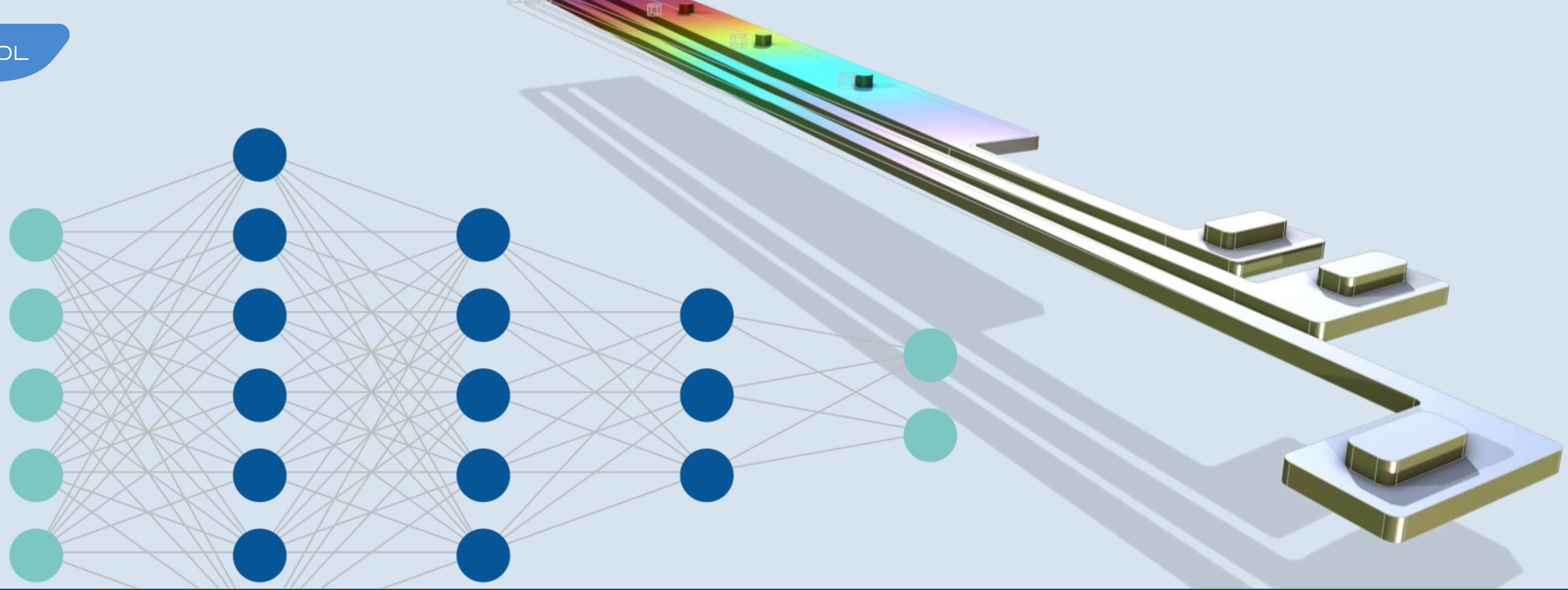


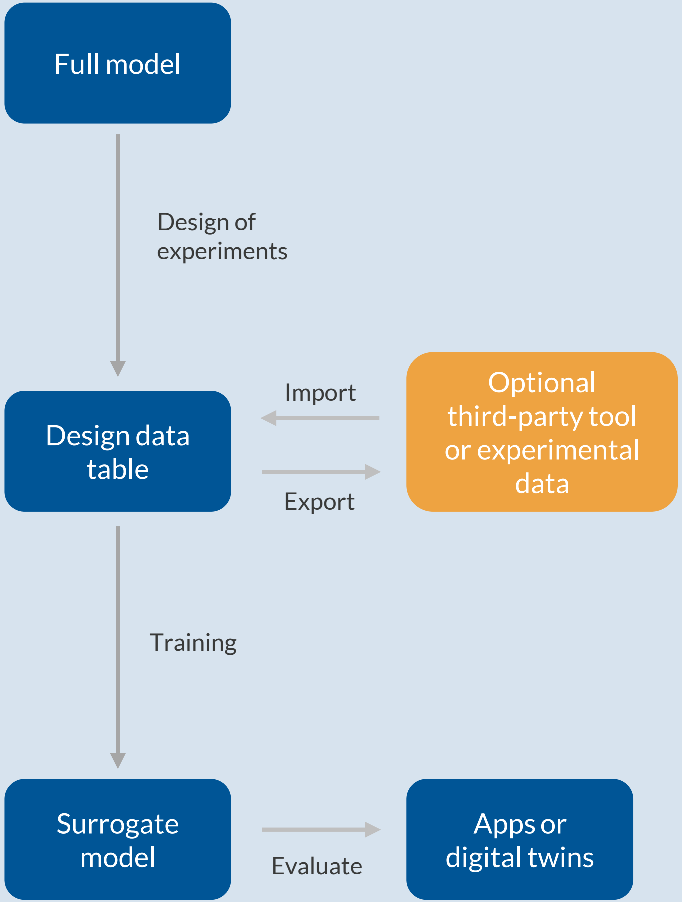
Surrogate Models

in COMSOL Multiphysics

Loïc Renversade
Applications
COMSOL France

Surrogate Models

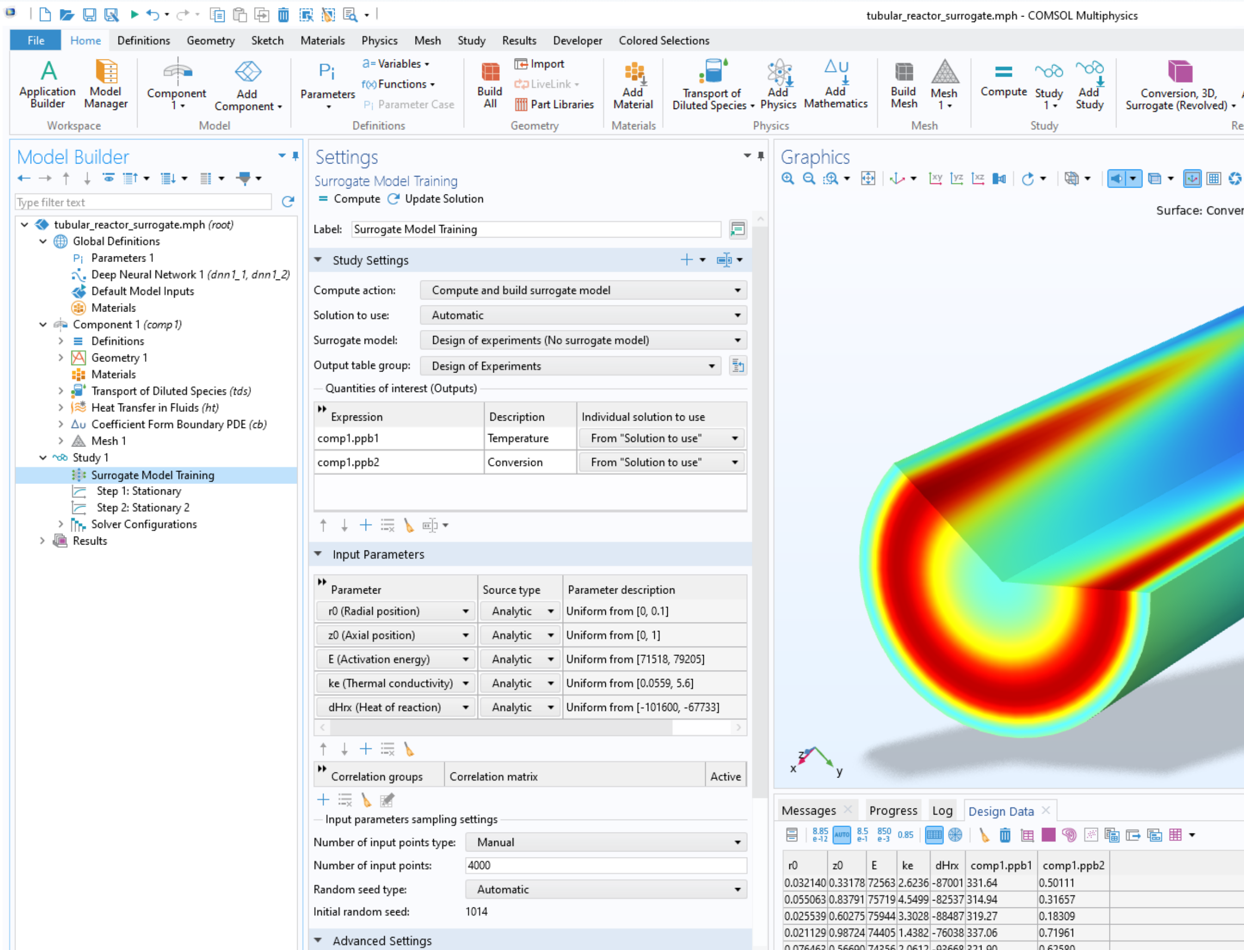
- Powerful new functionality for creating and using data-driven surrogate models in version 6.2
- What is a surrogate model?
  - A simpler, less computationally expensive model
  - Approximates the behavior of more complex, computationally intensive models
- Benefits of surrogate models:
  - Faster model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



COMSOL

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.)



Generating the Design Data Table

