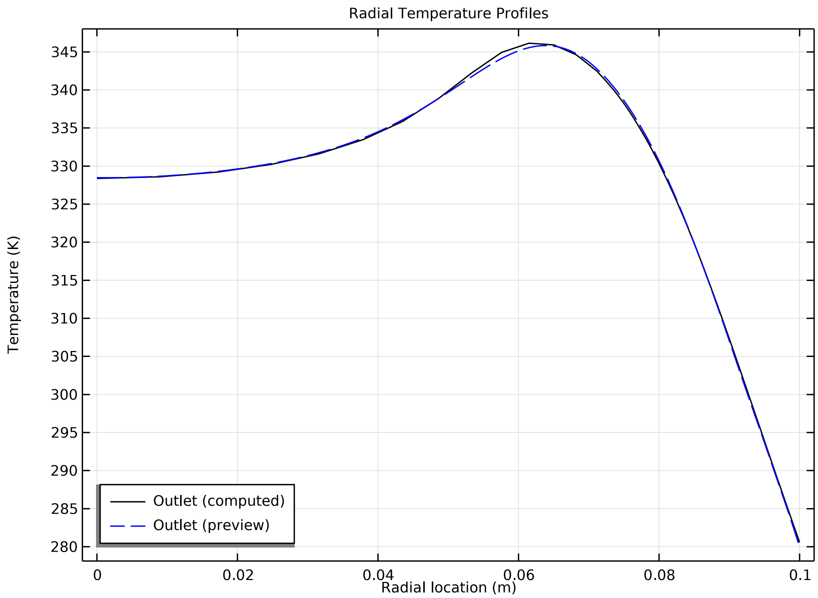


Figure 7: Computed and predicted temperature profiles at the outlet.

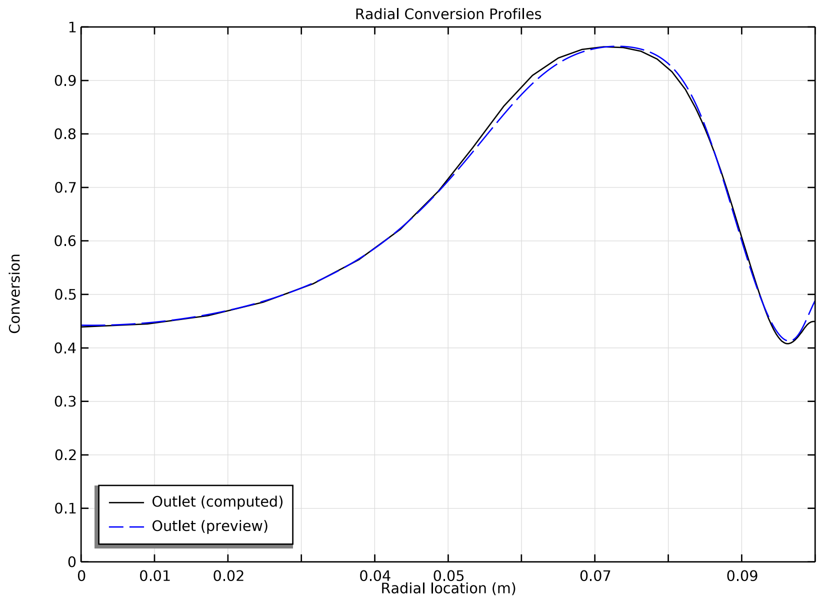


Figure 8: Computed and predicted conversion profiles at the outlet.

Application Library path: `COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor_surrogate`

Modeling Instructions

Start by loading the original tubular reactor model.

APPLICATION LIBRARIES

1 From the **File** menu, choose **Application Libraries**.

2 In the **Application Libraries** window, select **COMSOL Multiphysics>Chemical Engineering>tubular_reactor** in the tree.

3 Click **Open**.

Define two additional parameters for the radial and axial positions, respectively. These parameters will be used to probe the solution at various positions in the model in order to train the surrogate model.

GLOBAL DEFINITIONS

Parameters 1

1. In the **Model Builder** window, under **Global Definitions** click **Parameters 1**.
2. In the **Settings** window for **Parameters**, locate the **Parameters** section.
3. In the table, enter the following settings:

Name	Expression	Value	Description
`r0`	`0[m]`	0 m	`Radial position`
`z0`	`0[m]`	0 m	`Axial position`

COMPONENT 1 (COMP1)

In the **Model Builder** window, expand the **Component 1 (comp1)** node.

