Altair

HyperWorks

# Altair Flux™ 2019.1 User Guide

New Features

altairhyperworks.com

# Contents

# Introduction of new features of Flux 2019.1

1

This chapter covers the following:

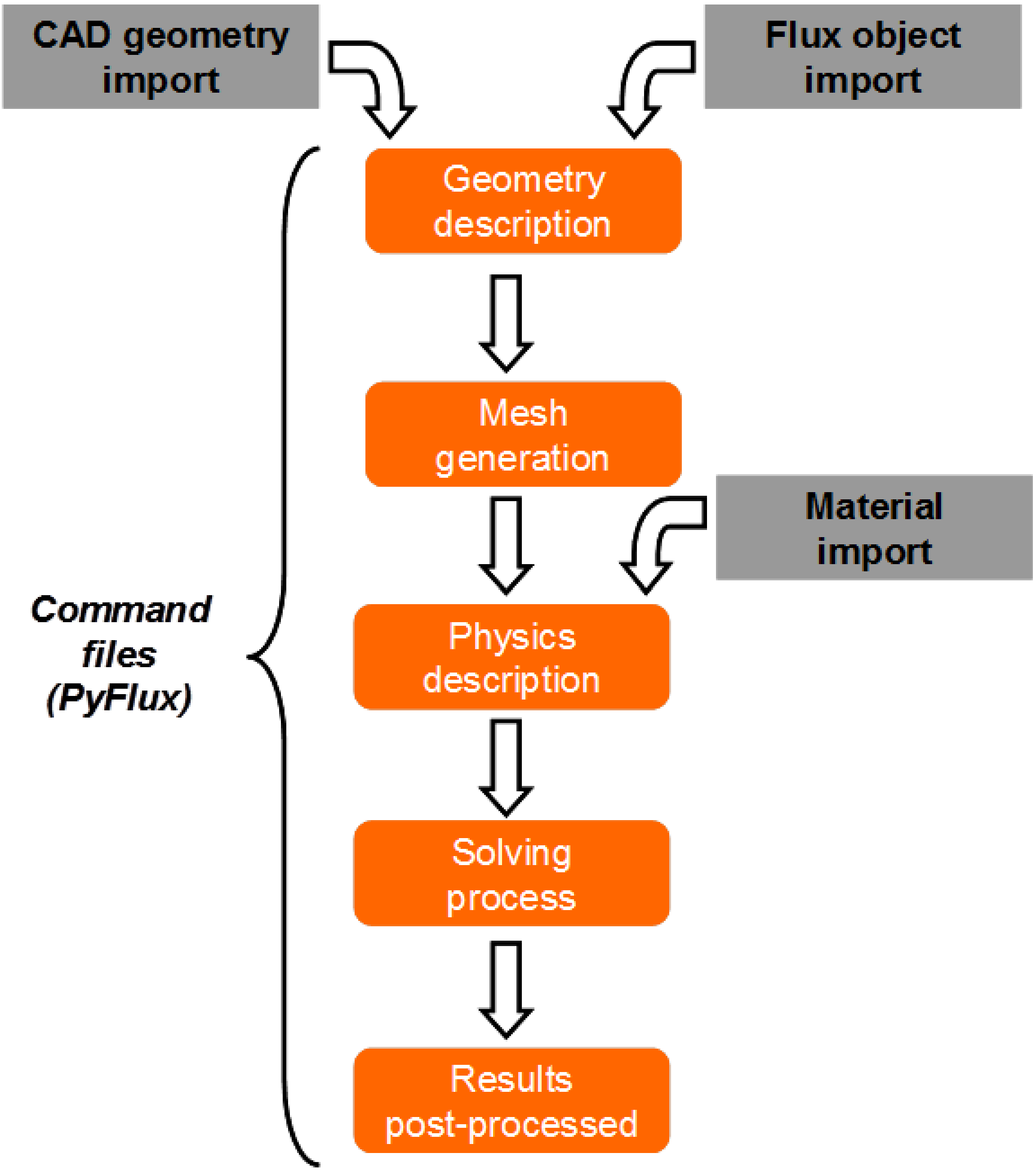


### Introduction

This chapter presents a list of the main new features of Flux 2019.1

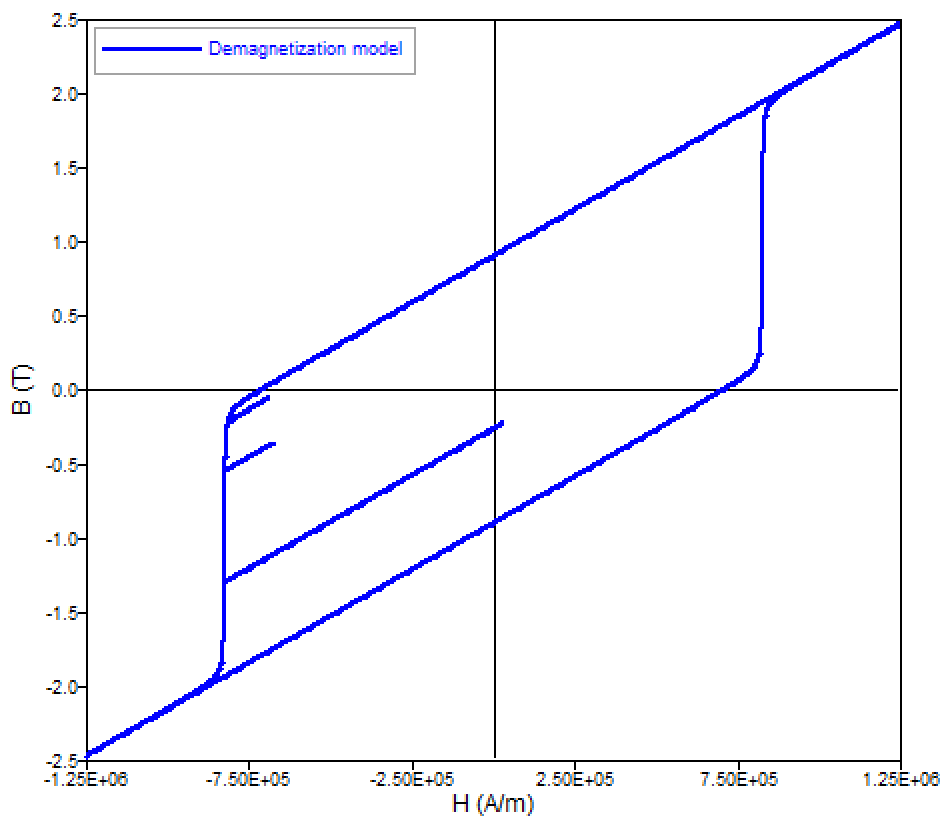
The main new features are listed and the chapter references are given, where the necessary information for a good usage of the new software capabilities is presented in detail.

To localize the impact of the various new features, the flowchart of Flux software principle is presented below.


CAD geometry import
Flux object import
Geometry description
Mesh generation
Material import
Physics description
Solving process
Results post-processed
Command files (PyFlux)

# 1.1 New features of Flux 2019.1

### New features dealing with Physics

| New features | Description |
|---|---|
| **Magnet demagnetization during solving process** | Flux 2019.1 offers the capability to take magnet demagnetization phenomena into account during solving process for 2D and 3D transient applications, thus empowering the modelling of devices like rotating electrical machines. For example, computation of typical quantities such as motor torque or electromotive force are now more accurate and new analysis like the evolution of the remanent flux density on magnets are available.<br><br>This feature, based on a static Preisach model implemented in the Flux solver, applies to permanent magnets described by non-linear materials defined by their coercive force Hc and their remanence Br.<br><br><br><br><br>*Figure 1: B(H) curve showing demagnetization* |

<table>
<tr><th>New features</th><th>Description</th></tr>
<tr><td></td><td>

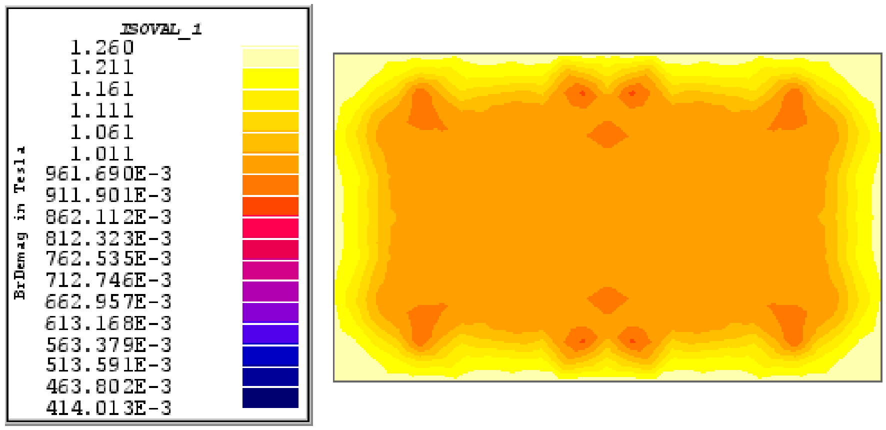
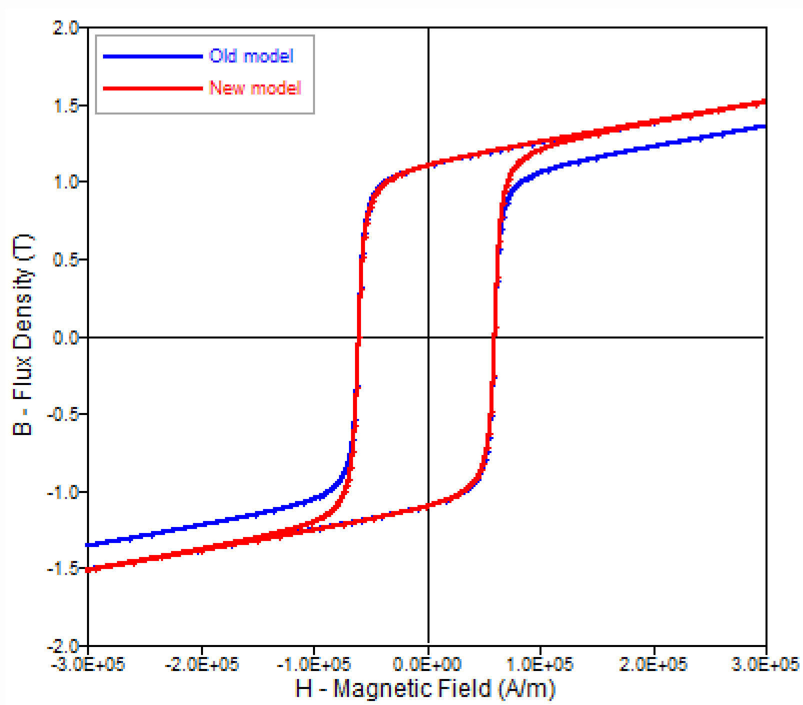



*Figure 2: Colour shades of remanent flux density on a magnet*

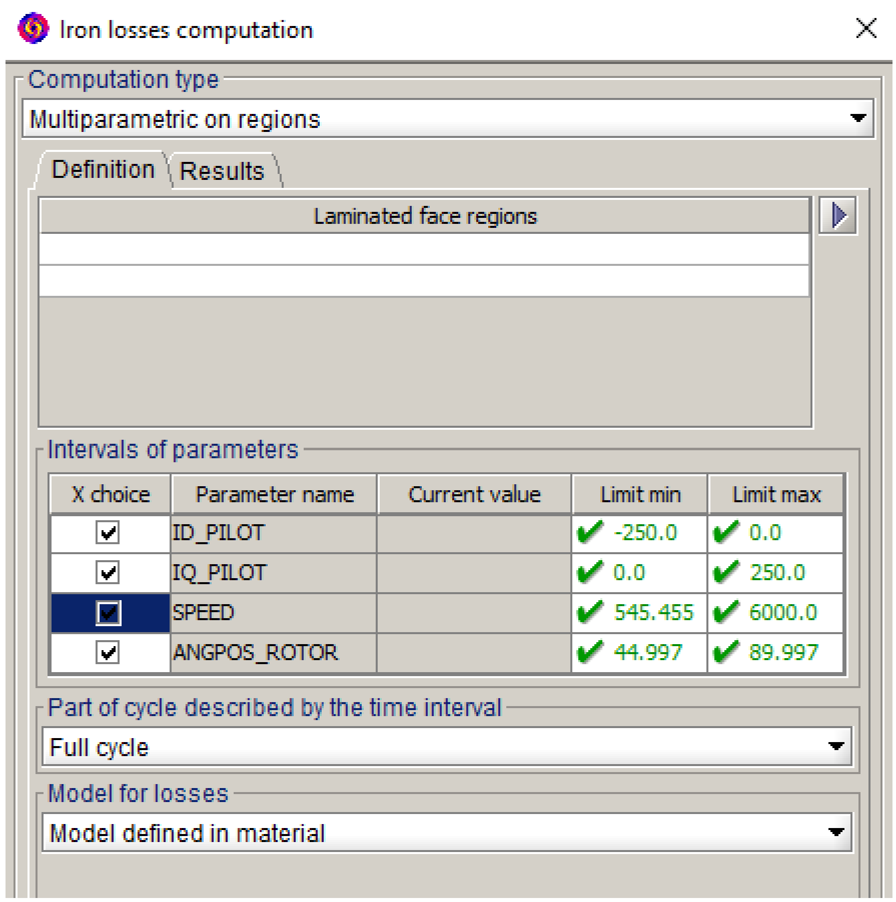
</td></tr>
<tr><td>Upgrade of one non-linear magnet model</td><td>

To make demagnetization possible during solving process, the model for permanent **"Nonlinear magnets described by Hc and Br module"** has been upgraded with Flux 2019.1, to have a closed-loop B(H) characteristic, which is necessary to compute demagnetization phenomena. The previous and the new models are compared in the picture below.

