



Surrogate Models

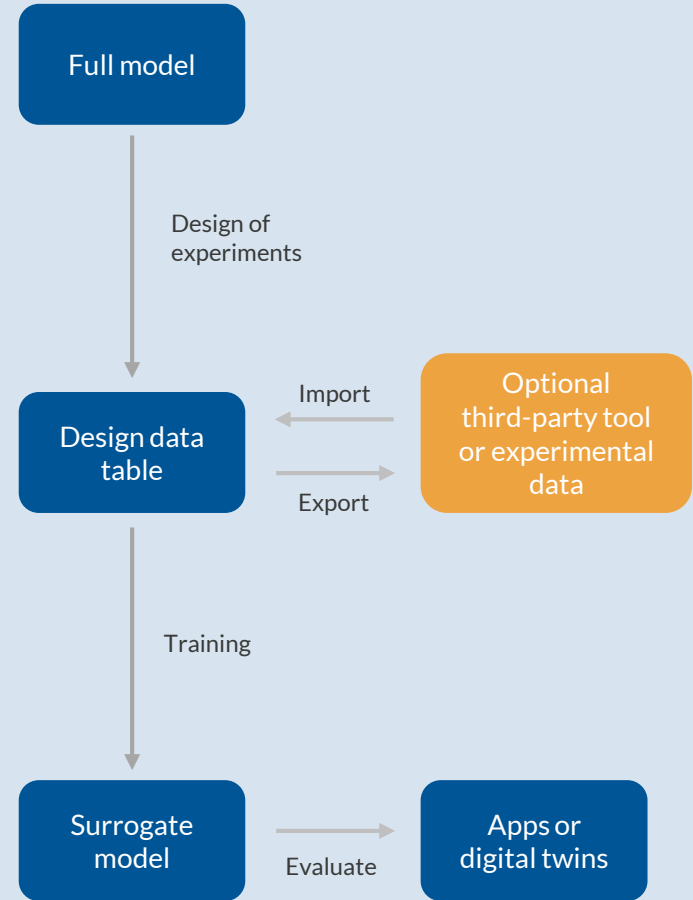
in COMSOL Multiphysics

Loïc Renversade

Applications
COMSOL France

Surrogate Models

- Powerful new functionality for creating and using data-driven surrogate models in version 6.2
- What is a surrogate model?
 - A simpler, less computationally expensive model
 - Approximates the behavior of more complex, computationally intensive models
- Benefits of surrogate models:
 - Faster model evaluation enhances interactive user experience
 - Enable new areas of use for apps and digital twins



Surrogate Model Training Study

- Use the *Surrogate Model Training* study to generate a *Design Data* table.
- Specify *Input Parameters* and *Quantities of Interest* (output parameters).
- The study generates data in the parameter space based on a design-of-experiments method.
- The study optionally defines a surrogate model function when finished. (Otherwise, add one manually.)

tubular_reactor_surrogate.mph - COMSOL Multiphysics

File Home Definitions Geometry Sketch Materials Physics Mesh Study Results Developer Colored Selections

Application Builder Model Manager Component Add Component Workspace Model

Parameters Variables Functions Build All Import LiveLink Part Libraries Add Material Transport of Diluted Species Add Add Build Mesh Compute Study Add Study Conversion, 3D, Surrogate (Revolved)

Model Builder

Surrogate Model Training

Compute Update Solution

Label: Surrogate Model Training

Study Settings

Compute action: Compute and build surrogate model

Solution to use: Automatic

Surrogate model: Design of experiments (No surrogate model)

Output table group: Design of Experiments

Quantities of interest (Outputs)

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

Input Parameters

Parameter	Source type	Parameter description
r0 (Radial position)	Analytic	Uniform from [0, 0.1]
z0 (Axial position)	Analytic	Uniform from [0, 1]
E (Activation energy)	Analytic	Uniform from [71518, 79205]
ke (Thermal conductivity)	Analytic	Uniform from [0.0559, 5.6]
dHrx (Heat of reaction)	Analytic	Uniform from [-101600, -67733]

Correlation groups Correlation matrix Active

Input parameters sampling settings

Number of input points type: Manual

Number of input points: 4000

Random seed type: Automatic

Initial random seed: 1014

Advanced Settings

Graphics

Surface: Conversion

Messages Progress Log Design Data

r0	z0	E	ke	dHrx	comp1.ppb1	comp1.ppb2
0.032140	0.33178	72563	2.6236	-87001	331.64	0.50111
0.055063	0.83791	75719	4.5499	-92537	314.94	0.31657
0.025539	0.60275	75944	3.3028	-88487	319.27	0.18309
0.021129	0.98724	74405	1.4382	-76038	337.06	0.71961
0.026463	0.56600	74256	7.0612	-92668	321.00	0.67590

The screenshot shows a window titled "Design data" containing a table with 7 columns and 50 rows of data. The columns are labeled r0, z0, E, ke, dHrx, T, and xA. The data consists of numerical values, some in scientific notation, representing the results of a surrogate model training study.