DEFINITIONS

Define two Domain Point Probes to use for sampling the solution values for the temperature, `T`, and conversion, `xA`, respectively.

Domain Point Probe 1

1. In the **Model Builder** window, expand the **Component 1 (comp1)>Definitions** node.
2. Right-click **Definitions** and choose **Probes>Domain Point Probe**.
3. In the **Settings** window for **Domain Point Probe**, locate the **Point Selection** section.
4. In row **Coordinates**, set **r** to `r0`.
5. In row **Coordinates**, set **z** to `z0`.

Point Probe Expression 1 (ppb1)

1. In the **Model Builder** window, expand the **Domain Point Probe 1** node, then click **Point Probe Expression 1 (ppb1)**.
2. In the **Settings** window for **Point Probe Expression**, locate the **Expression** section.
3. In the **Expression** text field, type `T`.
4. Click to expand the **Table and Window Settings** section.

Point Probe Expression 2 (ppb2)

1 In the **Model Builder** window, right-click **Domain Point Probe 1** and choose **Point Probe Expression**.

2 In the **Settings** window for **Point Probe Expression**, locate the **Expression** section.

3 In the **Expression** text field, type `xA`.

STUDY 1

Surrogate Model Training

1 In the **Model Builder** window, expand the **Study 1** node.

2 Right-click **Study 1** and choose **Surrogate Model Training**.

The quantities of interest will be the temperature, `T`, and conversion, `xA`, respectively, with values from the Domain Point Probes.

3 In the **Settings** window for **Surrogate Model Training**, locate the **Study Settings** section.

4 Find the **Quantities of interest (Outputs)** subsection. Click + **Add** twice.

5 In the table, enter the following settings:

Expression	Description	Individual solution to use
`comp1.ppb1`	`Temperature`	From "Solution to use"
`comp1.ppb2`	`Conversion`	From "Solution to use"

Next, define the input parameters `r0`, `z0`, `E`, `ke`, and `dHrx`. These represent the radial and axial positions, the activation energy, thermal conductivity, and heat of reaction, respectively. Additionally, specify the parameter bounds. These values are taken from the geometry dimension as well as from the original tubular reactor app. The number of sampled data points is 4000.

6 Locate the **Input Parameters** section. Click + **Add** five times.

7 Update the first column in the table so that the input parameters appear in the following order:

Parameter
r0 (Radial position)
z0 (Axial position)
E (Activation energy)
ke (Thermal conductivity)
dHrx (Heat of reaction)

8 In the table, click to select the cell at row number 1 and column number 3.

9 In the **Lower bound** text field, type `0`.

10 In the **Upper bound** text field, type `0.1`.

11 In the table, click to select the cell at row number 2 and column number 3.

12 In the **Lower bound** text field, type `0`.

13 In the **Upper bound** text field, type `1`.

14 In the table, click to select the cell at row number 3 and column number 3.

15 In the **Lower bound** text field, type `71518`.

16 In the **Upper bound** text field, type `79205`.

17 In the table, click to select the cell at row number 4 and column number 3.

18 In the **Lower bound** text field, type `0.0559`.

19 In the **Upper bound** text field, type `5.6`.

20 In the table, click to select the cell at row number 5 and column number 3.

21 In the **Lower bound** text field, type `-101600`.

22 In the **Upper bound** text field, type `-67733`.

23 Find the **Input parameters sampling settings** subsection. In the **Number of input points** text field, type `4000`.

24 Locate the **Advanced Settings** section. From the **Error handling** list, choose **Skip problematic parameters**. This setting makes the training stage more robust.

Start generating the table with training data. Solving for 4000 data points will take several hours, typically around six hours on a moderately powerful workstation.

25 In the **Study** toolbar, click **Compute**.

RESULTS

Define a Grid dataset for evaluating the surrogate model function.

Grid 2D 1

1 In the **Results** toolbar, click **More Datasets** and choose **Grid>Grid 2D**.

2 In the **Settings** window for **Grid 2D**, locate the **Parameter Bounds** section.

3 Find the **First parameter** subsection. In the **Name** text field, type `x1`.

4 In the **Maximum** text field, type `0.1`.

5 Find the **Second parameter** subsection. In the **Name** text field, type `x2`.

6 Click to expand the **Grid** section. In the **x1 resolution** text field, type `25`.

7 In the **x2 resolution** text field, type 75.

8 Locate the **Data** section. From the **Source** list, choose **Function**.

The next step defines the deep neural network function. The network will have four hidden layers with 50, 40, 30, and 20 nodes, respectively. Increase the number of epochs (solver iterations) to 50000 and start training the model.