In case you have a Flux project solved with a previous version of the software, you are invited to delete results and solve the project again to take benefit of this enhanced model.




</td></tr>
</table>

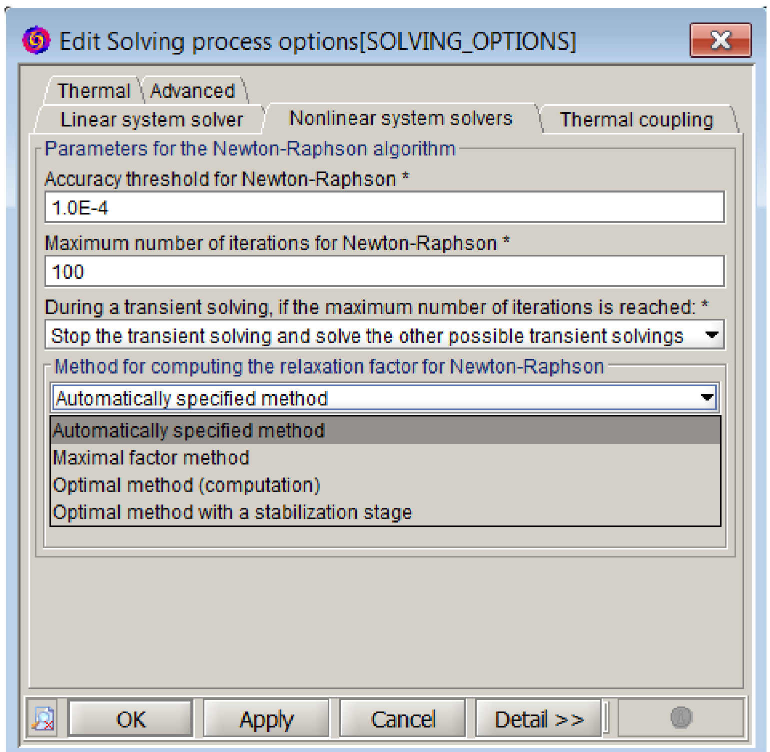
| New features | Description |
| --- | --- |
| **Multi-parametric computation of iron losses** | In addition to already-available iron losses computations on regions and points, Flux 2019.1 adds the ability to calculate these losses for multi-parametric scenarios with a single click.<br><br>Results of this computation, which applies to laminated regions for both Bertotti and LS models, are gathered and stored in a dedicated I/O parameter, whose values can be easily displayed by plotting 2D or 3D curves and also exported as row data tables. Creation of efficiency maps, as well as definition of optimization process are now more straightforward than ever.<br><br>At the same time, the dialog box for running all these iron losses computations was refactored to make it more user-friendly<br><br> |

### New features dealing with Solving

<table>
<tr><th>New features</th><th>Description</th></tr>
<tr><td>Simplification of "Nonlinear system solvers"solving options</td><td>

With Flux 2019.1, the **Nonlinear system solver** options tab has been simplified. In **Standard** mode, the user has only to choose between 4 types of method for computing the relaxation factor for Newton-Raphson before solving:

- **Automatically specified method**
- **Maximal factor method**
- **Optimal method (computation)**
- **Optimal method with a stabilization stage**




By doing this, the options have been standardized. The older solvers used in the previous version of Flux stay available through the **Advanced** mode. This mode can be set using the supervisor.

Solving process options are applicable for all physical applications of Flux 2D, 3D and Skew.

</td></tr>
<tr><td>Simplification of "Advanced" solving options</td><td>

With Flux 2019.1, the **Advanced**tab has been simplified. The **method for gauge for edge variables** choice appears only in 3D, when we could need it. So now, the **method for gauge for edge variables** choice doesn’t appear in 2D or Skew.

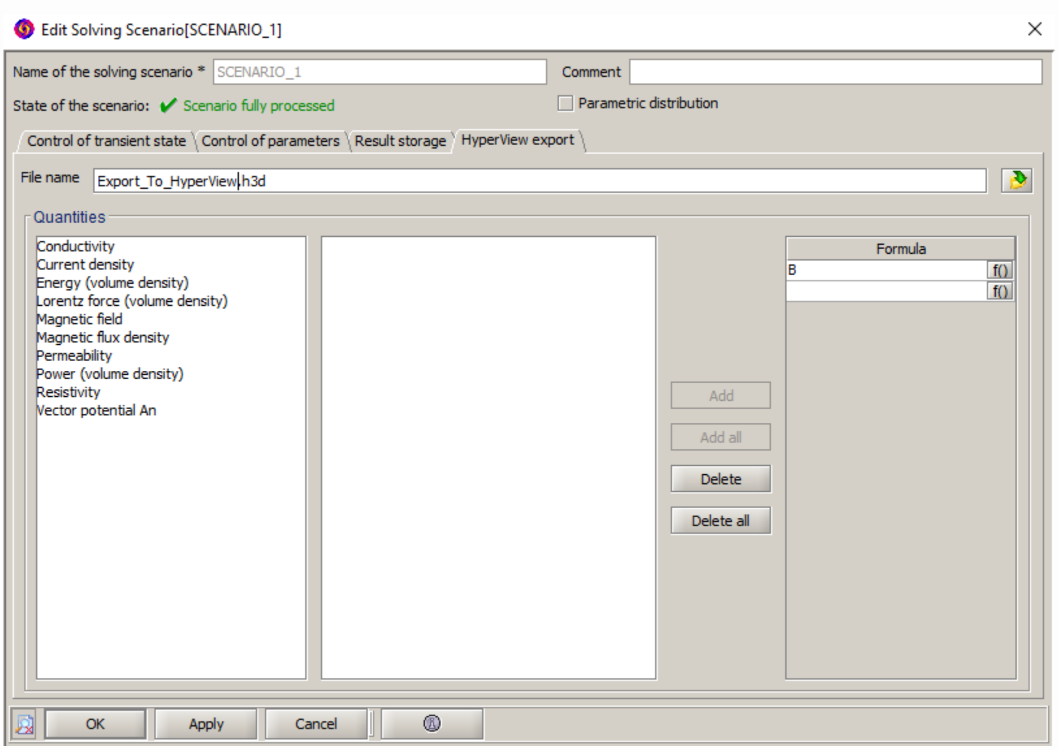
By doing this, the options have been simplified. The method for gauge for edge variables stay available through python command.

</td></tr>
</table>

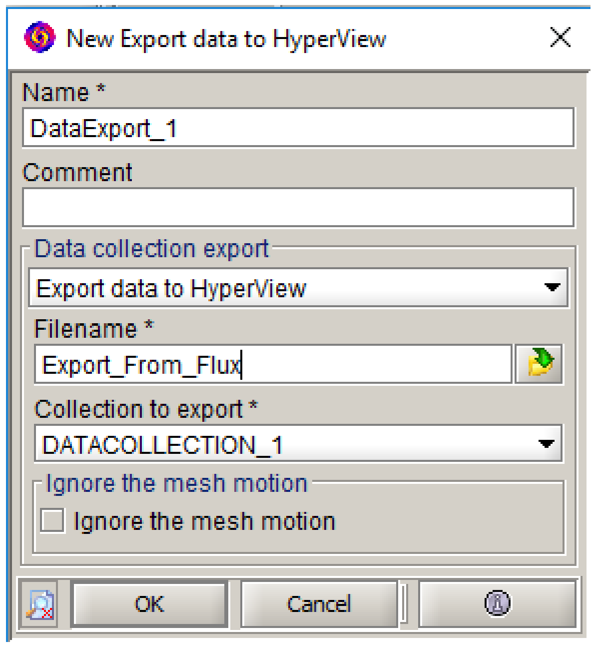
| New features | Description |
|---|---|
| **Speed-up the computation for non-meshed coils** | With Flux 2019.1, the **computation of Hj by Biot-Savart** for each **non-meshed coil** during the solving was **speed-up** by refactoring and parallelizing the code. Therefore, the computation of the flux for the magnetic scalar potential formulation without circuit was also speed-up. |
| **Remeshing with A formulation in 3D** | In Flux 2018.0, we implemented the magnetic vector potential formulation, or **A** formulation, for 3D project. This formulation cannot take into account the movement. Until now, you had to use the magnetic scalar potential formulation, or ϕ formulation, into your 3D projects with movement.<br>With Flux 2019.1, you could now use the 3D **A** formulation for translating motion, i.e. with compressible mechanical set. To take into the movement in this case, the idea is to remesh the compressible mechanical set.<br>This new feature is only available with the MeshGems mesher. |
| **Induction machine with a slip modeled in the AC state by the A formulation in 3D** | In Flux 2018.0, we implemented the magnetic vector potential formulation, or **A** formulation, for 3D project. This formulation could not take into account the slip of the 3D induction machine in the AC state. Until now, you had to use the magnetic scalar potential formulation, or ϕ formulation, into your 3D induction machine in the AC state.<br>With Flux 2019.1, you can now use the 3D **A** formulation to model the slip of induction machine in the AC state.<br>This new feature is available only without movement of the mechanical sets. |
| **Results preview Improvement** | During solving the evolution of predefined quantities and / or quantities defined by the user are displayed. Quantities which can be displayed during the solving are:<br>• Predefined quantities of mechanical sets<br>• I/O parameters<br>• Real scalar sensors<br>The user can thus predict the quantities that he wishes to control during the solving by creating the sensors / parameters necessary beforehand.<br>***Flux 2019.1 improvement:*** By default all eligible sensors / parameters are taken into account. If the user does not wish to see them all, the names of the unwanted sensors / parameters must start with the character **_**.<br>Example : **_Param1**. |

## New features dealing with Post-processing

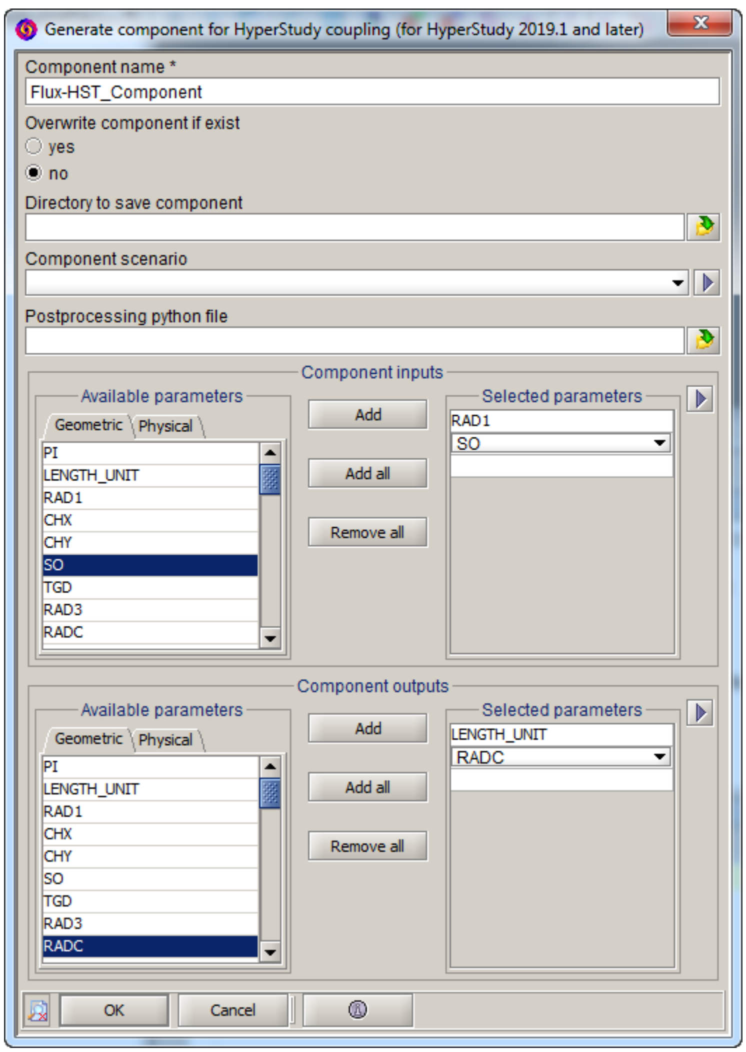
| New features | Description |
|---|---|
| **Animation improvement** | The codec used to generate movies via the Animation in Flux has been updated.<br>Now the **animations** are more robust, with a much smaller size but always with good quality |
| **H3D export of Flux results for augmented post-processing** | Two new features are available with Flux 2019.1 to export results in H3D format to be read by other Altair software, like HyperView, HyperGraph, OptiStruct, AcuSolve or SimLab.<br><br>The first one, presented here, is available via the Flux scenario dialog box and it is mainly proposed to users willing to take benefit of all the extended post-processing capabilities offered by HyperView and/or HyperGraph. The selected quantities are exported for all scenario steps and on the whole computation domain; the H3D file is generated during the Flux computation, thus allowing the user to have preview of colour shades, arrows, etc, without waiting the end of solving. The generation of this H3D file does not yet work with Flux 2019.1 in case of distributed computations and in case of domain remeshing (e.g. adaptive time-step, geometric parameters piloted by the scenario). |