r0	z0	E	ke	dHrx	T	xA
0.032140	0.33178	72563	2.6236	-87001	331.64	0.50111
0.055063	0.83791	75719	4.5499	-82537	314.94	0.31657
0.025539	0.60275	75944	3.3028	-88487	319.27	0.18309
0.021129	0.98724	74405	1.4382	-76038	337.06	0.71961
0.076463	0.56690	74356	2.0612	-93668	321.90	0.62580
0.055536	0.85834	77315	2.0946	-96237	317.41	0.20351
0.058982	0.88040	77580	2.0485	-68005	312.85	0.16747
0.084980	0.077961	72859	2.7652	-90287	311.78	0.20125
0.057236	0.43613	71701	5.0493	-76494	338.03	0.97476
0.044387	0.17322	76763	1.1608	-88070	313.42	0.037433
0.089308	0.36594	73519	4.2885	-86253	297.17	0.29208
0.022287	0.42756	75278	0.96614	-89042	318.18	0.15985
0.095388	0.34357	72469	2.7827	-79633	292.29	0.44212
0.085540	0.28680	77161	4.8944	-76845	295.47	0.054267
0.030070	0.34296	74204	3.0037	-86122	320.10	0.21194
0.073757	0.34116	75516	2.3065	-86545	312.85	0.20374
0.012400	0.69853	78952	1.0513	-77979	313.81	0.053645
0.049545	0.88894	71802	2.4372	-95665	350.77	1.0000
0.055523	0.56841	72127	1.3342	-70381	343.20	0.99349
0.010372	0.048138	77768	2.0350	-98085	312.23	0.0059084
0.022897	0.54397	78291	5.3872	-72130	313.63	0.055650
0.082455	0.60771	77466	5.4955	-94039	294.72	0.073436
0.0010003	0.92977	72268	3.6096	-73746	346.33	0.99929
0.019766	0.68401	74335	5.1606	-68052	324.41	0.40651
0.098617	0.52244	75575	0.21254	-73521	283.66	0.31163
0.050083	0.54124	77238	0.38528	-98097	317.03	0.11798
0.066830	0.98396	76112	2.5537	-85920	313.25	0.35728
0.037905	0.43830	73274	3.4282	-96045	341.00	0.65036
0.041177	0.79364	77152	4.5020	-99058	316.32	0.16182
0.048718	0.45829	71981	0.33683	-76970	345.64	0.96980
0.020808	0.37916	78699	2.8510	-95041	313.36	0.032643
0.099655	0.30141	74199	3.9883	-88433	283.74	0.35575
0.048728	0.38273	78786	1.0029	-75609	313.31	0.039680
0.068797	0.87032	72643	4.2244	-84383	324.75	0.98865
0.015222	0.29203	72342	2.7017	-81931	326.10	0.38898
0.0059082	0.86769	74652	1.9315	-81277	331.26	0.52367
0.099704	0.79487	74958	2.2975	-71550	283.13	0.42023
0.048177	0.41876	71949	2.9194	-98122	356.00	0.99950
0.038604	0.017792	73296	0.69155	-83712	312.48	0.013358
0.0063785	0.53145	71739	2.0289	-93535	354.48	0.99908
0.019122	0.44213	71870	2.4975	-96087	355.51	0.98932
0.076820	0.0022940	76515	2.3981	-95484	312.04	0.0010203
0.0021674	0.36601	77486	4.6911	-87214	313.87	0.048856
0.029091	0.55082	71604	4.2917	-99873	357.58	1.0000
0.092141	0.24980	72292	0.27465	-80922	323.10	0.99065

Generating the Design Data Table

- The *Design Data* table is generated by the *Surrogate Model Training* study.
 - Alternatively, import a text file from another source.
- In the study, specify the number of *Input Points*.
 - This is the number of full finite element simulations used to generate the data.
- The *Input Parameters* are sampled in the parameter space using Latin hypercube sampling.
 - A kind of sparse parametric sweep that covers the input space without requiring an excessive number of finite element computations
- Use a cluster to speed up this computation by choosing the *Distribute model evaluation* option.

Surrogate Model Functions

- The surrogate models are available as functions:
 - Deep neural network (DNN)
 - Polynomial chaos expansion (PCE)
 - Gaussian process (GP)
- The DNN, PCE, and GP functions are trained on the contents of the *Design Data* table.

tubular_reactor_surrogate.mph - COMSOL Multiphysics

File Home Definitions Geometry Sketch Materials Physics Mesh Study Results Developer Colored Selections

Application Builder Model Manager Component 1 Add Component Model

Workspace Model

Parameters Variables Functions Parameter Case Definitions

Build All Import LiveLink Part Libraries Geometry

Add Material Materials

Transport of Diluted Species Physics

Add Physics Add Mathematics

Build Mesh Mesh 1

Compute Study 1 Add Study Study

3D Plot Group 18 Add Plot Group Results

Model Builder

Type filter text

- tubular_reactor_surrogate.mph (root)
 - Global Definitions
 - Parameters 1
 - Deep Neural Network 1 (dnn_1, dnn_2)
 - Default Model Inputs
 - Materials
 - Component 1 (comp1)
 - Definitions
 - Geometry 1
 - Materials
 - Transport of Diluted Species (tds)
 - Heat Transfer in Fluids (ht)
 - Coefficient Form Boundary PDE (cb)
 - Mesh 1
 - Study 1
 - Surrogate Model Training
 - Step 1: Stationary
 - Step 2: Stationary 2
 - Solver Configurations
 - Results

Settings

Deep Neural Network

Plot Create Plot Train Model Continue Training

Label: Deep Neural Network 1

Layers

Type	Settings
Dense	Input, Input features=5, Output features=50, Activation=tanh
Dense	Hidden, Output features=40, Activation=tanh
Dense	Hidden, Output features=30, Activation=tanh
Dense	Hidden, Output features=20, Activation=tanh
Dense	Output, Output features=2, Activation=tanh

Output features: 40

Activation: tanh

Data

Data source: Result table

Result table: Design Data

Ignore NaN/Inf data points

Data Column Settings

Columns	Type	Settings
r0	Argument	Name=x1, Scaling=to01
z0	Argument	Name=x2, Scaling=to01
E	Argument	Name=x3, Scaling=to01
ke	Argument	Name=x4, Scaling=to01
dHrx	Argument	Name=x5, Scaling=to01
comp1.ppb1	Function values	Name=dnn_1, Scaling=to01
comp1.ppb2	Function values	Name=dnn_2, Scaling=to01