GLOBAL DEFINITIONS

Deep Neural Network 1

1 In the **Home** toolbar, click f(x) **Functions** and choose **Global>Deep Neural Network**.

2 In the **Settings** window for **Deep Neural Network**, locate the **Layers** section.

3 Click + **Add** five times.

4 In the table, click to select the cell at row number 1 and column number 4.

5 In the **Output features** text field, type 50.

6 In the table, click to select the cell at row number 2 and column number 4.

7 In the **Output features** text field, type 40.

8 In the table, click to select the cell at row number 3 and column number 4.

9 In the **Output features** text field, type 30.

10 In the table, click to select the cell at row number 4 and column number 4.

11 In the **Output features** text field, type 20.

12 In the table, click to select the cell at row number 5 and column number 4.

13 In the **Output features** text field, type 2.

14 Locate the **Data** section. From the **Data source** list, choose **Result table**.

15 Locate the **Data Column Settings** section. In the table, enter the following settings:

Columns	Type	Settings
comp1.ppb1	Function values	Name=dnn1_col6, Scaling=to01

16 In the table, click to select the cell at row number 6 and column number 3.

17 In the **Name** text field, type dnn1_1.

18 In the table, click to select the cell at row number 7 and column number 3.

19 In the **Name** text field, type dnn1_2.

20 Locate the **Training and Validation** section. Find the **Stop condition** subsection. In the **Number of epochs** text field, type 50000.

21 Click **Train Model**.

STUDY 1

Disable the Surrogate Model Training study to prepare the model for being used as the embedded model in an app.

Surrogate Model Training

1 In the **Model Builder** window, under **Study 1** right-click **Surrogate Model Training** and choose **Disable**.

2 In the **Home** toolbar, click **Compute**.

RESULTS

Now, define a Mirror dataset, a Revolution dataset, and a Grid 1D dataset for the various visualizations. Then, define different plots for the temperature and conversion. The temperature and conversion fields are visualized in 3D using a Revolution dataset and in 1D along a radius at the outlet.

Mirror 2D 2

1 In the **Results** toolbar, click **More Datasets** and choose **Mirror 2D**.

2 In the **Settings** window for **Mirror 2D**, locate the **Data** section.

3 From the **Dataset** list, choose **Grid 2D 1**.

4 Click to expand the **Advanced** section.

Revolution 2D 3

1 In the **Results** toolbar, click **More Datasets** and choose **Revolution 2D**.

2 In the **Settings** window for **Revolution 2D**, locate the **Data** section.

3 From the **Dataset** list, choose **Grid 2D 1**.

4 Click to expand the **Revolution Layers** section. In the **Start angle** text field, type -90.

5 In the **Revolution angle** text field, type 225.

6 Click to expand the **Advanced** section.

Grid 2D 1

1 In the **Model Builder** window, click **Grid 2D 1**.

2 In the **Settings** window for **Grid 2D**, locate the **Data** section.

3 From the **Function** list, choose **Deep Neural Network 1 (dnn1_1, dnn1_2)**.

4 In the **Model Builder** window, collapse the **Results>Datasets** node.

Temperature, 3D (Revolved)

1 In the **Results** toolbar, click **3D Plot Group**.
2 In the **Settings** window for **3D Plot Group**, type `Temperature, 3D (Revolved)` in the **Label** text field.

Surface 1

1 Right-click **Temperature, 3D (Revolved)** and choose **Surface**.
2 In the **Settings** window for **Surface**, locate the **Expression** section.
3 In the **Expression** text field, type `T`.
4 Locate the **Coloring and Style** section. Click **Change Color Table**.
5 In the **Color Table** dialog box, select **Thermal>ThermalLight** in the tree.
6 Click **OK**.
7 In the **Temperature, 3D (Revolved)** toolbar, click **Plot**.
8 Click the **Zoom Extents** button in the **Graphics** toolbar.

   Compare this plot with that shown in Figure 5.

Temperature, 3D, Surrogate (Revolved)

1 In the **Home** toolbar, click **Add Plot Group** and choose **3D Plot Group**.
2 In the **Settings** window for **3D Plot Group**, type `Temperature, 3D, Surrogate (Revolved)` in the **Label** text field.
3 Locate the **Data** section. From the **Dataset** list, choose **Revolution 2D 3**.
4 Locate the **Plot Settings** section. From the **View** list, choose **View 3D 3**.