Inputs | Outputs

Design data

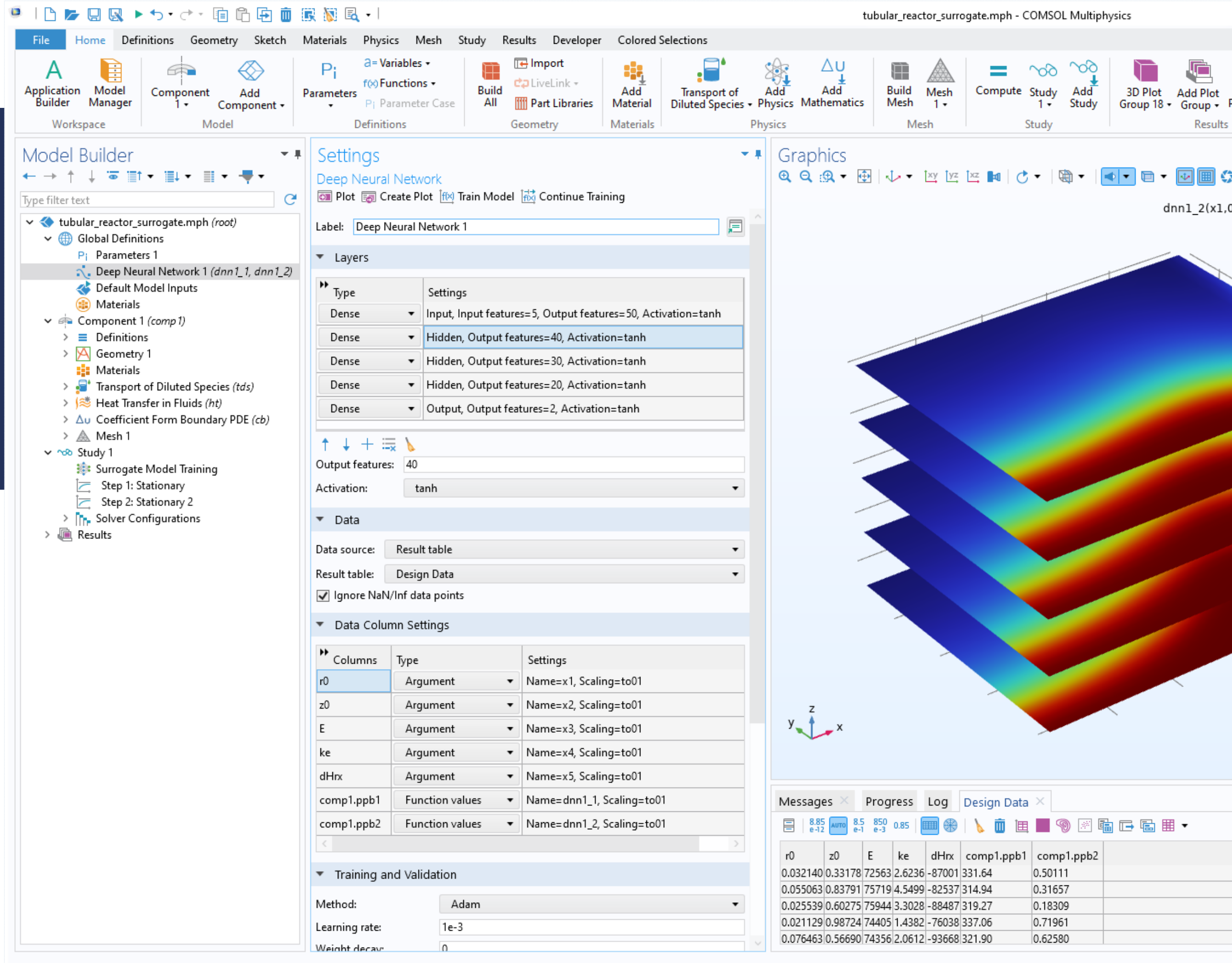
r0	z0	E	ke	dHrx	T	xA
0.032140	0.33178	72563	2.6236	-87001	331.64	0.50111
0.055063	0.83791	75719	4.5499	-82537	314.94	0.31657
0.025539	0.60275	75944	3.3028	-88487	319.27	0.18309
0.021129	0.98724	74405	1.4382	-76038	337.06	0.71961
0.076463	0.56690	74356	2.0612	-93668	321.90	0.62580
0.055536	0.85834	77315	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	0.0055084
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

- 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.

COMSOL

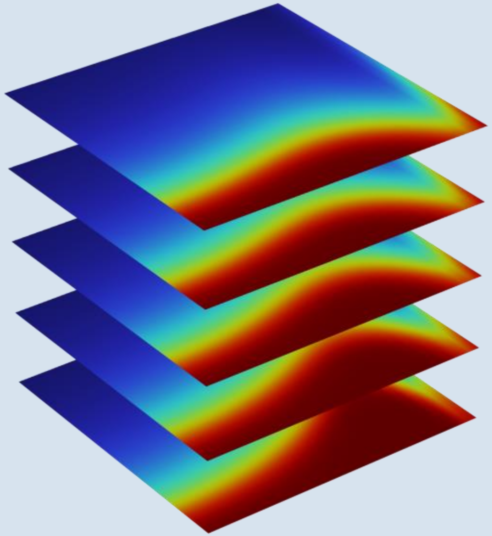
Surrogate Model Functions

- 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.