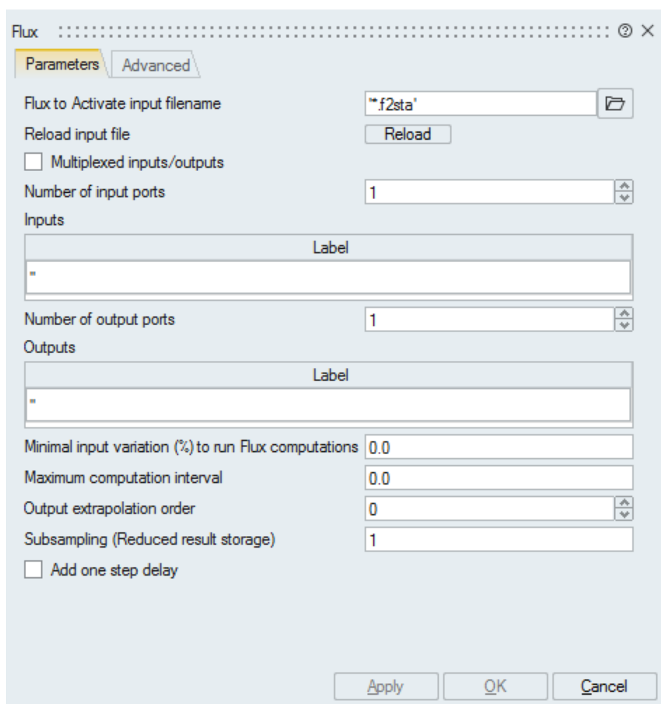


| New features | Description |
|---|---|
| **H3D export of Flux results for multiphysics couplings** | This second feature related to H3D export has the main goal to make Flux even more integrated into the multiphysics processes that involve other Altair tools, like OptiStruct, AcuSolve and Simlab. This H3D export is available only after the Flux computation is completed and it is accessible via the Import/Export context for all data collections, in the same manner as the other multiphysics exports are (Nastran, native OptiStruct, native Acusolve, etc.). The H3D generated file may contain one quantity only, computed on the user-defined support, on a user-selected list of scenario-steps. |



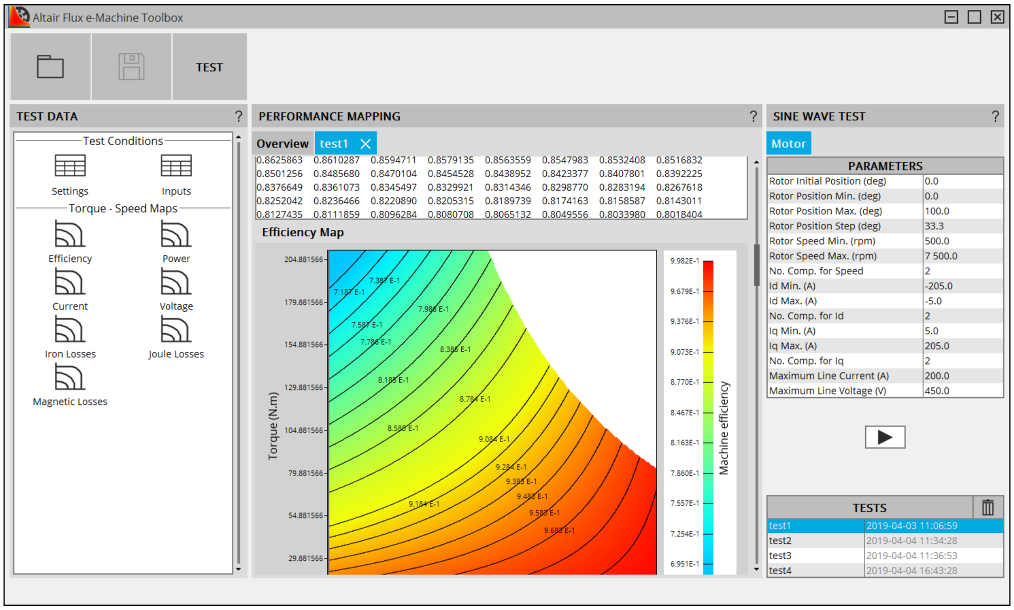
## New features dealing with Coupling

| New features | Description |
| --- | --- |
| **Flux - HyperStudy Coupling** | HyperStudy is a multi-disciplinary design exploration, study, and optimization software for engineers and designers.<br>It is interesting to couple it with Flux to leverage the HyperStudy approaches (DOE, Fit, Optimization and Stochastic) for Flux models design exploration and optimization.<br>The HyperStudy to Flux connection allows exploring efficiently the design space of any electromagnetic devices modelled in Flux (motor, actuator, transformer…)<br>Before this 2019.1 version, The coupling between Flux and HyperStrudy has been possible via the Got-It component. Starting from Flux 2019.1 it exist in Flux a dedicated command to generate a coupling component dedicated for HyperStudy. A xml file has been created and HyperStudy 2019.1 can read it to achieve a study of otpimisation with Flux.<br> |

<table>
<tr><th>New features</th><th>Description</th></tr>
<tr><td>Linux Flux API</td><td>With Flux 2019.1, we can now use in Linux the same APIs as in Windows to pilot Flux.</td></tr>
<tr><td>Flux-Activate co-simulations</td><td>An enhanced version of the Flux-Activate coupling component is introduced with Flux 2019.1 and Solid Thinking Activate 2019.2: in the Flux environment the definition of the coupling component remains the same, however its GUI on the Activate side was improved:
<ul>
<li>to introduce an “Advanced” tab for Flux memory settings</li>
<li>and to add in the “Parameters” tab two new options: the “maximum computation interval” and the “output extrapolation order”. Both are intended for providing more flexibility to users when they drive Flux by Activate and for enhancing simulation performance</li>
</ul>
In particular, the field “maximum computation interval” enables the user to force a Flux computation even if the input values provided by Activate have not changed during this interval. In fact, some electromagnetic phenomena may have modified the Flux output values that it is necessary to send to Activate to improve the accuracy of the whole co-simulation process.<br><br>
The option “output extrapolation order” allows Activate to linearly extrapolate (order set to 1) the Flux outputs for usage as Activate inputs in the next computation steps, thus guaranteeing better result quality with respect to previous versions where Activate inputs coming from Flux were never changed (extrapolation order to set 0).<br>

</td></tr>
</table>

## New features dealing with dedicated tool for e-Machine post-processing

<table>
<tr><th>New features</th><th>Description</th></tr>
<tr><td>Flux e-Machine Toolbox</td><td>
Flux e-Machine toolbox is a new application in Flux dedicated for Electric Machine. This tool permits to compute and postprocess results dedicated for e-Machine like efficiency maps. It's a first version. It is planned to enrich the possibilities of postprocessing in future versions.



</td></tr>
</table>

# 1.2 New features dealing with the documentation of Flux 2019.1

### New features about the user documentation

Starting from the Flux 2019 version, we begin a migration work of the user documentation with the following aim:

- to be homogeneous with all the online help of the different softwares of HyperWorks™ suite proposed by Altair
- to offer you more accurate and accessible online help

This transition will be organized on several versions. Temporarily the documentation of new features and updated features will be published in pdf format and will be accessible by links integrated directly into the current HTML documentation associated with Altair Flux™ software

### New features about examples

As a reminder, all examples are accessible via the Flux supervisor, in the context of **Open an example**.

| New examples | Description |
|---|---|
| **FeMT 2D** | The Efficiency MAP of electrical motors becomes one of the key points for the design of electric motor. Indeed, for users of Flux software, it is interesting to know the optimal operating points of their electrical machines. And this in order to find directly the most advantageous areas in terms of efficiency in the torque-speed envelope characteristics of the machine. The main objective of FeMT tool is to obtain the efficiency MAP for the electrical motors. |
| **Wireless power transformer (3D)** | This model is based on the wireless power transfer demo kit DC2386A by Linear Technology. It features two different boards for the transmitter and the receiver: the transmitter coil is circular spiral coil with reported inductance value of 24 μH, while the receiver coil has 47 μH inductance value. During this example coupling coefficient and self and mutual inductances of the coils will be calculated. Finally, for a deeper understanding of the transfer process and in order to improve it a capacitor will be added in series with the transmitter and receiver coils and its influence will be analyzed. |

| New examples | Description |
|---|---|
| **Coupling Flux-Activate on E-mobility motor** | Electrical machine are used everywhere and it will be a major issue for sustainable development. Today it's a true challenge to make an efficient design of an electrical machine which should answer to several points: the influence of eddy current on the machine works, how to optimize the machine, what is the best strategy command in order to manage the energy consumption… all those questions are crucial points wondering by designers. That's why it is necessary to know how to model the behavior of an electrical machine with different levels of modeling in order to simulate a complete system as a powertrain for electrical car. There are different approaches to represent the behavior of a Permanent Magnet Synchronous Machine (PMSM). The first one is an analytical approach with **Park model** (classical first approach). The second one is using magneto-static tables calculated with finite element (software Flux) or FluxMotor, these model gives a better accuracy. The last model is the co-simulation between the finite element software in **transient mode** and the Activate. The case of co-simulation depending on the type of the kinematics, there are two cases: with coupled load and multiphysics position. A comparative study has been made to show what are the advantages and disadvantages of each method. |

# 1.3 New features dealing with the macros of Flux 2019.1

### List of new macros

| New Macros | Description |
|---|---|
| Find_rotor_angle_2D | This macro is used to find the rotor angle for t=0s in order to align it with the magnetic induction generated in the airgap by the reference phase (e.g., phase A). It is applied to radial synchronous machines in 2D transient magnetic applications.<br>**Inputs:**<br>• Number of pole pairs<br>• Current sources corresponding to electrical phases (the first one becomes the reference phase)<br>• The rotor mechanical set<br>• The stator mechanical set<br>**Output:**<br>• New variable (parameter defined by a formula) called POS_INI which contains the desired initial rotor angle for t=0s (in degrees) |
| Find_rotor_angle_Skew | This macro is used to find the rotor angle for t=0s in order to align it with the magnetic induction generated in the airgap by the reference phase (e.g., phase A). It is applied to radial synchronous machines in skewed transient magnetic applications.<br>**Inputs:**<br>• Number of pole pairs<br>• Current sources corresponding to electrical phases (the first one becomes the reference phase)<br>• The rotor mechanical set<br>• The stator mechanical set<br>**Output:**<br>• New variable (parameter defined by a formula) called POS_INI which contains the desired initial rotor angle for t=0s (in degrees) |

<table>
<tr><th>New Macros</th><th>Description</th></tr>
<tr><td>Find_rotor_angle_3D</td><td>This macro is used to find the rotor angle for t=0s in order to align it with the magnetic induction generated in the airgap by the reference phase (e.g., phase A). It is applied to radial synchronous machines in 3D transient magnetic applications.<br>Inputs:<ul><li>Number of pole pairs</li><li>Current sources corresponding to electrical phases (the first one becomes the reference phase)</li><li>The rotor mechanical set</li><li>The stator mechanical set</li></ul>Output:<ul><li>New variable (parameter defined by a formula) called POS_INI which contains the desired initial rotor angle for t=0s (in degrees).</li></ul></td></tr>
<tr><td>CreatePark_Iabc_Drivenby_Idq</td><td>Create Iabc variables driven from Id, Iq variables in order to compute Torque versus Id and Iq from magnet motors<br>Entrées:<ul><li>Parameter corresponding to I_A</li><li>Parameter corresponding tor I_B</li><li>Parameter corresponding to I_C</li><li>Mobile mechanical set</li><li>Number of poles</li><li>Angle for which rotor is aligned with phase a</li><li>Maximum value for Id</li><li>Number of values for Id</li><li>Maximum values for Iq</li><li>Number of values for Iq</li><li>Number of positions for rotor position</li></ul>Sorties:<ul><li>Create ID and IQ parameters</li><li>Create ID_DRIVEN and IQ_DRIVEN I/O parameters driven by solving scenario</li><li>A solving scenario is created (ready to be solved with distribution of computation)</li></ul></td></tr>
</table>

| New Macros | Description |
|---|---|
| ComputeSkewEffectFromCurve | Trouver le couple avec skew.<br>**Entrées:**<br>• Valeur de l'angle du « skew »<br>• Quantité à calculer<br>• Ensemble mécanique mobile<br>**Sorties:**<br>• Courbes de grandeur avec et sans effet du skew<br>• Paramètre entrée/sortie |
| ConvertCSVFileToOMLFile | Convert directly CSV file into OML file. The input are recognized.<br>.<br>**Input:**<br>• Select CSV file<br>**Output:**<br>• Créer un fichier OML à partir du nom initial (*_res.oml) |
| CreateLookUpTableFromTMProject | Create look up table from TM project of Flux abc and torque versus Id, Iq and rotor position.<br>**Input:**<br>• Select CSV file<br>**Outputs:**<br>• Créer un fichier OML à partir du nom initial (*_res.oml)<br>• It will also include more data such as phase resistance, end winding inductance, electric period and initial rotor position |

# Detail of new features of Flux 2019.1

# 2

This chapter gives more details of each main new features of Flux 2019.1.

This chapter covers the following:

# 2.1 Demagnetization during solving

For non linear magnets defined by **Nonlinear magnet describes by Hc and Br module**, Flux can take in acccount demagnetization during solving with demagnetization based on a static Preisach model, wich allows you to be more accurate for results like torque or EMF for permanent magnet synchronous machine.