Surface 1

1 Right-click **Temperature, 3D, Surrogate (Revolved)** and choose **Surface**.

   The surrogate model for temperature is visualized by entering the expression for the corresponding function `dnn1_1`.
2 In the **Settings** window for **Surface**, locate the **Expression** section.
3 In the **Expression** text field, type `dnn1_1(x1,x2,E,ke,dHrx)`.
4 Locate the **Coloring and Style** section. Click **Change Color Table**.
5 In the **Color Table** dialog box, select **Thermal>ThermalLight** in the tree.
6 Click **OK**.
7 In the **Temperature, 3D, Surrogate (Revolved)** toolbar, click **Plot**.

Conversion, 3D (Revolved)

1 In the **Home** toolbar, click **Add Plot Group** and choose **3D Plot Group**.

2 In the **Settings** window for **3D Plot Group**, type `Conversion, 3D (Revolved)` in the **Label** text field.

Surface 1

1 Right-click **Conversion, 3D (Revolved)** and choose **Surface**.

2 In the **Settings** window for **Surface**, locate the **Expression** section.

3 In the **Expression** text field, type `xA`.

4 Locate the **Coloring and Style** section. Click **Change Color Table**.

5 In the **Color Table** dialog box, click **OK**.

6 In the **Settings** window for **Surface**, click to expand the **Range** section.

7 Select the **Manual color range** check box.

8 In the **Maximum** text field, type `1`.

9 In the **Conversion, 3D (Revolved)** toolbar, click **Plot**.

The plot should resemble the one in Figure 6.

Conversion, 3D, Surrogate (Revolved)

1 In the **Home** toolbar, click **Add Plot Group** and choose **3D Plot Group**.

2 In the **Settings** window for **3D Plot Group**, type `Conversion, 3D, Surrogate (Revolved)` in the **Label** text field.

3 Locate the **Data** section. From the **Dataset** list, choose **Revolution 2D 3**.

4 Locate the **Plot Settings** section. From the **View** list, choose **View 3D 3**.

Surface 1

1 Right-click **Conversion, 3D, Surrogate (Revolved)** and choose **Surface**.

The surrogate model for conversion is visualized by entering the expression for the corresponding function `dnn1_2`.

2 In the **Settings** window for **Surface**, locate the **Expression** section.

3 In the **Expression** text field, type `dnn1_2(x1,x2,E,ke,dHrx)`.

4 Locate the **Coloring and Style** section. Click **Change Color Table**.

5 In the **Color Table** dialog box, click **OK**.

6 In the **Settings** window for **Surface**, locate the **Range** section.

7 Select the **Manual color range** check box.

8 In the **Maximum** text field, type `1`.

9 In the **Conversion, 3D, Surrogate (Revolved)** toolbar, click **Plot**.

STUDY 1

Surrogate Model Training

1 In the **Model Builder** window, under **Study 1** right-click **Surrogate Model Training** and choose **Disable**.

2 In the **Home** toolbar, click **Compute**.

RESULTS

Now create line graphs of the temperature and conversion versus the radial position at the outlet.

Outlet

1 In the **Model Builder** window, under **Results>Datasets** click **Cut Line 2D 1**.

2 In the **Settings** window for **Cut Line 2D**, type `Outlet` in the **Label** text field.

3 Locate the **Line Data** section. In row **Point 1**, set **Z** to `L`.

4 In row **Point 2**, set **Z** to `L`.

5 Clear the **Additional parallel lines** check box.

Line Graph 1

1 In the **Model Builder** window, expand the **Results>Temperature, 1D** node, then click **Line Graph 1**.

2 In the **Settings** window for **Line Graph**, click to expand the **Legends** section.

3 Click to select row number 1 in the table.

4 Click **Delete** twice.

5 In the table, enter the following settings:

Legends
`Outlet (computed)`

Grid 1D 2

1 In the **Results** toolbar, click **More Datasets** and choose **Grid>Grid 1D**.

2 In the **Settings** window for **Grid 1D**, locate the **Parameter Bounds** section.

3 In the **Name** text field, type `x1`.

4 In the **Maximum** text field, type `0.1`.

5 Locate the **Data** section. From the **Source** list, choose **Function**.

6 From the **Function** list, choose **Deep Neural Network 1 (dnn1_1, dnn1_2)**.

7 Click to expand the **Grid** section.

Temperature, 1D

1 In the **Model Builder** window, under **Results** click **Temperature, 1D**.

2 In the **Settings** window for **1D Plot Group**, click to expand the **Window Settings** section.

Line Graph 2

1 Right-click **Temperature, 1D** and choose **Line Graph**.

2 In the **Settings** window for **Line Graph**, locate the **Data** section.

3 From the **Dataset** list, choose **Grid 1D 2**.

4 Locate the **y-Axis Data** section. In the **Expression** text field, type `dnn1_1(x1,L,E,ke,dHrx)`.

5 Click to expand the **Coloring and Style** section. Find the **Line style** subsection. From the **Line** list, choose **Dashed**.