Surrogate Models

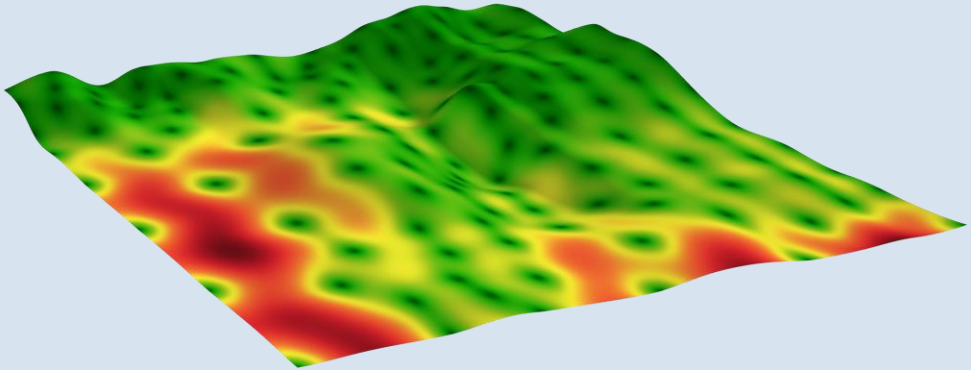
- Deep neural network (DNN):
  - General-purpose surrogate model
  - Can handle data tables with millions of rows
  - Included in COMSOL Multiphysics® without add-on products
- Gaussian process (GP) and polynomial chaos expansion (PCE):
  - Include uncertainty estimates with regard to the quality of 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.

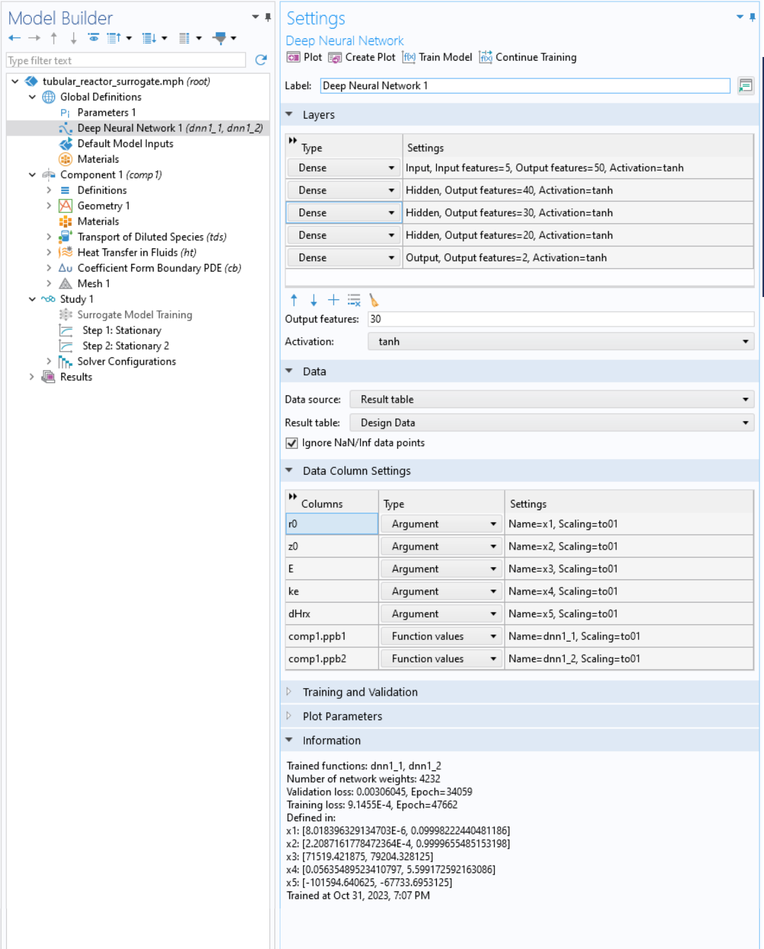
Function Approximation

- 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.

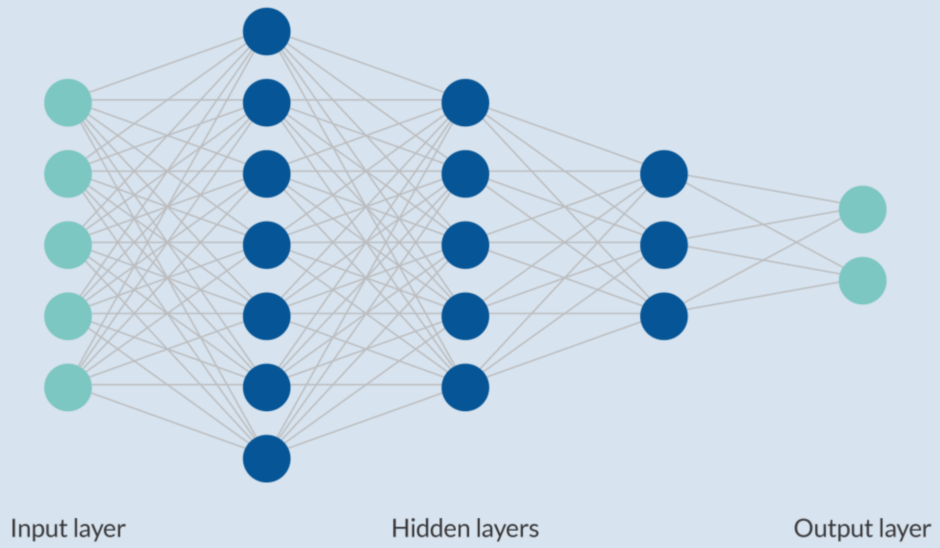
The Deep Neural Network (DNN) Surrogate Model



- 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.

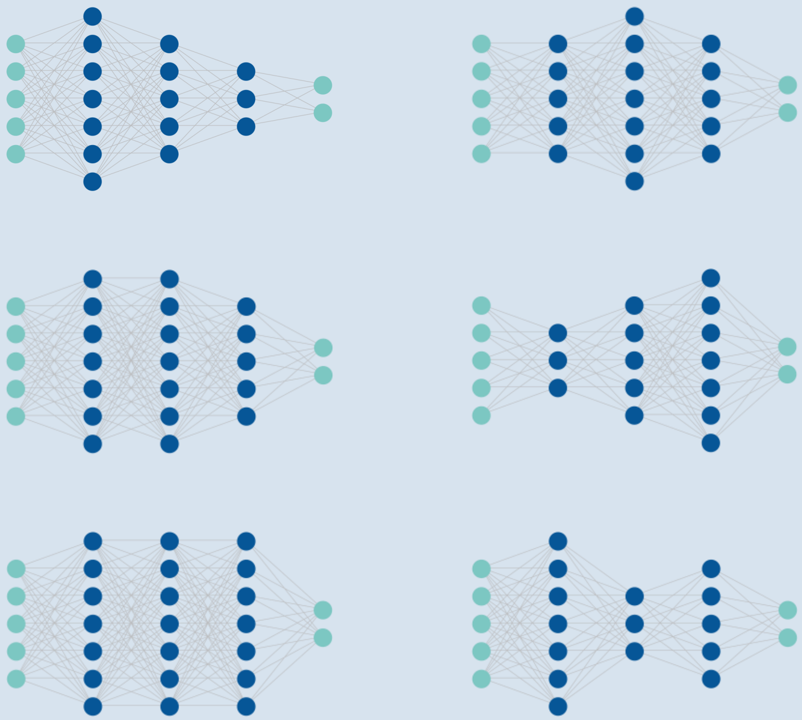
DNN Layers

- 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.