Training and Validation

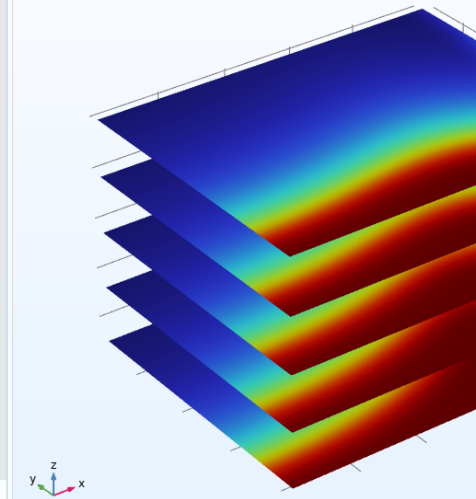
Method: Adam

Learning rate: 1e-3

Weight decay: 0

Graphics

dnn_2(x1, z1)

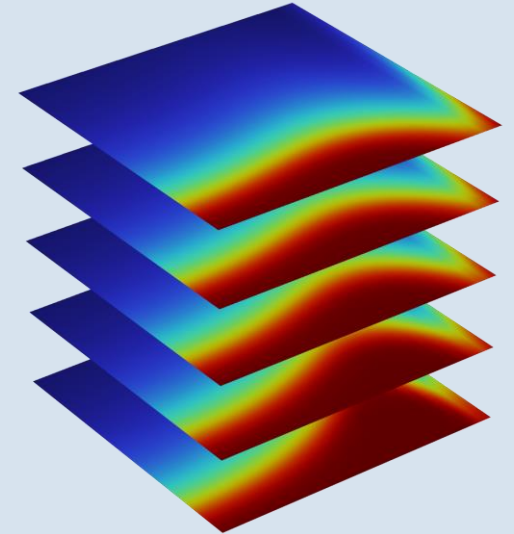


Messages Progress Log Design Data

r0	z0	E	ke	dHrx	comp1.ppb1	comp1.ppb2
0.032140	0.33178	72563	2.6236	-87001	331.64	0.50111
0.055063	0.83791	75719	4.5499	-82537	314.94	0.31657
0.025539	0.60275	75944	3.3028	-88487	319.27	0.18309
0.021129	0.98724	74405	1.4382	-76038	337.06	0.71961
0.076463	0.56690	74356	2.0612	-93668	321.90	0.62580

Surrogate Models

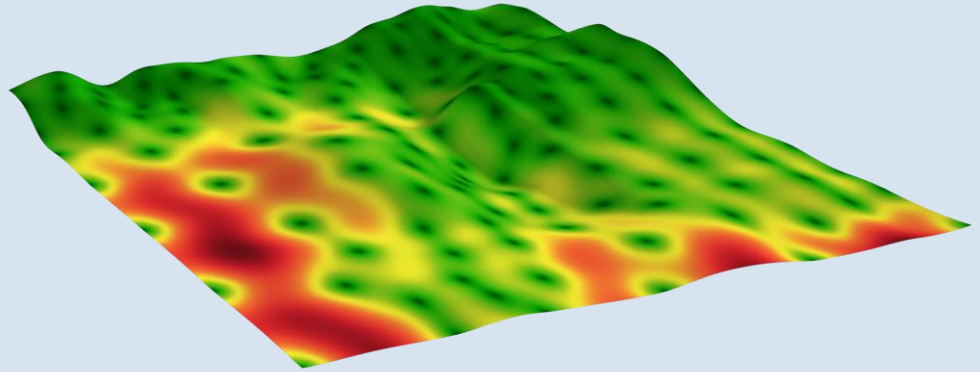
- Deep neural network (DNN):
 - General-purpose surrogate model
 - Can handle data tables with millions of rows
 - Included in COMSOL Multiphysics® without add-on products
- Gaussian process (GP) and polynomial chaos expansion (PCE):
 - Include uncertainty estimates with regard to the quality of the data fit
 - Can handle data tables with up to 2000 rows
 - Require the Uncertainty Quantification Module to be trained
 - Already-trained model can be used without any add-on products
 - For more information on GP and PEC, see the Uncertainty Quantification Module examples and documentation



Visualization of a section of a DNN surrogate model function.

Function Approximation

- Surrogate models are used for multidimensional function approximation.
- Ideal for capturing complex nonlinear relationships within datasets
- Applicable beyond apps and uncertainty quantification, including:
 - Representation of material data
 - Optimization
 - As components in systems modeling
- Supports multiple differentiations with respect to any input parameter



A Gaussian Process surrogate model function, visualizing the standard deviation estimate in color.

Model Builder

Settings

Deep Neural Network

Plot Create Plot Train Model Continue Training

Label: Deep Neural Network 1

Layers

Type	Settings
Dense	Input, Input features=5, Output features=50, Activation=tanh
Dense	Hidden, Output features=40, Activation=tanh
Dense	Hidden, Output features=30, Activation=tanh
Dense	Hidden, Output features=20, Activation=tanh
Dense	Output, Output features=2, Activation=tanh

Output features: 30

Activation: tanh

Data

Data source: Result table

Result table: Design Data

Ignore NaN/Inf data points

Data Column Settings

Columns	Type	Settings
r0	Argument	Name=x1, Scaling=to01
z0	Argument	Name=x2, Scaling=to01
E	Argument	Name=x3, Scaling=to01
ke	Argument	Name=x4, Scaling=to01
dHrx	Argument	Name=x5, Scaling=to01
comp1.ppb1	Function values	Name=dnn1_1, Scaling=to01
comp1.ppb2	Function values	Name=dnn1_2, Scaling=to01