6 From the **Color** list, choose **Blue**.

7 Locate the **Legends** section. Select the **Show legends** check box.

8 From the **Legends** list, choose **Manual**.

9 In the table, enter the following settings:

Legends
`Outlet (preview)`

Temperature, 1D

1 In the **Model Builder** window, click **Temperature, 1D**.

2 In the **Temperature, 1D** toolbar, click **Plot**.

Compare with Figure 7.

Line Graph 1

1 In the **Model Builder** window, expand the **Results>Conversion, 1D** node, then click **Line Graph 1**.

2 In the **Settings** window for **Line Graph**, locate the **Legends** section.

3 Click to select row number 1 in the table.

4 Click **Delete** twice.

5 In the table, enter the following settings:

Legends
`Outlet (computed)`

Line Graph 2

1 In the **Model Builder** window, right-click **Conversion, 1D** and choose **Line Graph**.

2 In the **Settings** window for **Line Graph**, locate the **Data** section.

3 From the **Dataset** list, choose **Grid 1D 2**.

4 Locate the **y-Axis Data** section. In the **Expression** text field, type `dnn1_2(x1,L,E,ke,dHrx)`.

5 Locate the **Coloring and Style** section. Find the **Line style** subsection. From the **Line** list, choose **Dashed**.

6 From the **Color** list, choose **Blue**.

7 Locate the **Legends** section. Select the **Show legends** check box.

8 From the **Legends** list, choose **Manual**.

9 In the table, enter the following settings:

Legends
`Outlet (preview)`

Conversion, 1D

1 In the **Model Builder** window, click **Conversion, 1D**.

2 In the **Settings** window for **1D Plot Group**, click to expand the **Title** section.

3 In the **Title** text area, type `Radial Conversion Profiles`.

4 Locate the **Axis** section. Select the **Manual axis limits** check box.

5 In the **x minimum** text field, type `0`.

6 In the **x maximum** text field, type `0.1`.

7 In the **y minimum** text field, type `0`.

8 In the **y maximum** text field, type `1`.

9 Locate the **Legend** section. From the **Position** list, choose **Lower left**.

10 In the **Conversion, 1D** toolbar, click **Plot**.

Compare with Figure 8.

Finally, change the plot titles to more easily identify the surrogate model plots (preview plots) from the computed plots.

Temperature, 3D (Revolved)

1 In the **Model Builder** window, click **Temperature, 3D (Revolved)**.

2 In the **Settings** window for **3D Plot Group**, locate the **Plot Settings** section.

3 Clear the **Plot dataset edges** check box.

4 Click to expand the **Title** section. From the **Title type** list, choose **Manual**.

5 In the **Title** text area, type `Surface: Temperature (K) (Computed)`.

6 Click the **Zoom Extents** button in the **Graphics** toolbar.

7 In the **Temperature, 3D (Revolved)** toolbar, click **Plot**.

Temperature, 3D, Surrogate (Revolved)

1 In the **Model Builder** window, click **Temperature, 3D, Surrogate (Revolved)**.

2 In the **Settings** window for **3D Plot Group**, locate the **Title** section.

3 From the **Title type** list, choose **Manual**.

4 In the **Title** text area, type `Surface: Temperature (K) (Preview)`.

Conversion, 3D (Revolved)

1 In the **Model Builder** window, click **Conversion, 3D (Revolved)**.

2 In the **Settings** window for **3D Plot Group**, locate the **Plot Settings** section.

3 Clear the **Plot dataset edges** check box.

4 Locate the **Title** section. From the **Title type** list, choose **Manual**.

5 In the **Title** text area, type `Surface: Conversion, species A (Computed)`.

Conversion, 3D, Surrogate (Revolved)

1 In the **Model Builder** window, click **Conversion, 3D, Surrogate (Revolved)**.

2 In the **Settings** window for **3D Plot Group**, locate the **Title** section.

3 From the **Title type** list, choose **Manual**.

4 In the **Title** text area, type `Surface: Conversion, species A (Preview)`.