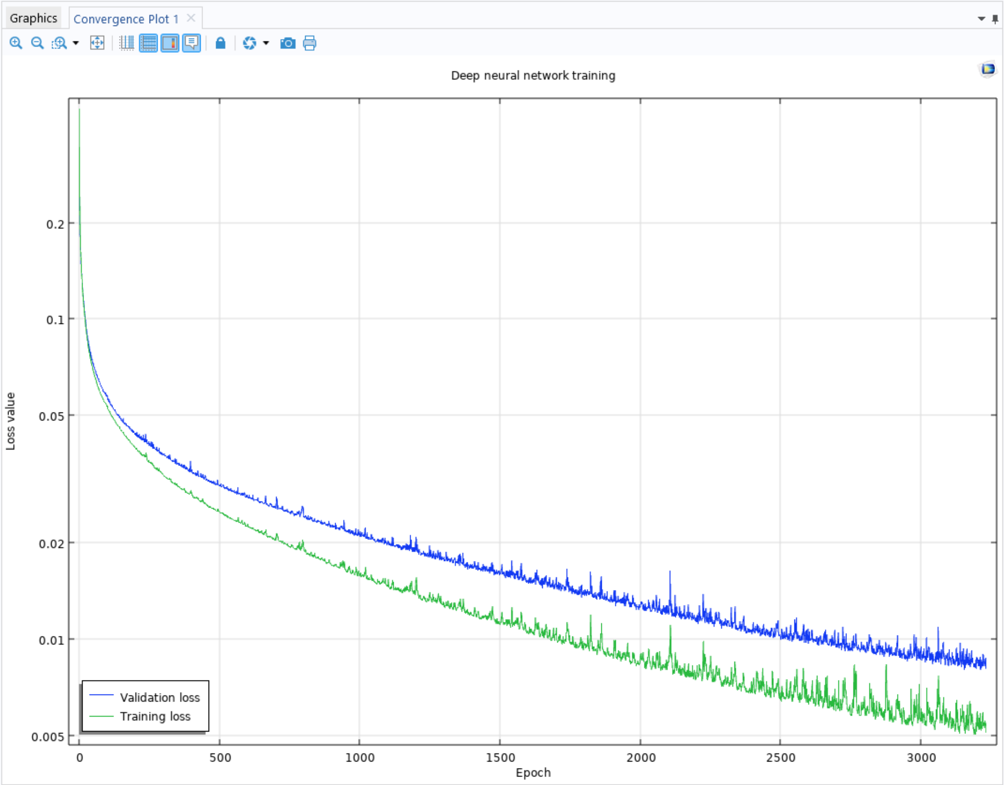
Optimizing Neural Network Architecture

- 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



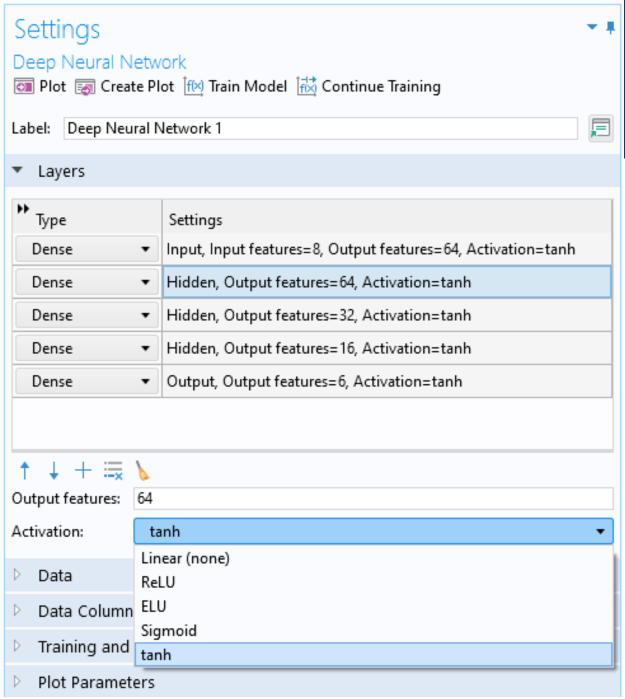
Training a DNN

- 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).



COMSOL

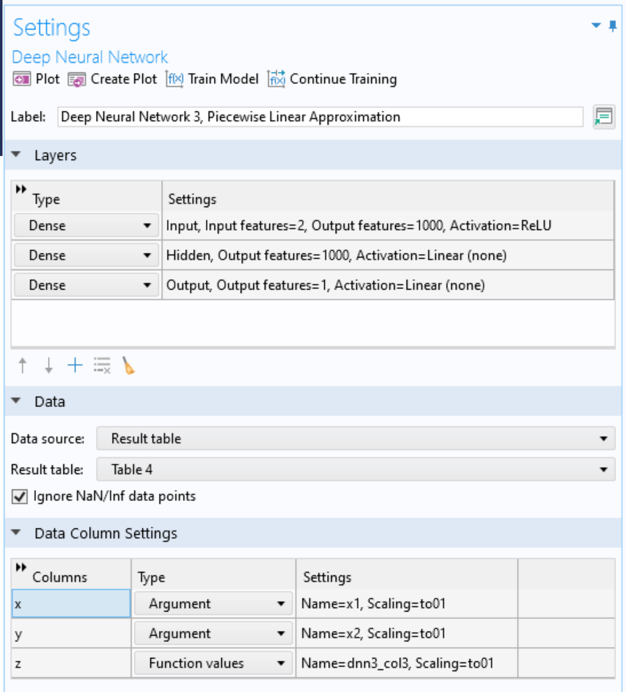
Activation Functions



- 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.