Training and Validation

Plot Parameters

Information

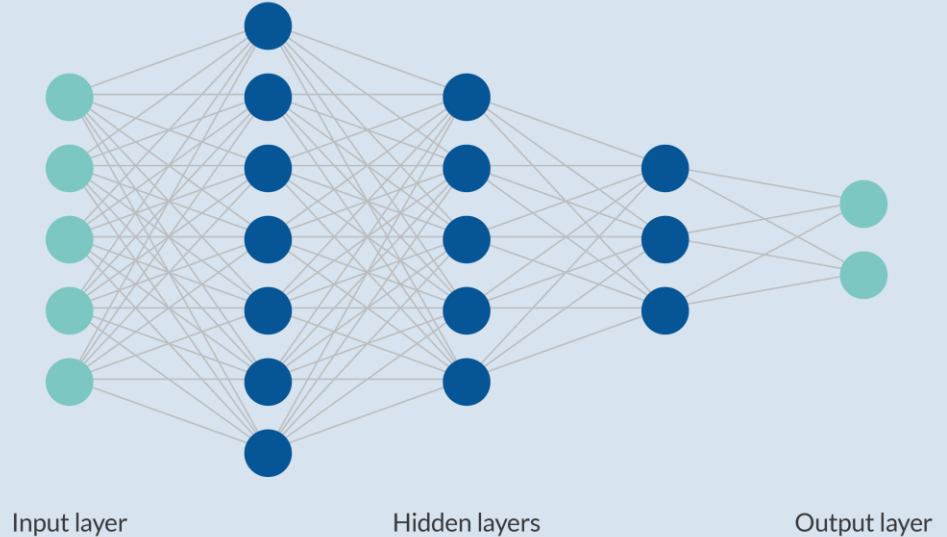
Trained functions: dnn1_1, dnn1_2
 Number of network weights: 4232
 Validation loss: 0.00306045, Epochs=34059
 Training loss: 9.1455E-4, Epochs=47862
 Defined in:
 x1: [8.018396329134703E-6, 0.09998222440481186]
 x2: [2.2087161778472364E-4, 0.9999655485153198]
 x3: [71519.421875, 79204.328125]
 x4: [0.03653548932410797, 5.599172592163086]
 x5: [-101594.640625, -67733.6953125]
 Trained at Oct 31, 2023, 7:07 PM

The Deep Neural Network (DNN) Surrogate Model

- The DNN surrogate model references a *Data Source*, which, in this case, is the *Design Data* table but can also be a text file.
- In the example shown here, the DNN surrogate model defines two functions of five input parameters:
 - $T = f_s(r, z, E, k, dH_{rx})$
 - $x_A = g_s(r, z, E, k, dH_{rx})$
- There is no limit to the number of input or output parameters for a surrogate model.

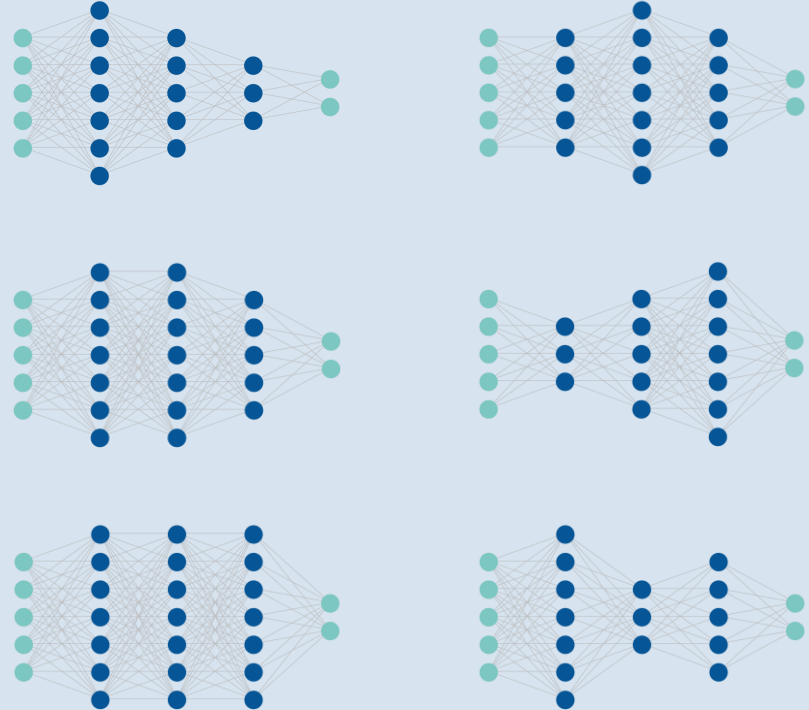
DNN Layers

- 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. The figure shows a graph for a network with three hidden layers, five input nodes, and two output nodes.
- You can define any number of layers and nodes for a DNN surrogate model.