- Available for 2D and 3D in magnetic transient application
- Initialization by static calculation (Application > Transient initialization)
- This model does not take in account temperature variations

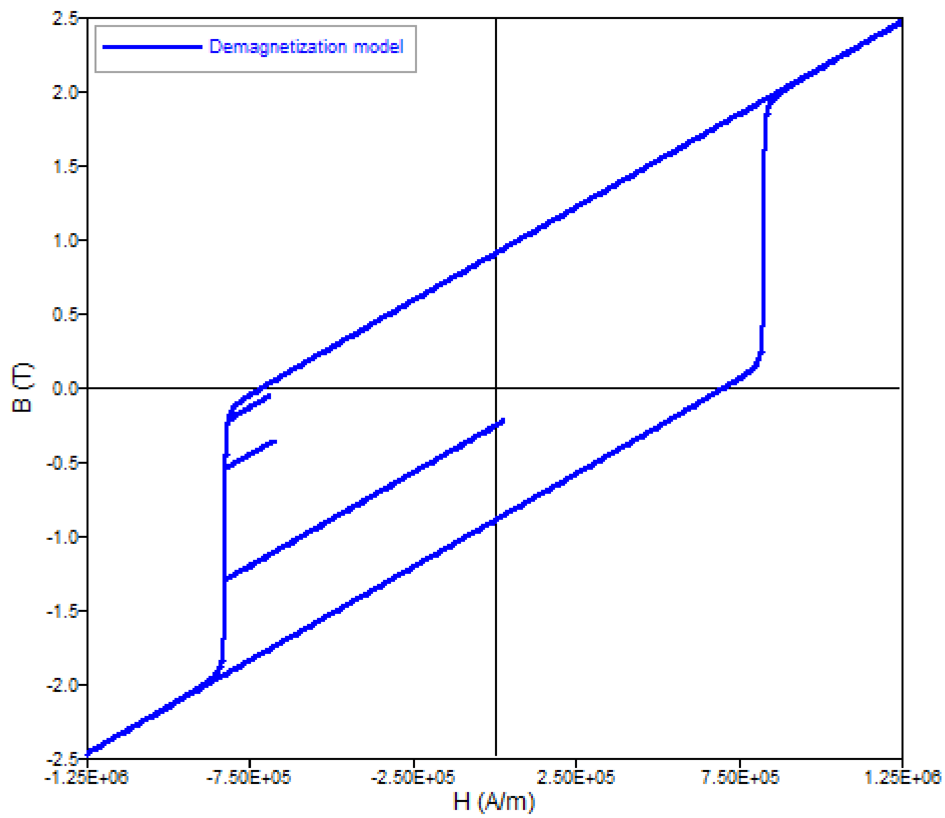



*Figure 3: Demagnetization model*

To use this new model with a solved project :

- Destroy results
- Go in **Application** > **Transient initialization** and select : **Initialization by static calculation**
- Create a new material **Nonlinear magnet describes by Hc and Br module**
- Check **the thick  Taking in account demagnetization during solving**
- Assign the material to regions
- Go to **Physics** > **Face regions** (in 2D) ou **Voume regions** (in 3D) > **Orient material for face / volume regions**
- Run the scenario

- Create a new **isovalues**, select magnet and add **BrDemag** in the formulas field

Example of results : Surface permanent magnets motor 3D

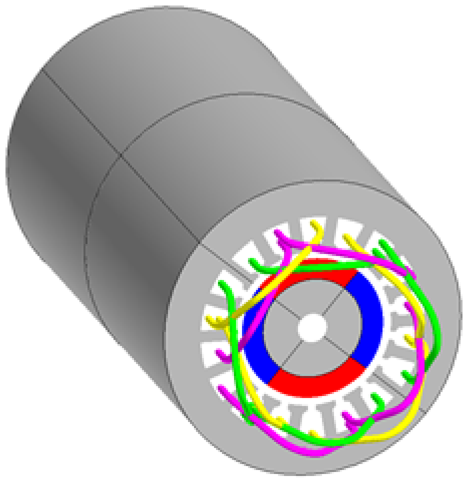


*Figure 4: Full device*

For a control angle $\Psi = \left[\ 0\ ;\ \frac{\pi}{6}\ ;\ \frac{\pi}{4}\ ;\ \frac{\pi}{3}\ \right]$, we can see that there is a local effect on the remanent flux density :.

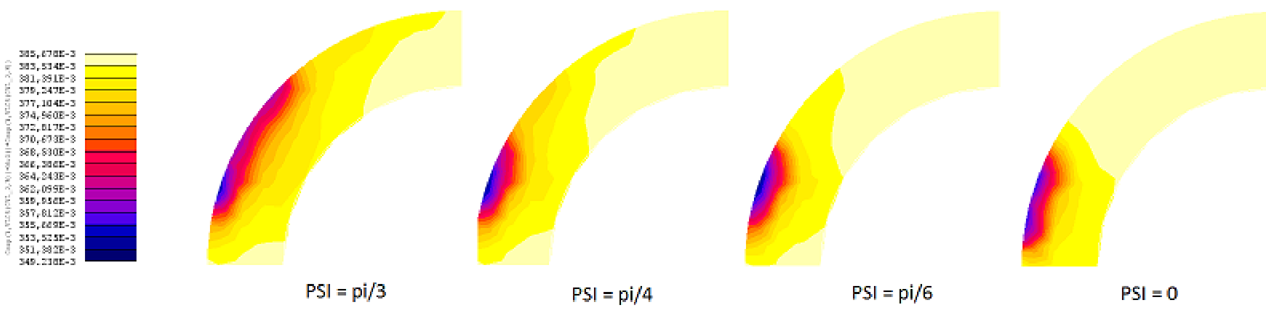



*Figure 5: Br in the magnet*

This local effect can bring some effect on global values like torque or EMF.

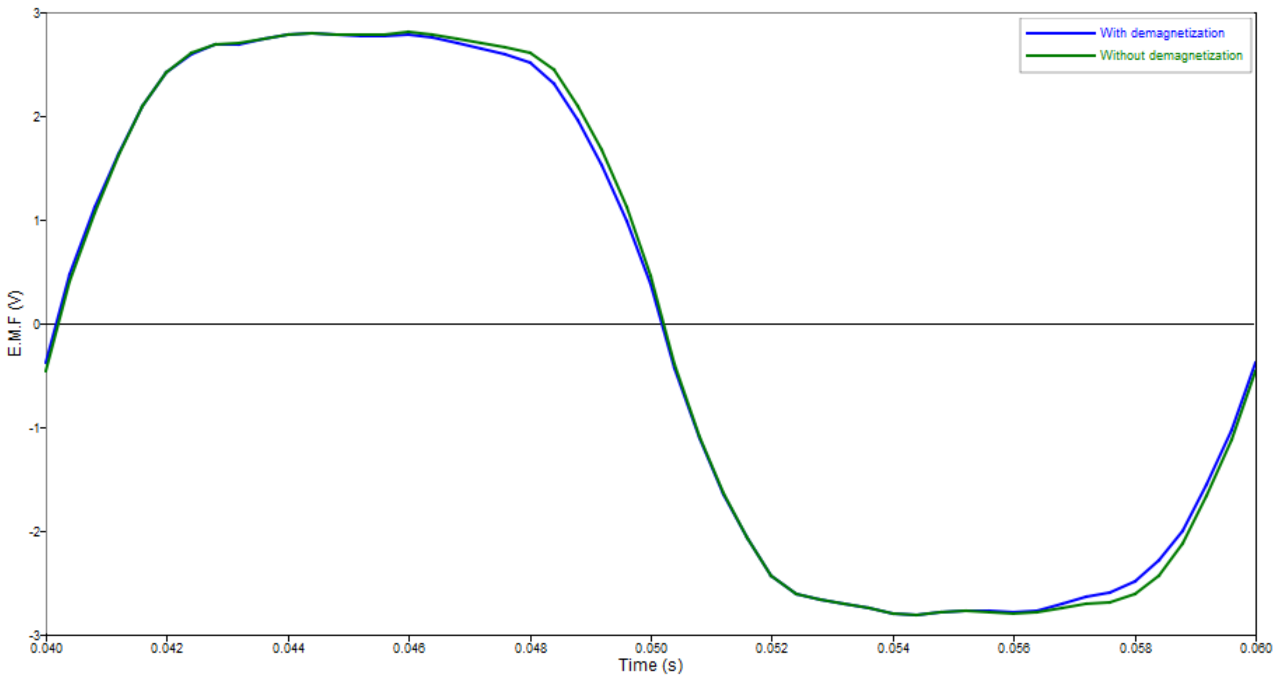



*Figure 6:E.M.F with and without demagnetization for* $\Psi = \frac{\pi}{4}$

**Note:**

- This new level of modeling can increase the computation time and the memory (ram & disk)
- Not available in 3D with the potential vector

# 2.2 Iron Losses

### Content

This chapter deals with iron losses handling in Flux:

# 2.2.1 Computation of iron losses

### Introduction

Magnetic losses (or iron losses) computation is a subsequent computation, carried out in Flux post-processor.

This chapter introduces iron losses computation, from modified Bertotti formulas, or with LS model (Loss Surface).

The power losses in electromechanical devices are mainly of three types:

- the magnetic losses in the magnetic circuits (also called 'iron losses')
- the losses by Joule effect in coils (also called 'copper losses')
- the mechanical losses (mainly by friction in rotating machines)

**Losses in magnetic materials:**

The power losses in magnetic materials are connected to the phenomena associated with the time variation of the magnetic field. They are subdivided into hysteresis losses (of microscopic origin) and Foucault currents losses (of macroscopic origin). In fact, it is eddy current in both cases.

- The hysteresis losses (microscopic Eddy currents) are associated to currents at a small scale. These currents are the result of local induction variation caused by the magnetic structure in movement (essentially wall movement).
- The Foucault losses (macroscopic Eddy current), are due to the excitation frequency. They appear when domain wall displacement increases due to frequency increase.

**Magnetic losses and hysteresis cycle:**

In practice, magnetic materials are characterized by their hysteresis cycle and the magnetic losses can be represented by this cycle, as presented below:

- the volume energy created by hysteresis losses is corresponding to static hysteresis cycle (f<1Hz).
- as the frequency increases, the cycle area increases and the volume energy created by eddy current losses is corresponding to the difference between the dynamic hysteresis cycle area and the static one.

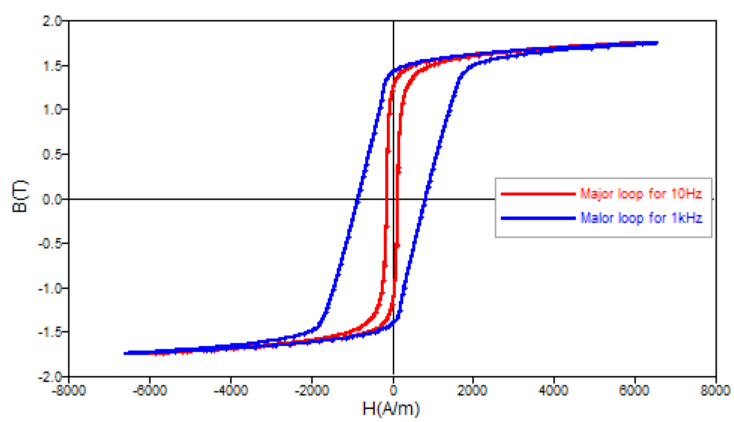



*Figure 7: Major cycles for different frequencies*

### Access to computation of iron losses

Iron losses computation can be accessed in 2D and 3D for laminated regions and for the following applications:

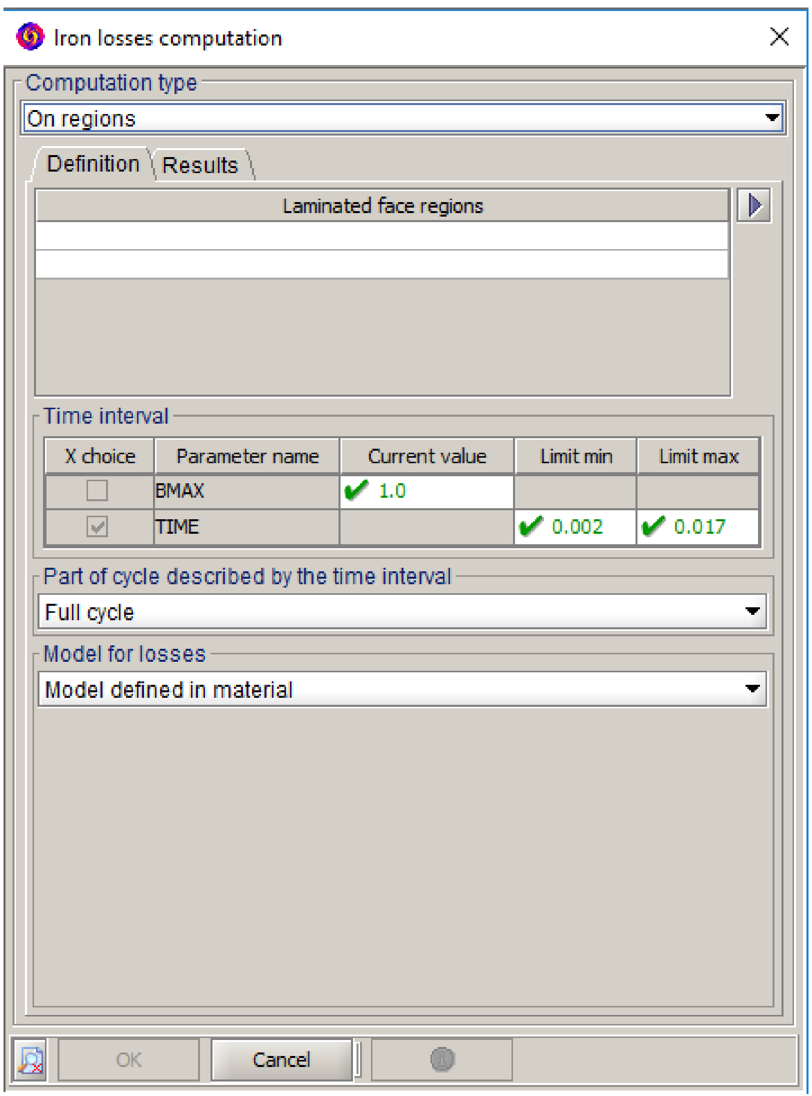
- For modified Bertotti model: Steady state AC magnetic (averaged iron losses computation) and transient magnetic (averaged or instantaneous iron losses computation)
- For LS model: Transient magnetic only (averaged or instantaneous iron losses computation) in Flux **Advanced** mode

**Note:** * Laminated regions are often used in magnetic circuits of transformers and of some electric machines in order to reduce Eddy current losses.