ReLU = rectified linear unit   ELU = exponential linear unit

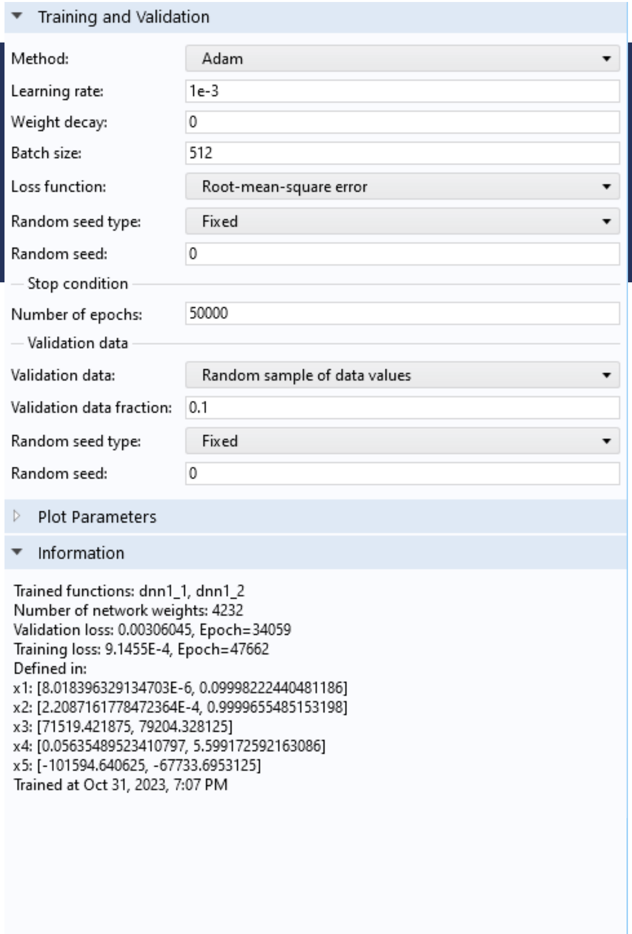
Activation Function Summary



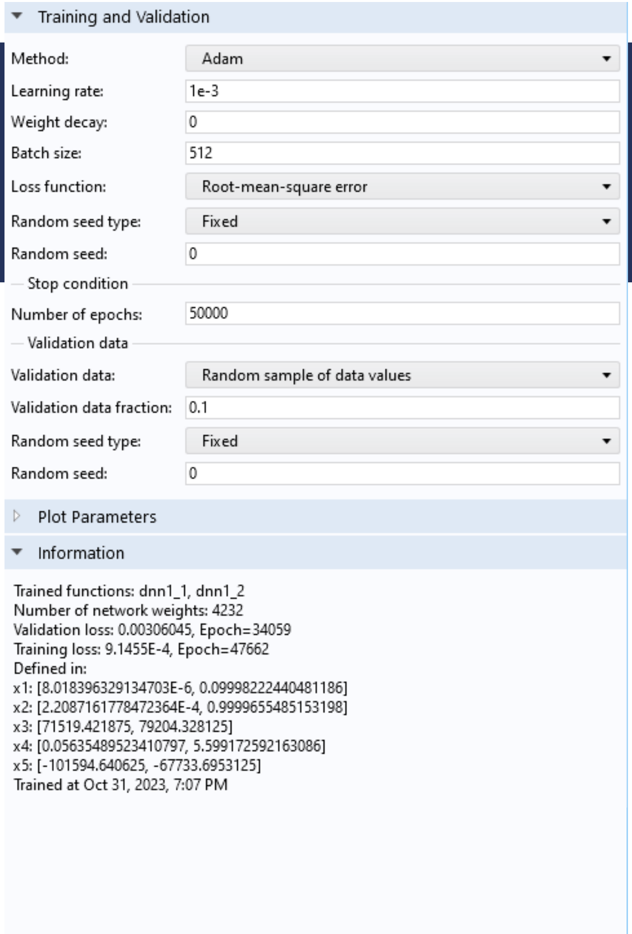
- Linear
  - Linear function (“no activation function”), unbounded outputs
- ReLU
  - Piecewise linear, zero for negative values, linear for positive values, unbounded outputs
  - Generates a network that is sometimes easier to train than other activation functions
- ELU
  - Similar to ReLU but smoother 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

ReLU = rectified linear unit ELU = exponential linear unit

Optimization Solver Settings

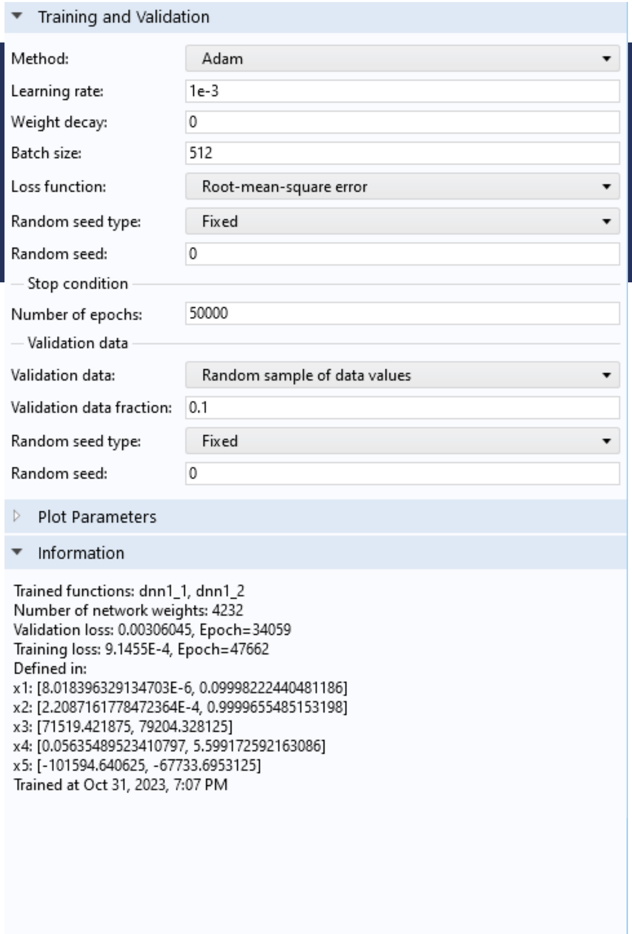


- The DNN is trained using a specialized optimization solver.
- The solver settings are called *hyperparameters*. Some of the most important are:
  - *Learning rate*:
    - Controls optimization step size
    - Analogous to numerical damping in nonlinear solvers
  - *Batch size*:
    - Determines subdivision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



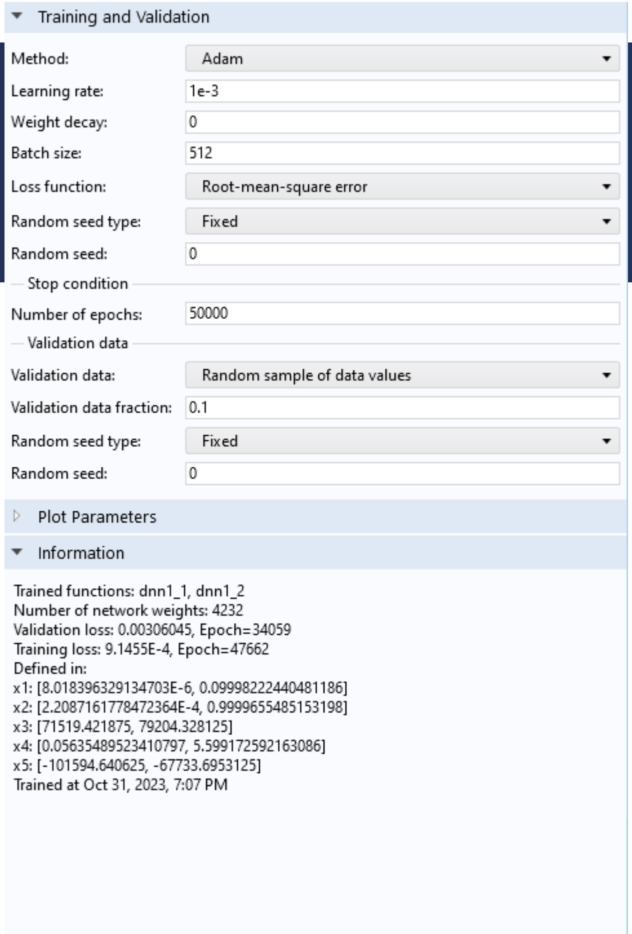
Hyperparameter Guidance

- *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 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.