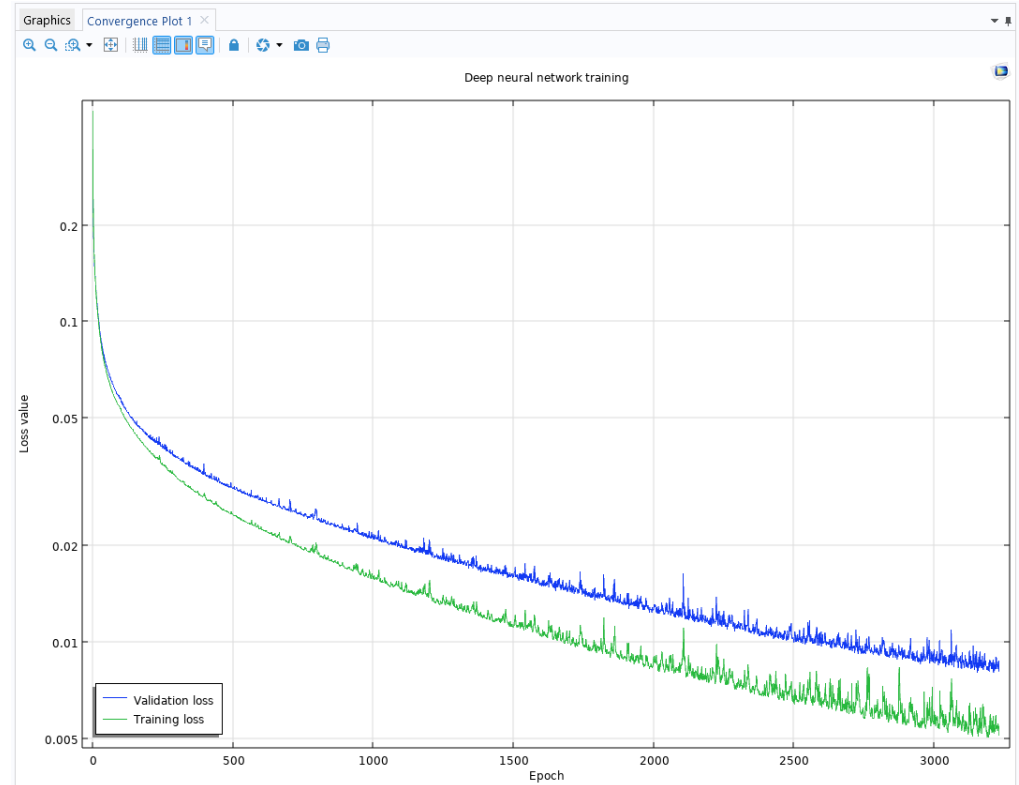
Optimizing Neural Network Architecture

- Selection of layers and nodes is iterative and based on:
 - Problem-specific knowledge
 - Empirical testing
- Balance is key:
 - Too few layers/nodes may lead to underfitting and be inadequate for complex surrogate modeling.
 - Excess layers/nodes can cause overfitting, yielding high accuracy on training data but poor generalization.
- Consider computational efficiency:
 - More layers/nodes increase model evaluation time.
 - Strive for a model that is both accurate and efficient.
 - Quick rule-of-thumb starting point: three hidden layers with 64, 32, and 16 nodes



Training a DNN

- The internal parameters of the neural network are called *weights* and *biases*.
- Training involves optimizing weights and biases to minimize error.
- Objective of training: Align the surrogate model closely with the finite element model.
- Error measurement is calculated via the loss function.
- Different types of loss functions can be used.
- The default loss function is *Root-mean-square error* (RMSE).



Settings

Deep Neural Network

Plot Create Plot Train Model Continue Training

Label: Deep Neural Network 1

Layers

Type	Settings
Dense	Input, Input features=8, Output features=64, Activation=tanh
Dense	Hidden, Output features=64, Activation=tanh
Dense	Hidden, Output features=32, Activation=tanh
Dense	Hidden, Output features=16, Activation=tanh
Dense	Output, Output features=6, Activation=tanh

Output features: 64

Activation: tanh

- Data
 - Linear (none)
 - ReLU
- Data Column
 - ELU
- Training and
 - Sigmoid
 - tanh
- Plot Parameters

Activation Functions

- The activation function determines the output of a neuron, based on the weighted sum of its inputs.
- Available activation functions:
 - Linear
 - ReLU
 - ELU
 - Sigmoid
 - Tanh
- Each layer can have a different activation function.
- Note: The nonlinear activation functions are not necessarily polynomials and can represent general nonlinear behavior. For example, they can represent saturation effects.

Settings

Deep Neural Network

Plot Create Plot Train Model Continue Training

Label: Deep Neural Network 3, Piecewise Linear Approximation

Layers

Type	Settings
Dense	Input, Input features=2, Output features=1000, Activation=ReLU
Dense	Hidden, Output features=1000, Activation=Linear (none)
Dense	Output, Output features=1, Activation=Linear (none)

Data

Data source: Result table

Result table: Table 4

Ignore NaN/Inf data points

Data Column Settings

Columns	Type	Settings
x	Argument	Name=x1, Scaling=to01
y	Argument	Name=x2, Scaling=to01
z	Function values	Name=dnn3_col3, Scaling=to01

Activation Function Summary

- Linear
 - Linear function (“no activation function”), unbounded outputs
- ReLU
 - Piecewise linear, zero for negative values, linear for positive values, unbounded outputs
 - Generates a network that is sometimes easier to train than other activation functions
- ELU
 - Similar to ReLU but smoother and more computationally intensive
- Sigmoid
 - Smoothly nonlinear, s-shaped, bounded outputs between 0 and 1
 - Mostly used for probabilities and classification
- Tanh
 - Smoothly nonlinear, s-shaped, zero-centered and symmetric, bounded outputs between -1 and 1
 - Is the default option

▼ Training and Validation

Method: Adam

Learning rate: 1e-3

Weight decay: 0

Batch size: 512

Loss function: Root-mean-square error

Random seed type: Fixed

Random seed: 0

— Stop condition

Number of epochs: 50000

— Validation data

Validation data: Random sample of data values

Validation data fraction: 0.1

Random seed type: Fixed

Random seed: 0

▶ Plot Parameters

▼ 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 Oct 31, 2023, 7:07 PM

Optimization Solver Settings

- The DNN is trained using a specialized optimization solver.
- The solver settings are called *hyperparameters*. Some of the most important are:
 - *Learning rate*:
 - Controls optimization step size
 - Analogous to numerical damping in nonlinear solvers
 - *Batch size*:
 - Determines subdivision of training data into subsets
 - *Number of epochs*:
 - Indicates total passes through the full dataset