### Computation of iron losses

The computation box of iron losses can be accessed in the post-processing context via **Computation** > **Computation of iron losses** > **Computation of iron losses**

The box is as follows:



> **Note:** For information, the former iron losses models can be accessed via menu **Computation** > **Computation of iron losses** > **Deprecated versions**. For more information on these models, please refer to Deprecated version

## Computation type

The **Computation type** field enables to choose among the following possibilities:

- **On regions** : enables to compute modified Bertotti or LS iron losses on a selection of laminated regions. The iron losses model can be defined on the material or in this box via **Model for losses** field. The computation gives instant losses and averaged losses on the time interval for a set of fixed geometric parameters or fixed I/O parameters.
- **Multi-parametric on regions** : enables to compute modified Bertotti or LS iron losses on a selection of laminated regions. The iron losses model can be defined on the material or in this box via **Model for losses** field. The computation gives instantaneous losses for several sets of geometric parameters or I/O parameters. This computation can be used in optimizations or efficiency maps.
- **On point with LS model defined in the material**: this option can only be accessed in **Advanced mode**, in transient magnetic application. This calculus enables to compute LS iron loss volume densities on a point in a laminated region. Iron losses model must be defined in the material. The computation gives instantaneous losses and averaged losses volume densities on the time period for a set of fixed geometric parameters or I/O parameters. With this computation, it is possible to display the hysteresis cycle.

# 2.2.2 Iron losses model

## Introduction

Iron losses model can be defined:

- In the material
- In the post-processor, in Iron losses computation box

In both cases, the defined model is only used for the computation in the post-processor. Iron losses are not taken into account in the solving process.

This part describes both models as well as the definition steps in the material.

## Modified Bertotti model

**Presentation of the model:**

The theory of Bertotti gives us the expression of the magnetic losses depending on the frequency and the magnetic flux density.

The power density is expressed with transient magnetic relation:

$$dP = \left( k_1 \left( \frac{\max\ (B) - \min\ (B)}{2} \right)^{\alpha_1} f \quad + \quad k_2\ g(\alpha_2)\ \left| \frac{dB}{dt} \right|^{\alpha_2} \quad + \quad k_3\ g(\alpha_3)\ \left| \frac{dB}{dt} \right|^{\alpha_3} \right) K_f$$

With

$$g(\alpha) = \frac{2}{(2\pi)^{\alpha\text{-}1}\ 4\int \cos^{\alpha}\ (\theta)\ d\theta}$$

where:

- k1: is the coefficient of losses by hysteresis
- k2: is the coefficient of classical Foucault currents losses
- k3: is the coefficient of supplementary losses or in excess*
- α1: is the exponent of losses by hysteresis
- α2: is the exponent of classical Foucault currents losses
- α3: is the exponent of supplementary losses or in excess*
- f: frequency
- B: induction
- Kf : fill factor of the region

* the distinction between supplementary losses / in excess losses and classical losses is artificial. They can be grouped in one term and they therefore correspond to real induced currents flowing in the sheet.

Note: An Excel sheet permitting to determine, for a sinusoidal magnetic flux density, coefficients and exponents of Bertotti formulation is available. This tool, as well as its documentation, have been stored on your space disk, during Flux installation, at the following path :

C:\Program Files\Altair\2019\flux\Flux\DocExamples\Tools\BertottiLossesCoefficients

(If Flux is installed at the path proposed by default (C:\Program Files\Altair\)

**Limits of validity**

In a Steady state AC Magnetic application, power density is expressed with the following equation:

$$dP = k_1 B_{max}^{\alpha_1} f + k_2 (B_{max} f)^{\alpha_2} + k_3 (B_{max} f)^{\alpha_3}$$

It is important to note that in the previous formula, the Bmax variable stands for the peak value of the magnetic flux density.

The software utilizes the value of the magnetic flux density in each point. Consequently, it is convenient to be very careful with the results concerning the problems represented by the rotating machines with the Steady state AC Magnetic simulation. Indeed, for this type of simulation the rotor has a fixed position with respect to the stator, and the real rotor movement is modeled by changing the resistivity of the conductors of the rotor electric circuit. Thus, the calculated magnetic flux density is maximum in a point on the given position of the rotor in relationship with the stator, due to space harmonics. It follows that the calculated magnetic flux density does not correspond to the peak value of the magnetic flux density over a period in the time domain if the rotor were turning. Consequently, the computation of the magnetic losses must be utilized in this case with much caution.

Moreover, in the case of a non-linear approximation for the magnetic behavior law B(H), the saturation phenomenon, introduced by means of an equivalent model of magnetization, can alter the local values of the magnetic flux density.

**Steps of the modified Bertotti model definition in the material**

The steps of the modified Bertotti model definition in the material are as follows

1. Create material
2. In **Iron losses** tab, check **Model for iron losses computation** and choose **Modified Bertotti model**
3. Enter coefficients and exponents

## LS model (advanced)

CAUTION: This model is only available in Flux **advanced** mode.

**Presentation:**

The LS (Loss Surface) model is a method of estimation of the magnetic losses a posteriori, based on a model of dynamic hysteresis associated to a finite elements simulation. The LS model requires that the magnetic behavior of a material be perfectly well defined, having knowledge of a characteristic surface H(B,dB/dt) (determined experimentally). Thus, for a B(t) signal of a certain shape and frequency, we can go up via the H(B,dB/dt) surface to the H(t) field, and thus reconstruct the dynamic cycle of hysteresis corresponding to it. This principle is represented below.

- B(t) signal with common form and frequency
- Creation of a characteristic surface H(B, dB/dt) of the material measured experimentally.
- Reconstitution of H(t) signal (i.e. of the hysteresis cycle)
- Calculus of losses

**Characteristic surface H(B, dB/dt):**

For each of the materials, the characteristic surface H(B,dB/dt) can be obtained by using a Epstein type device for magnetic measurements in medium frequency.

An example of this type of surface is represented in the figure below.

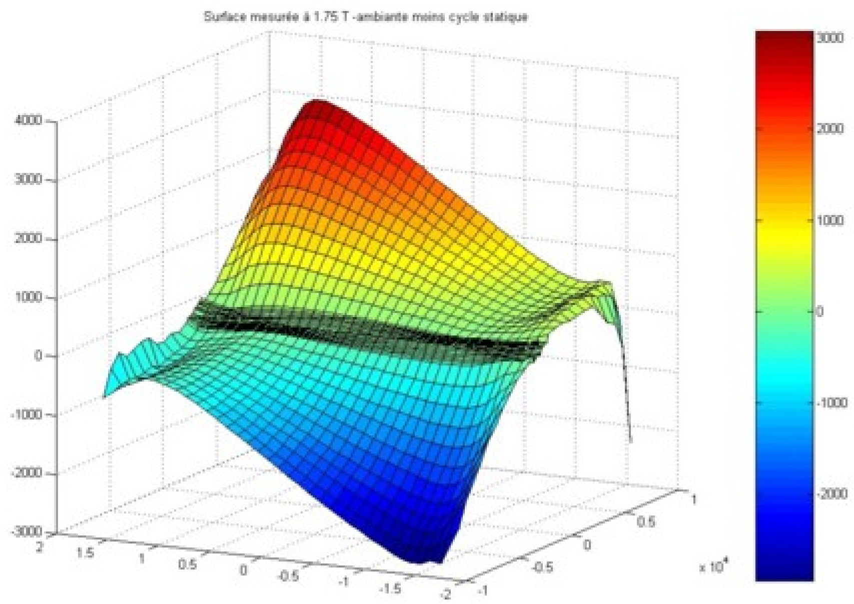



*Figure 8: H(B, dB/dt) surface*

**Reconstruction of the cycle:**

An analytical model permits the reconstruction of H field from values of B:

H(B,dB/dt) = Hstatic(B) +Hdynamic (B,dB/dt)

B(H) curves like those in the figure below can therefore be obtained, permitting the computations of iron losses quite accurately.

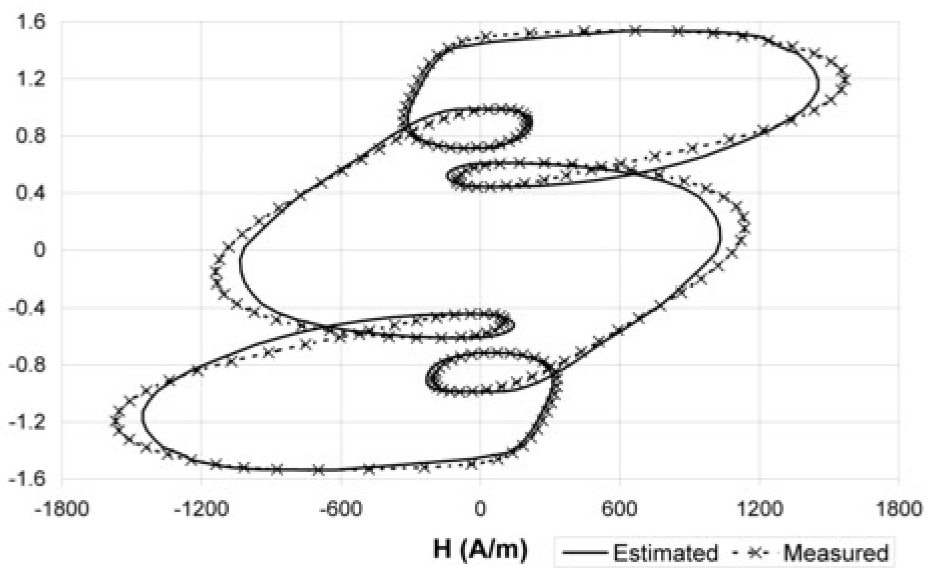



*Figure 9: Sinusoidal magnetic flux density & 5th harmonic at 200Hz*

The LS model is assigned to different nuances of laminations, which have been especially described to this purpose.

**Available materials:**

These are (by the international nomenclature IEC 60404-8-4-1998):

- M330-35A
- M800-50A

Iron sheet characteristic:

Magnetic polarization at saturation (Js in T) = 2.03T

Volume density (kg/m3) = 7650 kg/m3

If you will wish to add a new material, get in touch with Altair and the G2ELAB (Laboratoire de génie électrique de Grenoble); you should expect a minimum delay of approximately six months.

It is also possible to contact Spin (https://www.spinmag.it/); you should expect a delay of one month approximatively.