Understanding Loss Metrics

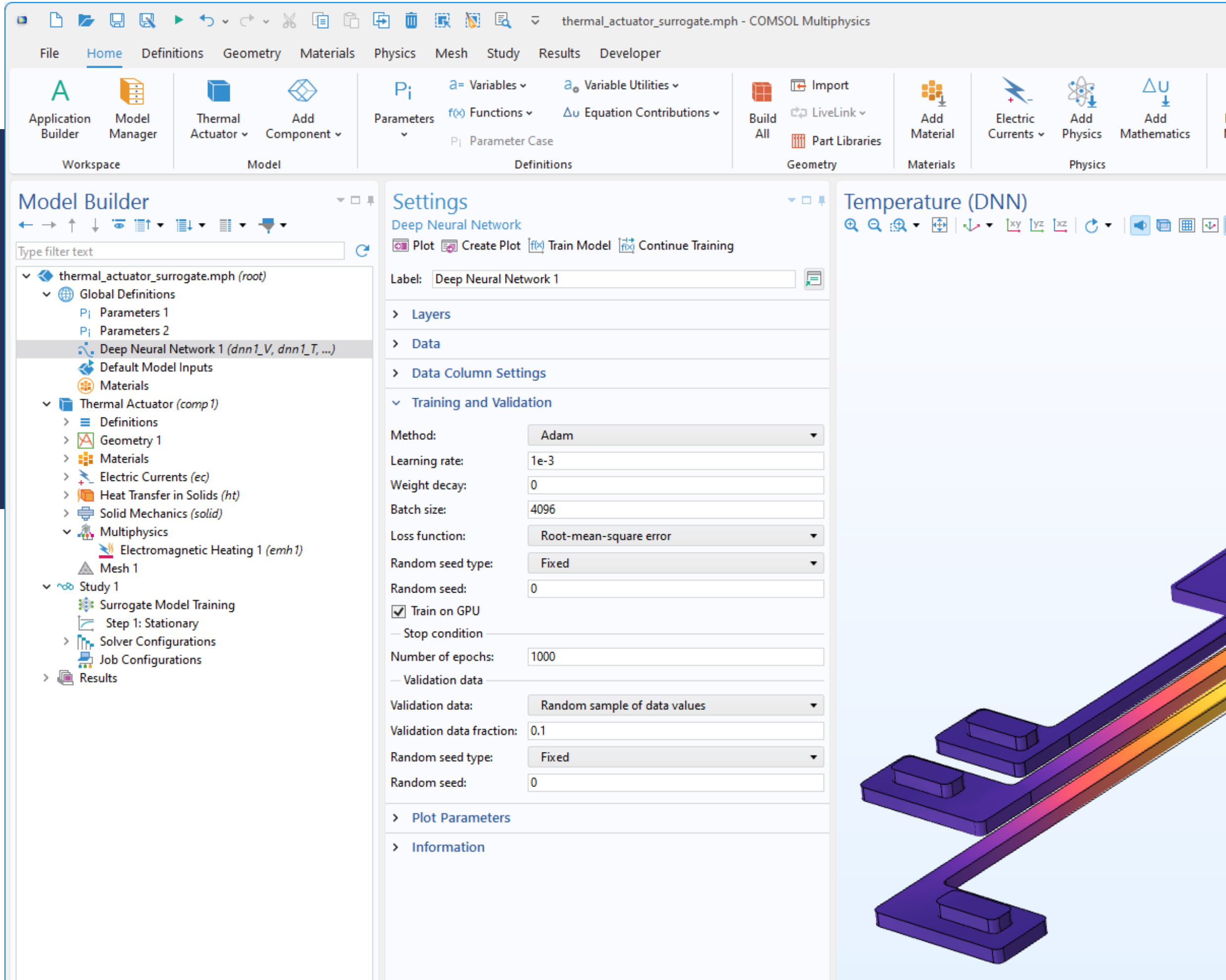
- *Training loss*:
  - Decreases as the model learns — beware of potential overfitting
- *Validation loss*:
  - Helps estimate the model's ability to generalize to new data
- Monitoring overfitting:
  - A sign of overfitting: Validation loss increases while training loss decreases.
- Hyperparameter tuning:
  - 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]