▼ Training and Validation

Method: Adam

Learning rate: 1e-3

Weight decay: 0

Batch size: 512

Loss function: Root-mean-square error

Random seed type: Fixed

Random seed: 0

— Stop condition

Number of epochs: 50000

— Validation data

Validation data: Random sample of data values

Validation data fraction: 0.1

Random seed type: Fixed

Random seed: 0

▶ Plot Parameters

▼ 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 Oct 31, 2023, 7:07 PM

Hyperparameter Guidance

- *Learning rate:*
 - Too low: may get trapped in local minima
 - Too high: can overshoot the minimum, leading to poor convergence
- *Batch size:*
 - Small size: can cause noisy gradients and longer training
 - Large size: may result in poor generalization and inefficient computation
- *Number of epochs:*
 - Insufficient epochs: risk of underfitting
 - Excessive epochs: potential for overfitting

▼ Training and Validation

Method: Adam

Learning rate: 1e-3

Weight decay: 0

Batch size: 512

Loss function: Root-mean-square error

Random seed type: Fixed

Random seed: 0

— Stop condition

Number of epochs: 50000

— Validation data

Validation data: Random sample of data values

Validation data fraction: 0.1

Random seed type: Fixed

Random seed: 0

▶ Plot Parameters

▼ 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 Oct 31, 2023, 7:07 PM

Training and Validation Loss

- *Training loss:*
 - Reflects model performance on the primary training data
- *Validation loss:*
 - Indicates model performance on a separate, unseen subset of data:
 - The solver sets aside a portion of the data and considers it unseen data for validation purposes.

▼ Training and Validation

Method: Adam

Learning rate: 1e-3

Weight decay: 0

Batch size: 512

Loss function: Root-mean-square error

Random seed type: Fixed

Random seed: 0

— Stop condition

Number of epochs: 50000

— Validation data

Validation data: Random sample of data values

Validation data fraction: 0.1

Random seed type: Fixed

Random seed: 0

► Plot Parameters

▼ 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 Oct 31, 2023, 7:07 PM

Understanding Loss Metrics

- *Training loss:*
 - Decreases as the model learns — beware of potential overfitting
- *Validation loss:*
 - Helps estimate the model's ability to generalize to new data
- *Monitoring overfitting:*
 - A sign of overfitting: Validation loss increases while training loss decreases.
- *Hyperparameter tuning:*
 - Aim to balance and minimize both losses.
 - Ensures the model is learning effectively and can generalize well:
 - Generalize well = applies well to unseen data

EXAMPLE Thermal Actuator

- The thermal actuator app and surrogate model demonstrate using the efficient geometry sampling method to generate a surrogate model of a parametric CAD model.
- The surrogate model is based on a design data table (text file) with several millions of rows.
- The surrogate model defines six functions, each with five input arguments.

[Download the application](#)

The screenshot displays the COMSOL Multiphysics interface for a surrogate model of a thermal actuator. The title bar indicates the file name is 'thermal_actuator_surrogate.mph - COMSOL Multiphysics'. The ribbon menu includes tabs for File, Home, Definitions, Geometry, Materials, Physics, Mesh, Study, Results, and Developer. The 'Model Builder' tree on the left shows the model structure, including Global Definitions, Parameters, a Deep Neural Network (DNN) component, Materials, Thermal Actuator (comp 1), and Study 1. The 'Settings' panel for the DNN is active, showing training and validation parameters. The 'Temperature (DNN)' panel on the right displays a 3D visualization of the thermal actuator with a color gradient representing temperature distribution.

Model Builder

- thermal_actuator_surrogate.mph (root)
 - Global Definitions
 - Parameters 1
 - Parameters 2
 - Deep Neural Network 1 (dnn1_V, dnn1_T, ...)
 - Default Model Inputs
 - Materials
 - Thermal Actuator (comp 1)
 - Definitions
 - Geometry 1
 - Materials
 - Electric Currents (ec)
 - Heat Transfer in Solids (ht)
 - Solid Mechanics (solid)
 - Multiphysics
 - Electromagnetic Heating 1 (emh1)
 - Mesh 1
 - Study 1
 - Surrogate Model Training
 - Step 1: Stationary
 - Solver Configurations
 - Job Configurations
 - Results

Settings

Deep Neural Network

Plot Create Plot Train Model Continue Training

Label: Deep Neural Network 1

> Layers

> Data

> Data Column Settings

> Training and Validation

Method: Adam

Learning rate: 1e-3

Weight decay: 0

Batch size: 4096

Loss function: Root-mean-square error

Random seed type: Fixed

Random seed: 0

Train on GPU

— Stop condition

Number of epochs: 1000

— Validation data

Validation data: Random sample of data values

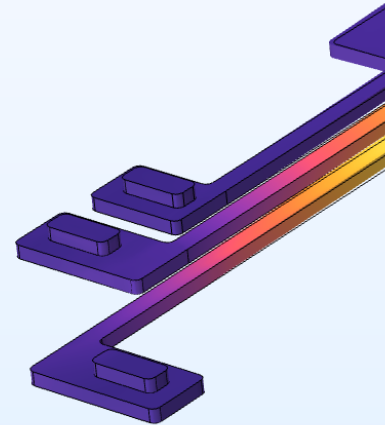
Validation data fraction: 0.1

Random seed type: Fixed

Random seed: 0

> Plot Parameters

> Information



EXAMPLE Tubular Reactor

- Surrogate model of a tubular reactor simulating an elementary, exothermic, irreversible reaction
- The app has three scalar input parameters: the activation energy, the thermal conductivity, and the heat of reaction
- The resulting surrogate model consists of two functions: $T(r,z,E,k,dH_{rx})$ and $x_A(r,z,E,k,dH_{rx})$.