**Steps of LS model definition in the material**

The steps of LS model definition in the material are as follows:

1. Launch command with **Physics** > **Material** > **Create an LS material**
2. Choose definition of LS model:
    - Select iron sheet
    - Select a .mils file
3. Enter the following option:
    - Selection of anhysteretic curve / first magnetization
    - Selection of replace existing material / create a new material

**Note:**

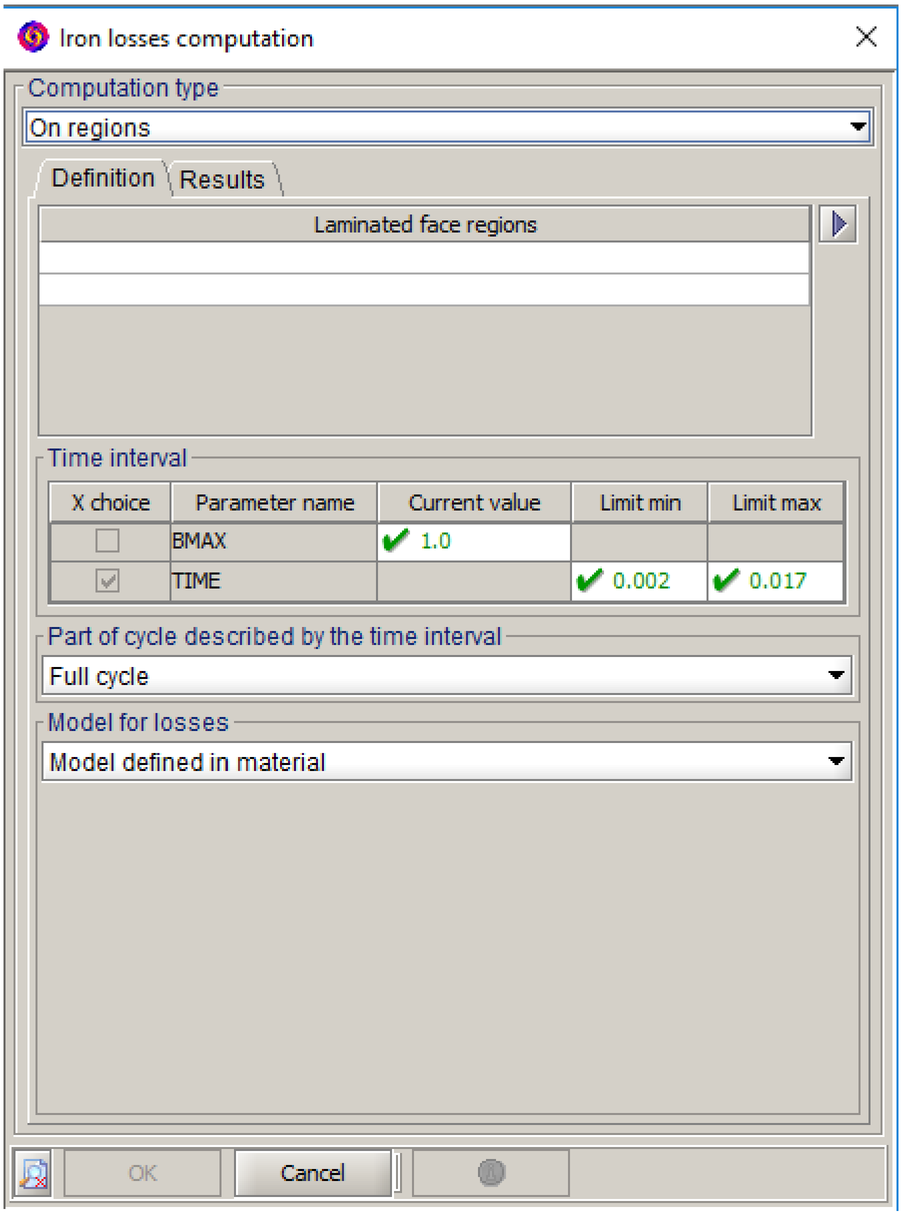
- Additional information on iron loss with LS model is available in the following document: « Module LS pour l'estimation des pertes fer dans les machines électriques » - A. Lebouc – Rapport d'étude mai 2004

# 2.2.3 Iron losses computation on regions

Iron losses computation on regions enables to compute modified Bertotti or LS iron losses on a selection of laminated regions. The iron losses model can be defined on the material or in the iron losses computation box in post-processing. The computation gives instantaneous losses and averaged losses over a period for a set of fixed geometric parameters or fixed I/O parameters

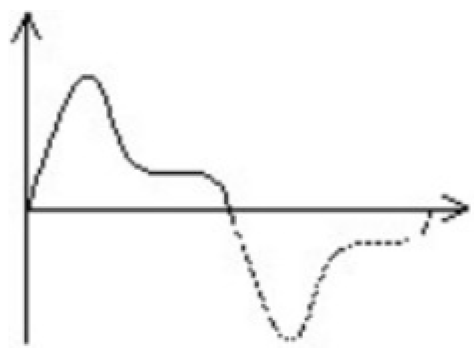
## Steps for iron losses computation on regions

This section describes the steps to carry out iron losses computation on regions:




1. Choose a **Computation type**: **On regions**
2. Select laminated regions for iron losses computation

3. Choose:
   - Fix values for geometric parameters and I/O parameters
   - In transient magnetic: time interval to obtain a time period. This choice is linked with the field **Part of cycle described by the time interval**
4. In Transient magnetic, define the time period selected in the previous step. The various possibilities are presented in the table below:

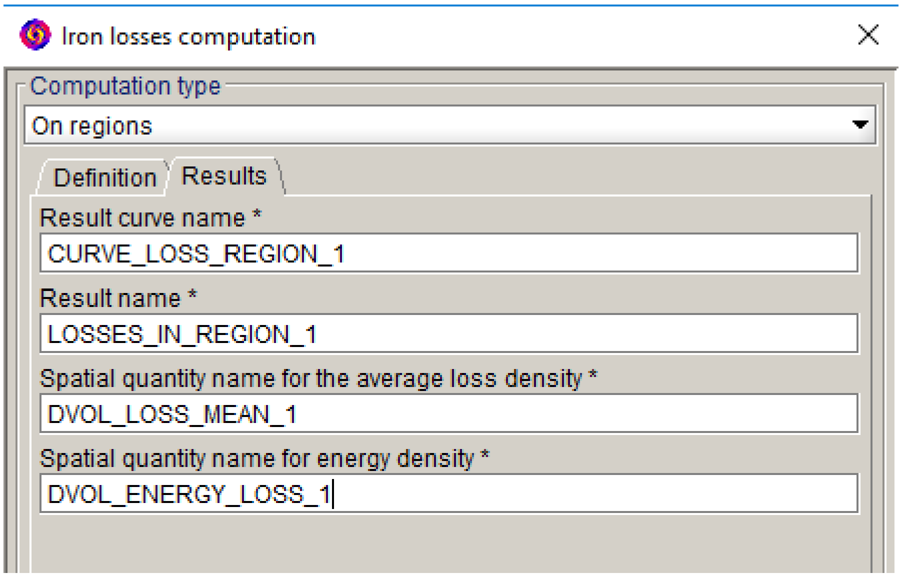
| Choices | Description |
| --- | --- |
| **Full cycle** | If the period is composed by N time steps,<br>The user selects time steps i and N+i<br>(The time steps i and N+i are identical)<br>Remark: The first two steps of the scenario are hidden as they cannot be selected for this computation. |
| **Half cycle** | If the period is composed by 2N time steps,<br>The user selects time steps i and N+i<br>(Flux reconstitutes the electrical period which comprises 2N+i time steps. The time steps i and 2N+i are identical)<br>Pair symmetry f(T/2+t) = - f(t)<br><br>**Note:** Half cycle option can manage a signal with a DC component. |
| **No cycle** | Flux does not take into consideration the period. |

5. Choose losses model:
   - Model defined in material
   - Modified Bertotti: Enter the coefficients values and modify exponents if necessary
   - LS predefined sheets (in transient magnetic, in Advanced mode): Enter the sheet iron
   - LS defined by importing of a MILS file (in transient magnetic, in Advanced mode)

# Results of iron losses computation on regions

## Computation results

The computation of iron losses on regions gives the following results




## Curve

"CURVE_LOSS_REGION_i" curve can be displayed automatically and is stored in the tree in the area **Postprocessing** > **Curve** > **2D curve (I/O Parameter)**.

It represents instantaneous losses on the region(s) versus time.

## Result

The results are automatically displayed in the data tree in the area **Postprocessing** > **Result** > **Iron losses Result**.

The results include:

- Average losses over a period on region(s) (in W)
- Iron losses energy on region(s) (in J)

## Spatial quantities

Spatial quantities, denominated "DVOL_ENERGY_LOSS_i" and "DVOL_LOSS_MEAN_ i" are stored in the data tree in the area **Parameter/Quantity** > **Spatial quantity**

Spatial quantities correspond to:

- Average loss density "DVOL_LOSS_MEAN_i"
- Losses energy density "DVOL_ENERGY_LOSS_i"

## Display loss density (isovalues)

It is possible to display the above mentioned spatial quantities in a color shade graphic (isovalues).

For this:

- In **Graphic** menu:

  Point on **Isovalues** and click on **New**
- In the box:
  - Choose a support (laminated regions for iron losses computation)
  - In the spatial quantities area, open the formula editor... (click on f())
- In the **Editor of formula and spatial quantities** box / **User** tab, select one of the following spatial quantities:
  - DVOL_LOSS_MEAN_i
  - DVOL_ENERGY_LOSS_i
- Confirm by clicking **OK**

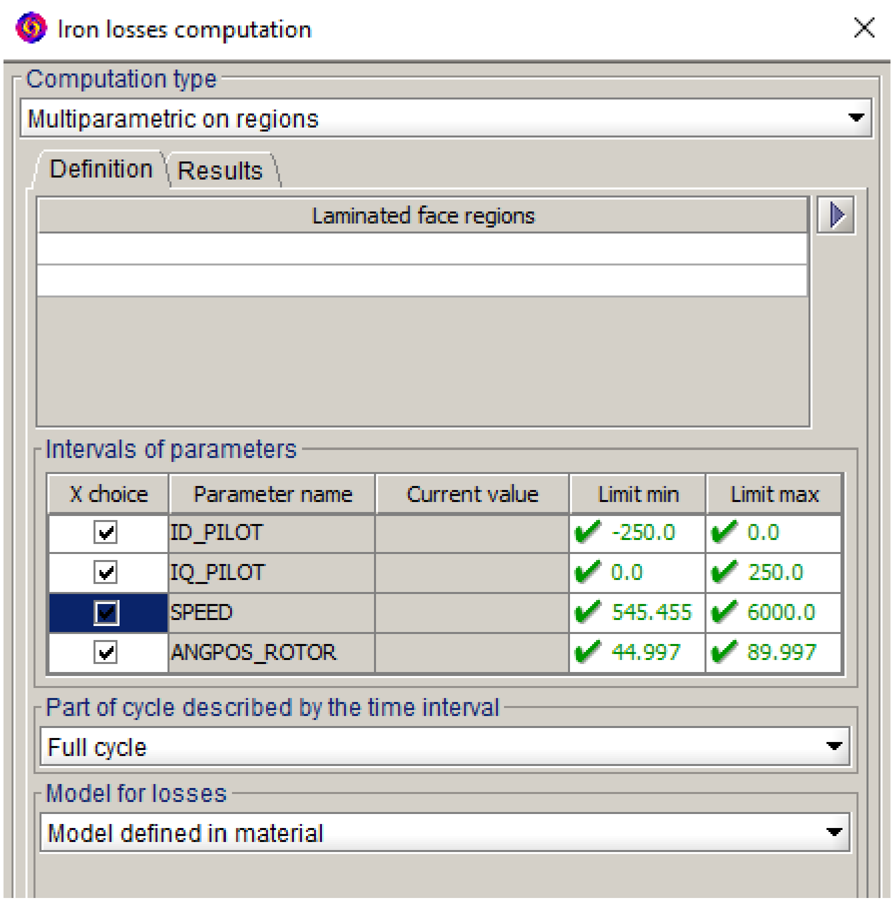
Isovalues are computed and displayed on the selected support in the graphic windows.

# 2.2.4 Iron losses multiparametric computation

Iron losses multiparametric computation enables to compute modified Bertotti iron losses or LS iron losses on a selection of laminated regions. The iron losses model can be defined on the material or in the iron losses computation box in post-processing. The computation gives instantaneous losses for several sets of geometric parameters or I/O parameters.

## Steps for iron losses multiparametric computation

This section describes the steps to carry out multiparametric computation for iron losses:



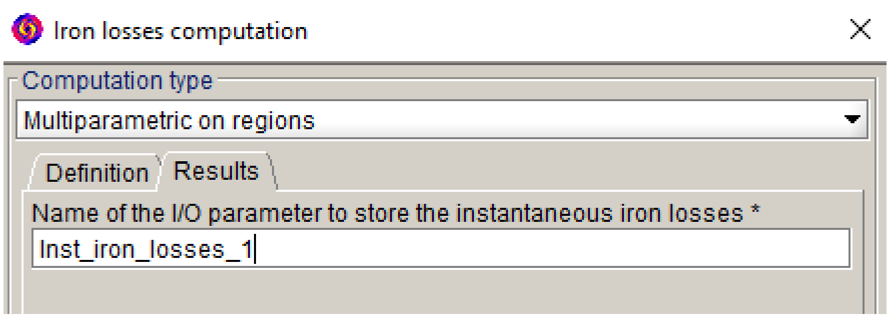

1. Choose a **Computation type**: **Multi-parametric on regions**
2. Select laminated regions for iron losses computation
3. Choose:
   - Intervals of values for geometric parameters and I/O parameters after having checked the corresponding boxes (or fixed parameters values)
   - In transient magnetic: time interval to obtain a time period. This choice is linked with the field **Part of cycle described by the time interval**
4. In Transient magnetic, define the time period select in the previous step. The various possibilities are presented in the table of page: Steps for iron losses computation on regions

**5.** Choose the model for losses :
- Model defined in material
- Modified Bertotti: Enter the coefficient values and modify exponents if necessary.
- LS predefined sheets (in transient magnetic, in Advanced mode): Enter the sheet iron
- LS defined by importing a MILS file (in transient magnetic, in Advanced mode)

# Results of iron losses multiparametric computation

## Computation results

The multiparametric computation of iron losses gives the following results




## I/O Parameters

I/O Parameters called "Inst_iron_losses_i" is stored in the tree, in the area **Parameter/Quantity** > **Parameter I/O**

It represents instantaneous losses on the region(s) versus time, geometric and I/O parameters.