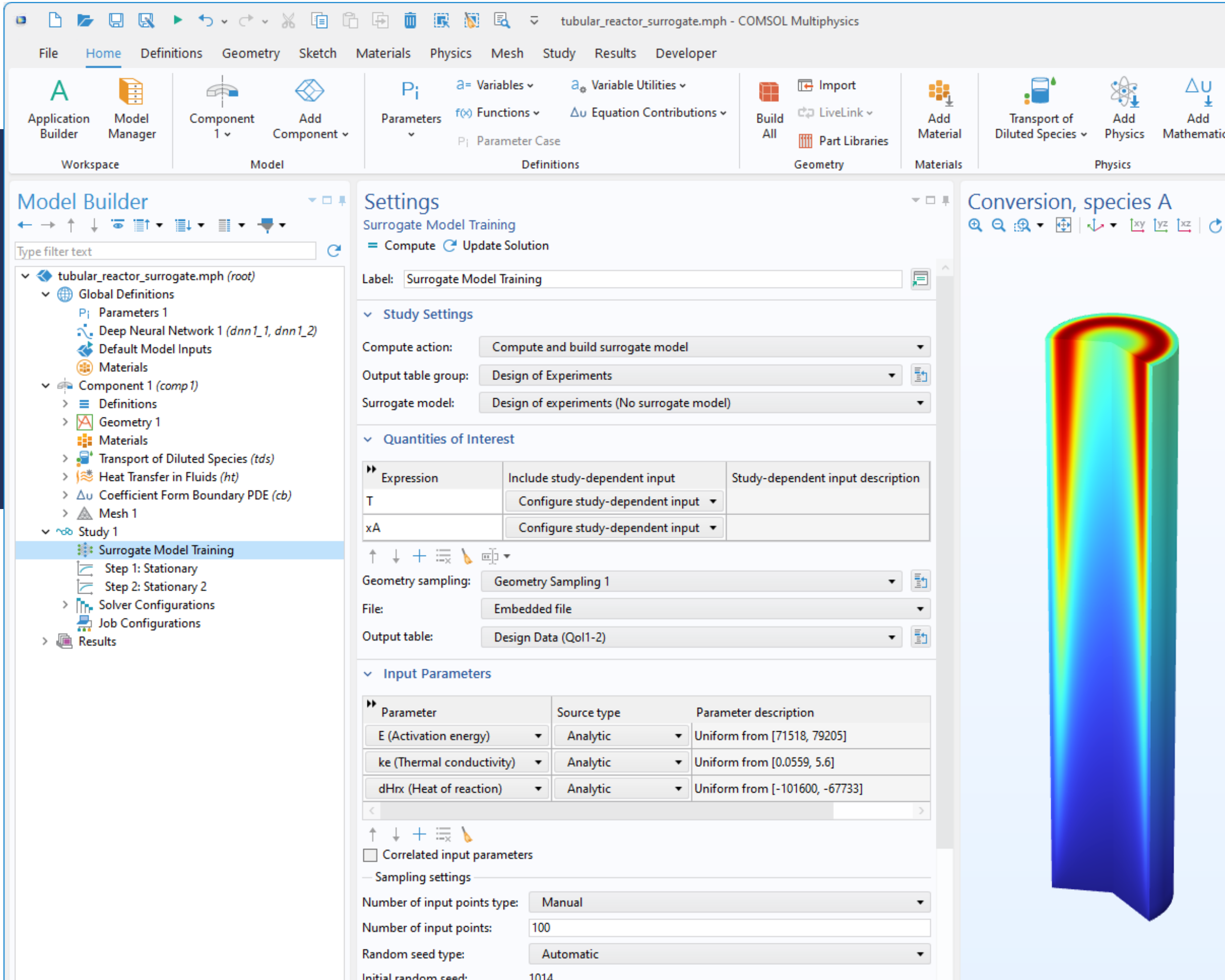


EXAMPLE

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 $x_A(r,z,E,k,dH_{rx})$.

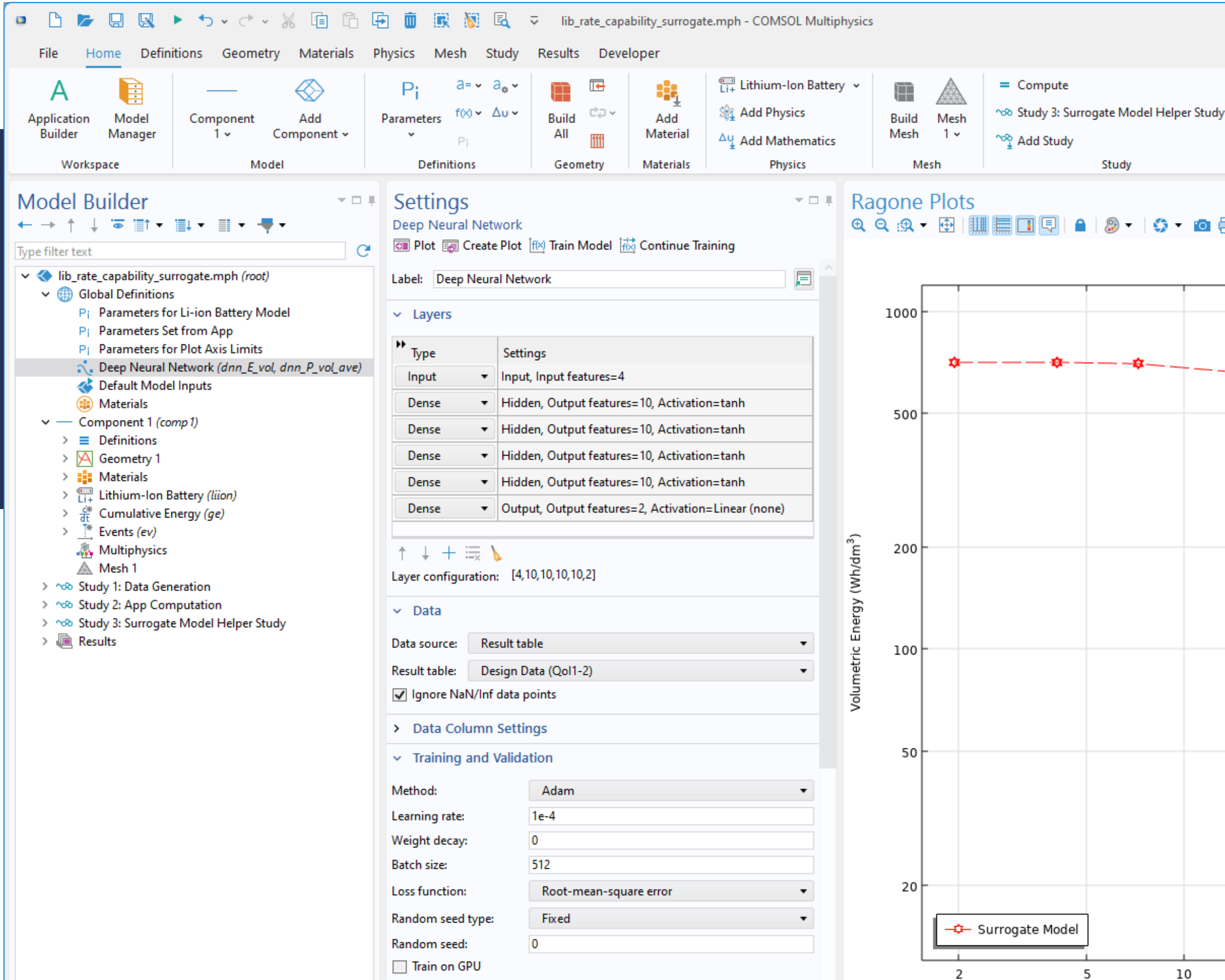
[Download the application]



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.



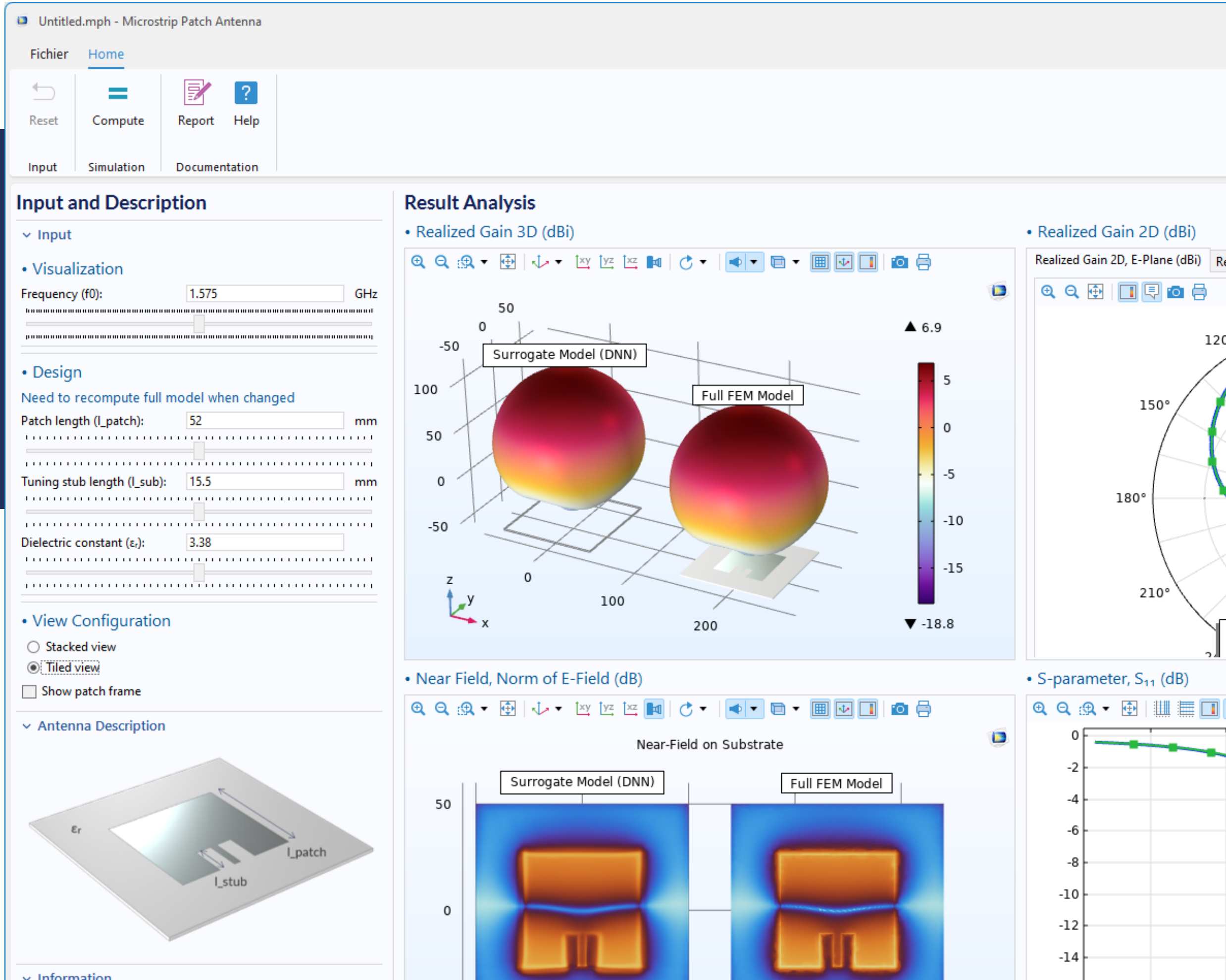
[Download the application](#)