[Download the application](#)

The screenshot displays the COMSOL Multiphysics software interface for a Surrogate Model Training application. The main window title is "tubular_reactor_surrogate.mph - COMSOL Multiphysics". The interface is divided into several panels:

- Model Builder:** Shows the project hierarchy for "tubular_reactor_surrogate.mph (root)". It includes sections for Global Definitions (Parameters 1, Deep Neural Network 1, Default Model Inputs), Materials (Component 1), and Study 1 (Surrogate Model Training, Step 1: Stationary, Step 2: Stationary 2, Solver Configurations, Job Configurations, Results).
- Settings:** Configured for "Surrogate Model Training".
 - Label: Surrogate Model Training
 - Compute action: Compute and build surrogate model
 - Output table group: Design of Experiments
 - Surrogate model: Design of experiments (No surrogate model)
 - Quantities of Interest:

Expression	Include study-dependent input	Study-dependent input description
T	Configure study-dependent input	
x _A	Configure study-dependent input	
 - Geometry sampling: Geometry Sampling 1
 - File: Embedded file
 - Output table: Design Data (QoI1-2)
 - Input Parameters:

Parameter	Source type	Parameter description
E (Activation energy)	Analytic	Uniform from [71518, 79205]
ke (Thermal conductivity)	Analytic	Uniform from [0.0559, 5.6]
dH _{rx} (Heat of reaction)	Analytic	Uniform from [-101600, -67733]

- Conversion, species A:** A 3D visualization of a vertical tubular reactor. The reactor is filled with a fluid, and the color gradient represents the conversion of species A, ranging from blue (low conversion) to red (high conversion). The conversion is highest at the top of the reactor and lowest at the bottom.

EXAMPLE

Battery Rate Capability

- This app demonstrates the usage of a DNN function for predicting the rate capability of an NMC/graphite battery cell.
- Three input data values can be set: the thickness of the positive electrode, the active material volume fraction of the positive electrode, and the active material volume fraction of the negative electrode.

[Download the application](#)

The screenshot displays the COMSOL Multiphysics software interface for a project named "lib_rate_capability_surrogate.mph". The Model Builder tree on the left shows the model structure, including Global Definitions, Component 1 (comp 1), and a Deep Neural Network (dnn_E_vol, dnn_P_volAve). The Settings panel for the DNN is visible, showing a layer configuration of [4, 10, 10, 10, 2] and training parameters such as Adam method, 1e-4 learning rate, and Root-mean-square error loss function. The Ragone Plots graph on the right shows Volumetric Energy (Wh/dm³) on the y-axis (ranging from 20 to 1000) versus an unlabeled x-axis (ranging from 2 to 10). The plot shows a single data series labeled "Surrogate Model" with red square markers connected by a dashed red line, indicating a decreasing trend in volumetric energy as the x-axis value increases.

EXAMPLE Microstrip Patch Antenna

- A DNN-trained surrogate model estimates the performance of a microstrip patch antenna, depending on design parameters.
- The results can be optionally compared to those provided with a full finite element model.

Untitled.mph - Microstrip Patch Antenna

Fichier Home

Reset Compute Report Help

Input Simulation Documentation

Input and Description

Input

Visualization

Frequency (f0): 1.575 GHz

Design

Need to recompute full model when changed

Patch length (L_patch): 52 mm

Tuning stub length (L_sub): 15.5 mm

Dielectric constant (ε_r): 3.38

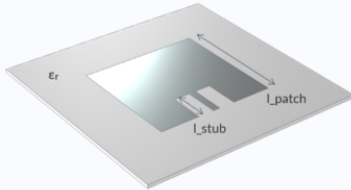
View Configuration

Stacked view

Tiled view

Show patch frame

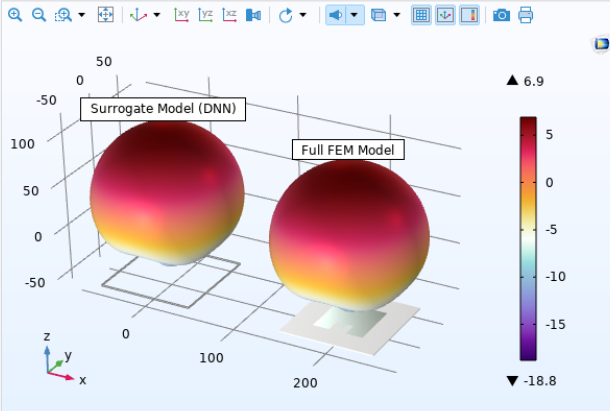
Antenna Description



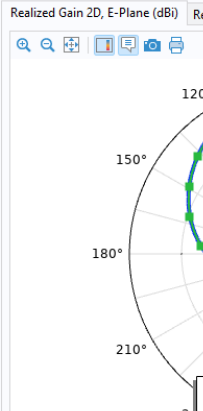
Information

Result Analysis

Realized Gain 3D (dBi)

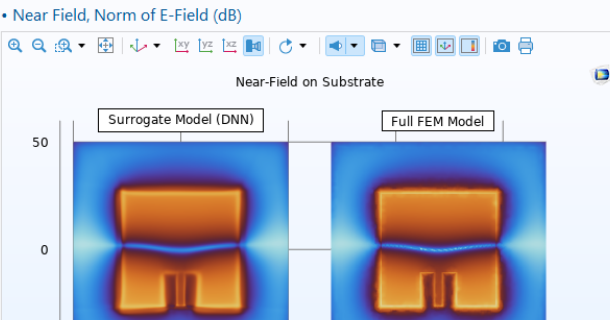


Realized Gain 2D (dBi)

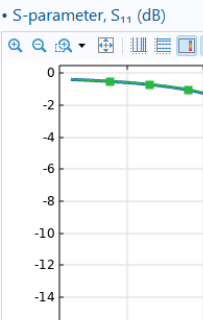


Near Field, Norm of E-Field (dB)

Near-Field on Substrate



S-parameter, S₁₁ (dB)



COMSOL MULTIPHYSICS®

The platform product for simulating real-world designs, devices, and processes. One user interface for all engineering applications.

- **MODEL BUILDER:** Combine physics phenomena in one model
- **APPLICATION BUILDER:** Build simulation apps from models
- **MODEL MANAGER:** Collaborate and organize models and apps

COMSOL Compiler™

Compile simulation apps into executable files. Run them freely on any computer.