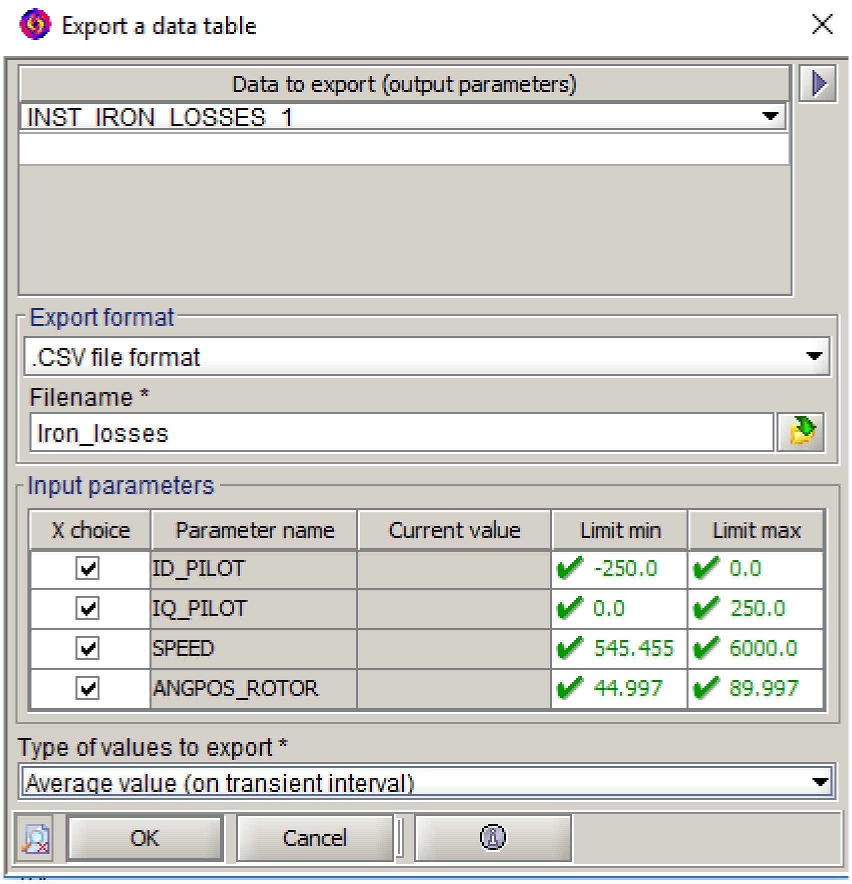
## Post-processing iron losses

It is possible to use the previous I/O parameter in order to display iron losses computation.

For instance, it is possible to display a 3D curve of iron losses versus two parameters (time or geometric parameter or I/O parameter) via menu **Curve** > **3D curve (2 I/O parameters)** > **New 3D curve (2 I/O parameters)**. I/O Parameter called "Inst_iron_losses_i" and containing loss values can be accessed in formula editor (Click on "f()").

## Export iron losses

It is possible to export iron loss values via menu **Data exchange** > **Export quantity** > **Export data table** :




For this:

- Select "Inst_iron_losses_i" I/O Parameter that has been created during iron loss computation
- Choose export format as well as the name of the file
- Select intervals corresponding to computation intervals
- Select types of values to be exported. For example:
  - **Average value (on transient interval)**. Remark: this is a statistical mean.
  - **Current value**
- click on **OK**

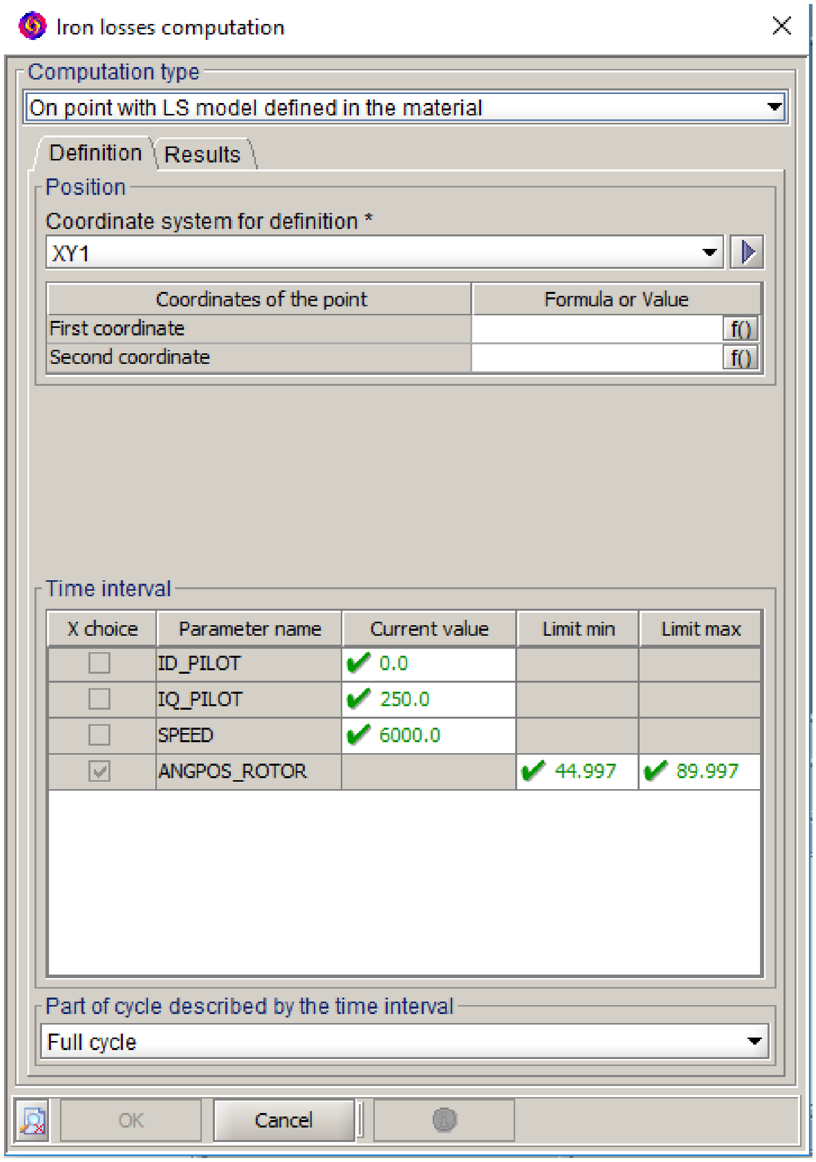
# 2.2.5 LS iron losses computation on point (Advanced)

Iron losses computation on point with LS model defined in the material enables to compute LS iron losses on a point belonging to a laminated region. Iron losses model must be defined in the material. The computation gives instantaneous losses volume densities and averaged losses volume densities over a period for a set of fixed geometric parameters or I/O parameters. This command is the only one that enables to have B(t) and H(t) on a time period, and to plot an hysteresis cycle.

**Caution** : LS iron losses computation on point is only available in Flux advanced mode.

## Steps of LS iron losses computation on point

This section describes the steps to carry out LS iron losses computation on point:



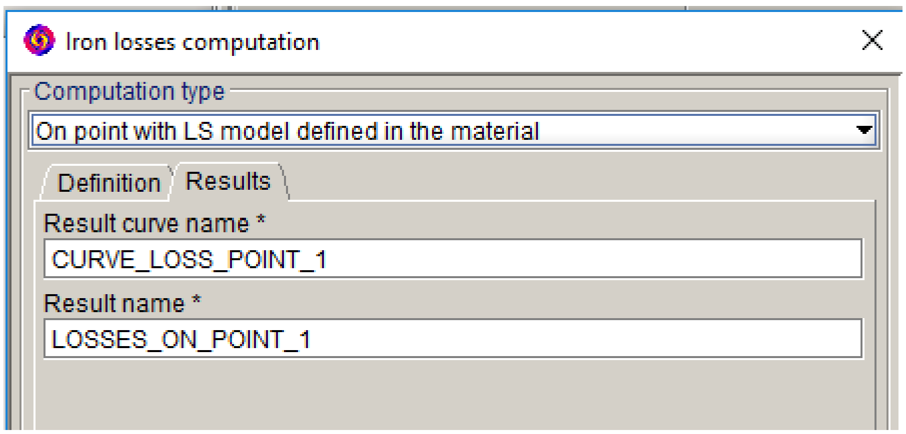

1. Choose a **Computation type**: **On point with LS model defined in the material**

2. Enter the coordinates of the computation point belonging to a laminated region
3. Choose:
   - fix values for geometric parameters and I/O parameters
   - time interval to obtain a time period. This choice is linked with the field **Part of cycle described by the time interval**
4. In Transient magnetic, define the time period selected in the previous step. The various possibilities are presented in the table of page: Steps for iron losses computation on regions

# Results of LS iron losses computation on point

### Computation results

The computation of LS iron losses on point gives the following results




### Curve

"CURVE_LOSS_POINT_i" curve can be displayed automatically and is stored in the tree in the area **Postprocessing** > **Curve** > **2D Curve (I/O Parameter)**

It contains three curves:

- volume density of instantaneous losses on the point versus time "DVOL_LOSS_LS_INST"
- Magnetic flux density on point versus time "BMAG1_LS”
- magnetic field on point versus time "HMAG1_LS”

Remark: the last two curves show the hysteresis cycle on the considered point.

### Result

The results are automatically displayed in the data tree in the area **Postprocessing** > **Result** > **Iron losses Result**.

The results include:

- Volume density of average iron losses over a period on the point (in W/m3)
- Volume density of iron losses energy over a period on the point (in J/m3)

# 2.2.6 Deprecated version

### Bertotti iron losses

This command is available via menu **Computation** > **Computation of iron losses** > **Deprecated version** > **Bertotti iron losses** and enables to compute iron losses with Bertotti model with the following equation:

$$P_{fer}^{(MH)} = k_h B_{max}^{\alpha_h} \ f^{\beta_h} + \frac{\sigma d^2}{12} \ 2\pi^2 \ B_{max}^{\alpha_c} \ f^{\beta_c} + 8.76 \ k_e \ B_{max}^{\alpha_e} \ f^{\beta_e}$$

$$p_{fer}^{(MT)}(t) = k_h B_{max}^{\alpha_h} \ f^{\beta_h} + \frac{\sigma d^2}{12} \ \left|\frac{dB}{dt}\right|^{\alpha_c} + k_e \ \left|\frac{dB}{dt}\right|^{\alpha_e}$$

The user should enter the following parameters:

- coefficients and exponents,
- Information about lamination: fill factor kf, lamination thickness d and information dealing with lamination direction.

### LS iron losses on regions

This command is available via menu **Computation** > **Computation of iron losses** > **Deprecated version** > **LS iron losses on regions**.

The user chooses a Type of iron sheet among the proposed list. This menu enables to compute losses thanks to former LS models of Flux.

### LS iron losses at a point

This command is available via menu **Computation** > **Computation of iron losses** > **Deprecated version** > **LS iron losses at a point**.

This command is the same as the former one, but the computation is at a point instead of a region.

# 2.3 Data Export to HyperView and HyperGraph

### Introduction

Export to HyperView enables to view and postprocess Flux results in HyperView or HyperGraph during or after the solving process. This permits to benefits from HyperView or HyperGraph advanced postprocessing capacities.

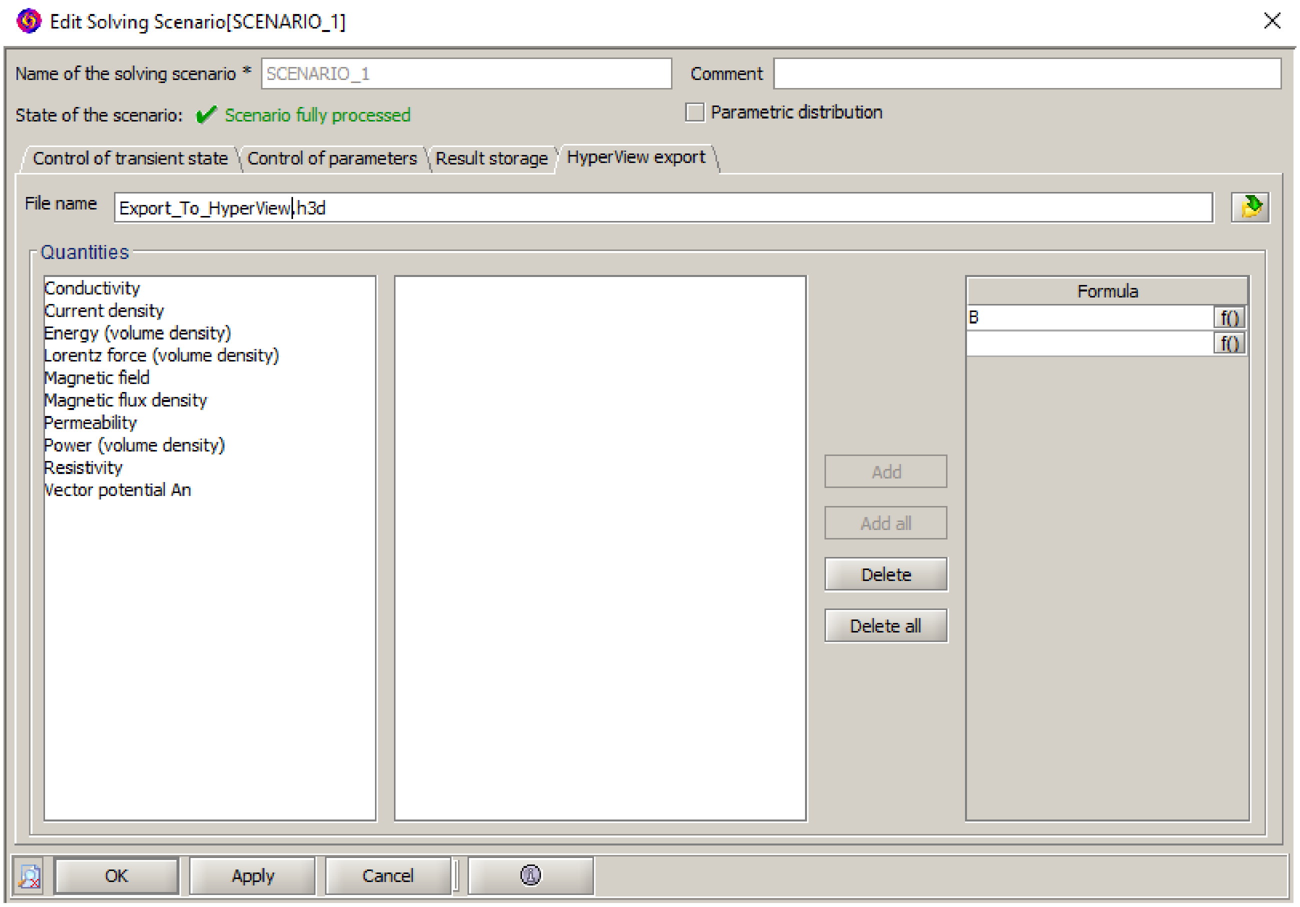
Export function can be accessed by scenario box. This process generates an .h3d file.