COMSOL

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.



COMSOL

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

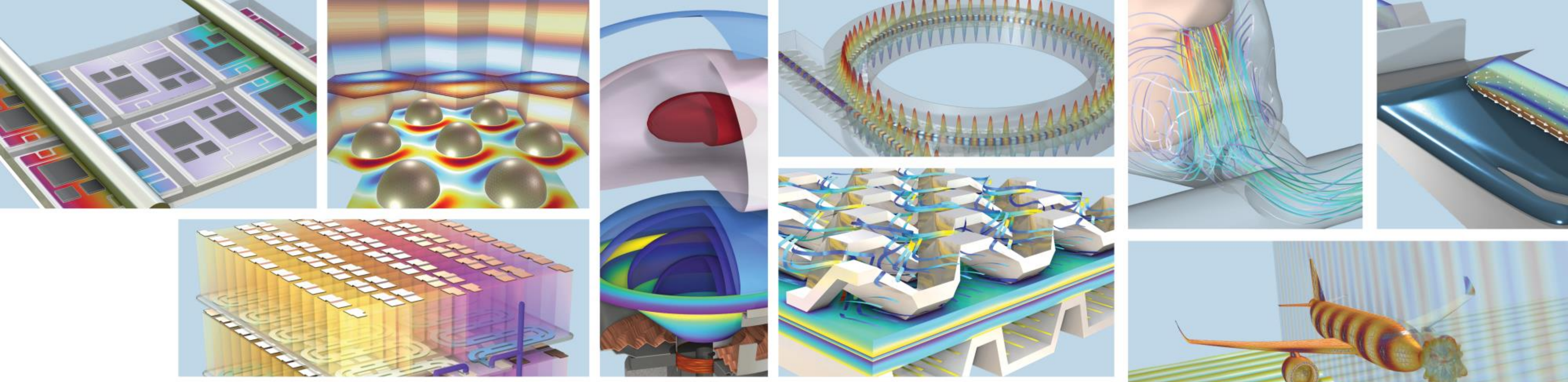
- Chemical Reaction Engineering Module
- Battery Design Module
- Fuel Cell & Electrolyzer Module
- Electrodeposition Module
- Corrosion Module
- Electrochemistry Module

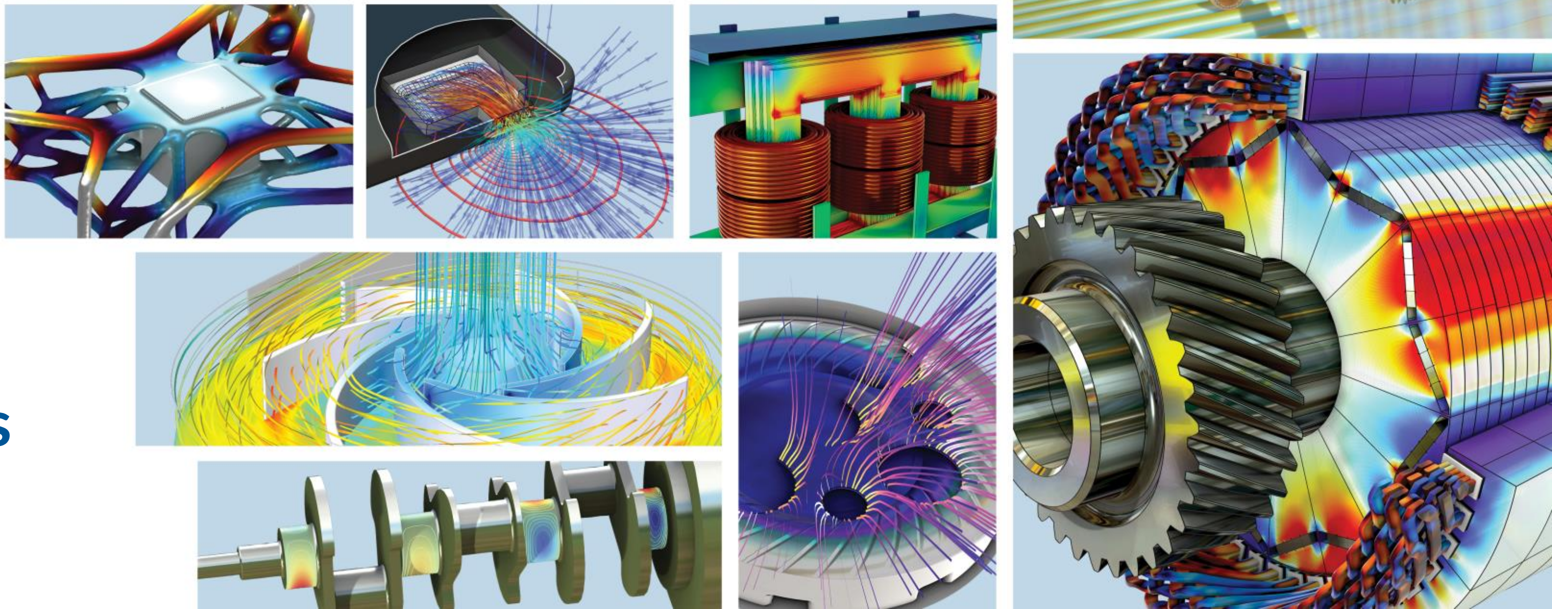
MULTIPURPOSE

- 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