COMSOL Server™

Host and administrate your simulation apps. Run them through a web interface.

ADD-ON PRODUCTS

ELECTROMAGNETICS

- AC/DC Module
- RF Module
- Wave Optics Module
- Ray Optics Module
- Plasma Module
- Electric Discharge Module
- Semiconductor Module

FLUID & HEAT

- CFD Module
 - Mixer Module
- Polymer Flow Module
- Microfluidics Module
- Porous Media Flow Module
- Subsurface Flow Module
- Pipe Flow Module
- Molecular Flow Module
- Metal Processing Module
- Heat Transfer Module

STRUCTURAL & ACOUSTICS

- Structural Mechanics Module
 - Nonlinear Structural Materials Module
 - Composite Materials Module
 - Geomechanics Module
 - Fatigue Module
 - Rotordynamics Module
- Multibody Dynamics Module
- MEMS Module
- Acoustics Module

CHEMICAL

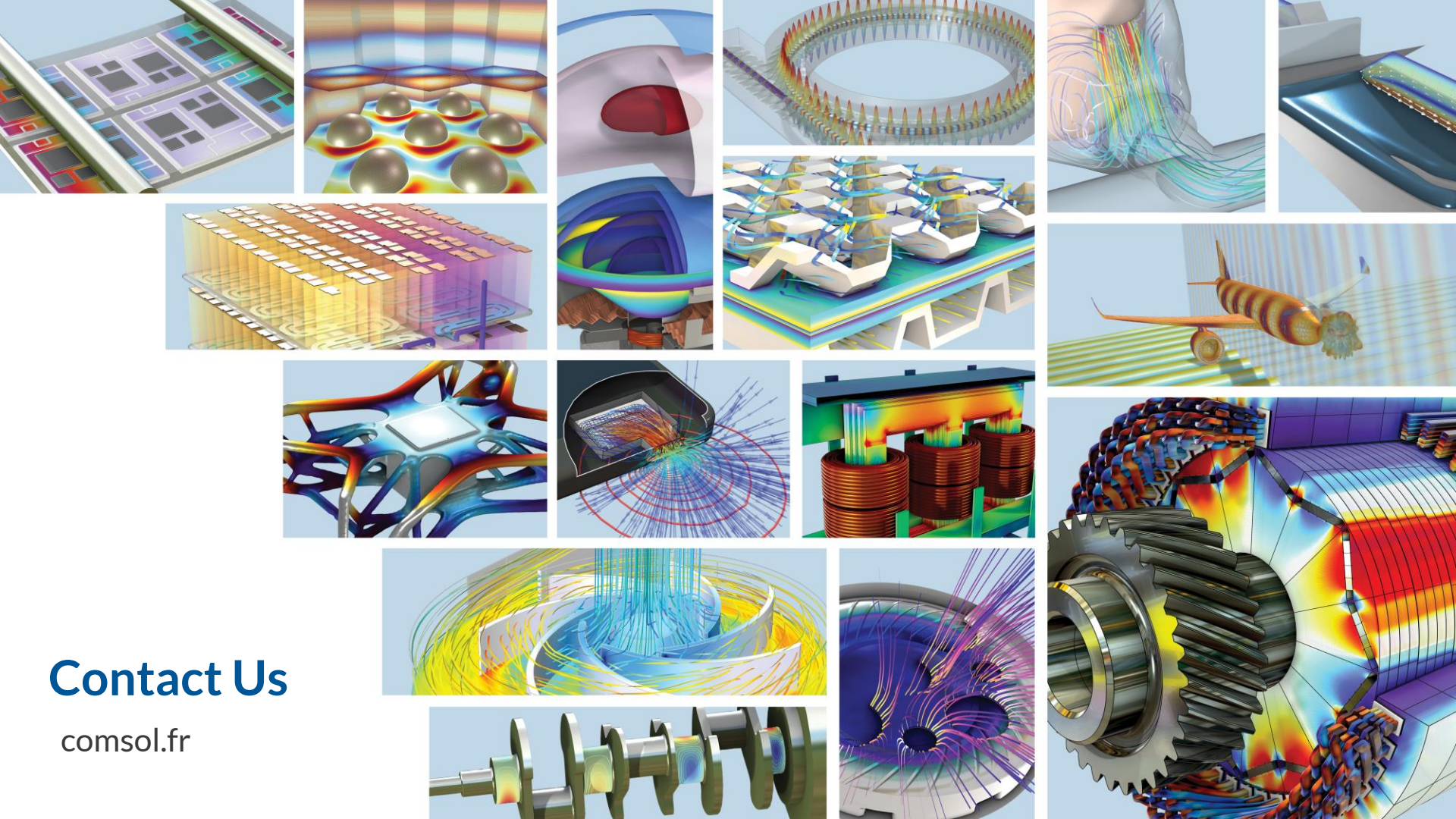
- Chemical Reaction Engineering Module
- Battery Design Module
- Fuel Cell & Electrolyzer Module
- Electrodeposition Module
- Corrosion Module
- Electrochemistry Module

MULTIPURPOSE

- Optimization Module
- Uncertainty Quantification Module
- Material Library
- Particle Tracing Module
- Liquid & Gas Properties Module

INTERFACING

- LiveLink™ for MATLAB®
- LiveLink™ for Simulink®
- LiveLink™ for Excel®
- CAD Import Module
- Design Module
- ECAD Import Module
- LiveLink™ for SOLIDWORKS®
- LiveLink™ for Inventor®
- LiveLink™ for AutoCAD®
- LiveLink™ for Revit®
- LiveLink™ for PTC Creo Parametric™
- LiveLink™ for Solid Edge®
- File Import for CATIA® V5



Contact Us

comsol.fr