**Note:** The user has also the possibility to export results to HyperView from Flux **Import/ Export context** after the solving process, even if such export is more dedicated to multiphysic couplings with OptiStruct, AcuSolve, etc. (Please refer to the "Data Import/ Export Context" chapter for more information). In order to help the user, the two workflows are compared below:

| | Export via scenario | Export via Import/Export context |
|---|---|---|
| Aimed softwares | HyperView / HyperGraph | OptiStruct / AcuSolve/ ... |
| Support | Full domain of the project | Imported mesh / Flux entities of the same mechanical set |
| Global quantity | One or several | Only one |
| Scenario step | All steps | Single step or all steps selection |
| Visualization | During of after solving process | After solving process |

### Export of .h3d file in Flux

Export function can be accessed via solving box, in **HyperView export** tab:



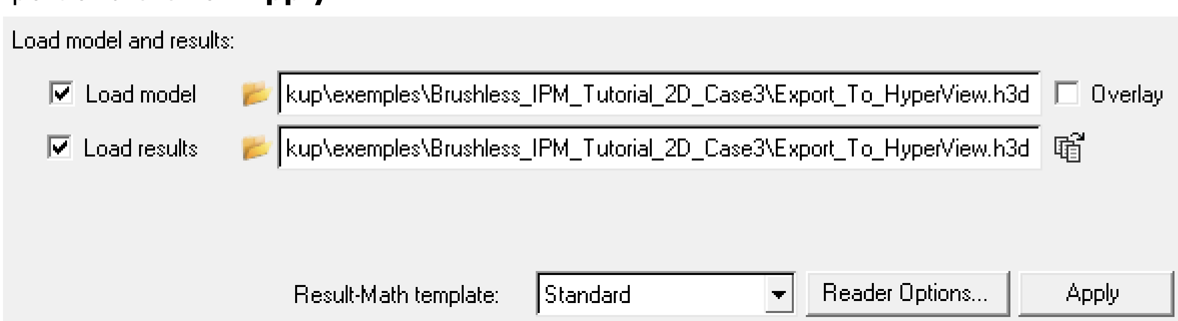

The steps of the .h3d file export are described below:

1. Chose a name for the .h3d file that will contain the data
2. Chose the quantities to be computed on the device

> **Note:** HyperView export can more or less decrease the speed of the solving process, depending on the project size.

## Import of .h3d file in HyperView

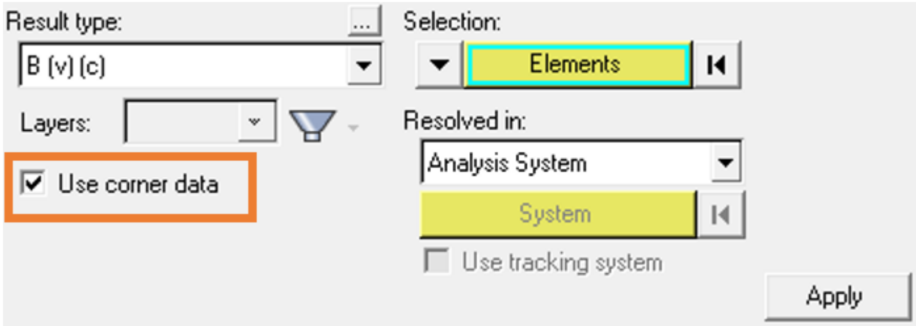
The .h3d file generated in Flux is imported in HyperView via menu: **File** > **Load** > **Results**. Chose the file to import and click on **Apply**.



> **Note:** During solving process, the results are not automatically updated in HyperView. In order to update them, click again on the previous **Apply** button.

A few tips are presented below in order to view the results:

- In order to view the results computed by Flux on nodes, **Use corner data** option must be checked:




- In order to visualize movement in HyperView, chose **Set Transient Animation Mode** via the following icon:

## Limitation

HyperView export is not possible in the following cases:

- Scenario with distribution
- Remeshing during solving process, ie.:
  - Geometric parametric scenario
  - Adaptive solver
  - Compressible mechanical set In such case, the export is realized on the device with the exception f the region including the compressible mechanical set.

# 2.4 Flux - HyperStudy Coupling

## Introduction

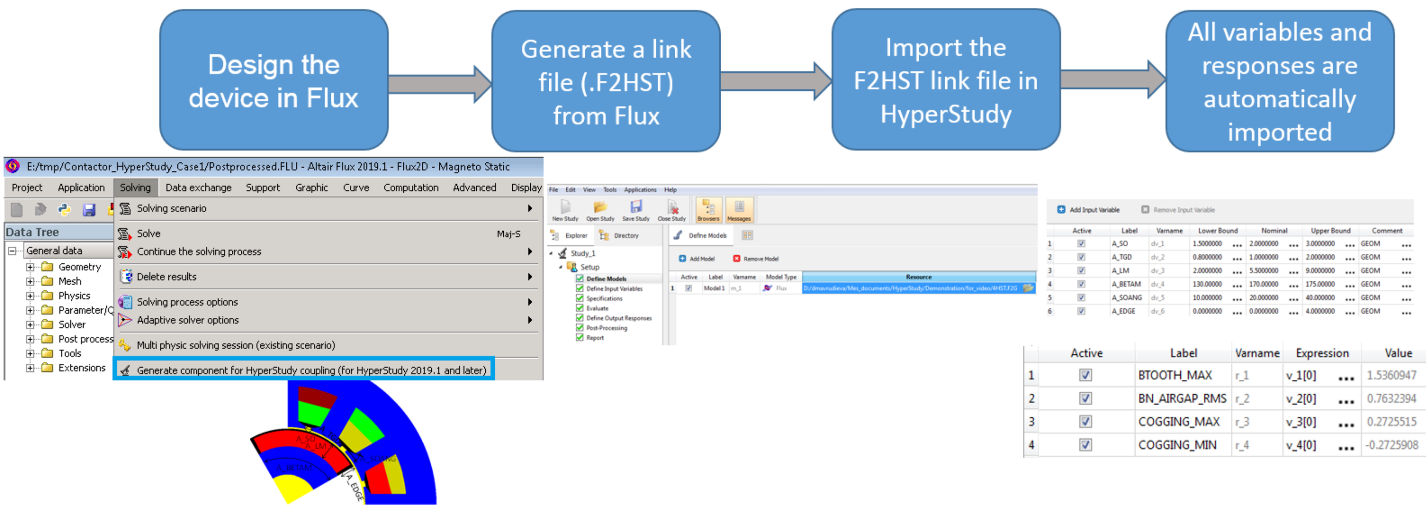
HyperStudy is a multi-disciplinary design exploration, study, and optimization software for engineers and designers.

It is interesting to couple it with Flux to leverage the HyperStudy approaches (DOE, Fit, Optimization and Stochastic) for Flux models design exploration and optimization.

The HyperStudy to Flux connection allows exploring efficiently the design space of any electromagnetic devices modelled in Flux (motor, actuator, transformer…)

## Workflow

The general workflow is:

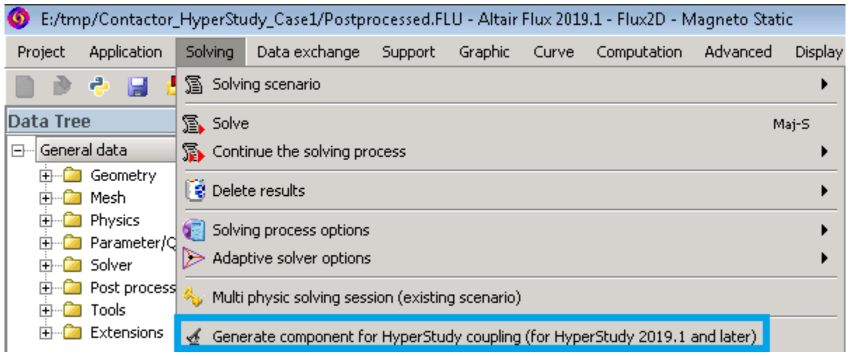



## In Flux

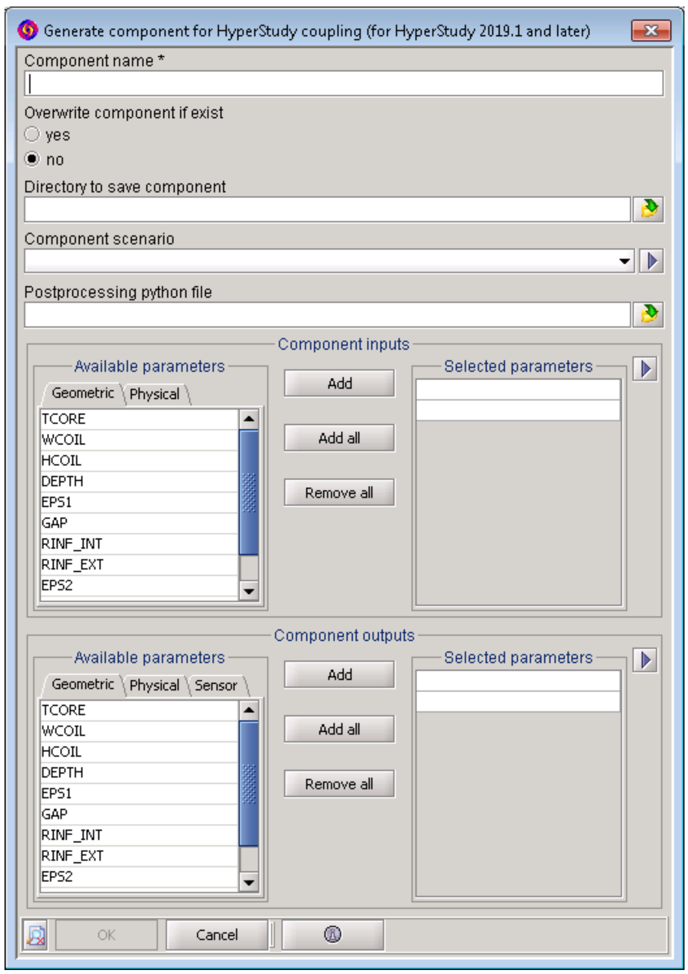
From Flux 2019.1, generate the component for HyperStudy coupling:

**Solving** > **Generate component for HyperStudy coupling**

> **Note:**
>
> - This command **Generate component for HyperStudy coupling** is compatible with Flux and HyperStudy 2019.1 and later versions.
> - This command **Generate component for HyperStudy coupling** replaces the previous command **Generate component for GOT-It coupling**.




When selecting this command, the following dialog window is displayed:




1. Define the name of the component

   For example: `4HST`
2. Choose to overwrite the component or not, if already existing
3. Optionally, choose the directory to save the component

   If nothing is selected, this directory will be the current working directory
4. Optionally, select a scenario

> **Note:** In transient applications, selecting a scenario is mandatory.

5. Optionally, select a postprocessing python file in order to automatically postprocess results

> **Note:** In transient applications, it is generally used to manage curves (extract values).
>
> Here is an example of a postprocessing python file:

```
# Delete previous curves

CurveVariation2D[ALL].delete()
CurveSpatial2D[ALL].delete()

# Load a macro, which allows extracting values from curves and create/update I/O parameters to store the values

import os

INSTALL = os.environ["INSTALLFLUX"]+ os.path.sep + ".."
macroFile = INSTALL+"/Extensions/Macros/Analyse2DCurve.PFM"
loadMacro(fileName=macroFile)


# Create curves

EvolutiveCurve2D(name='COGGING',
                 evolutivePath=EvolutivePath(parameterSet=[SetParameterXVariable(paramEvol=VariationParameter['TIME'],
                                                                                  limitMin=0.0,
                                                                                  limitMax=25.0)]),
                 formula=['TorqueElecMag(ROTOR_SET)'])


SpatialCurve(name='BN_AIRGAP',
             path=Path['PATH_GAP'],
             formula=['Comp(1,VLCS(ROTOR_COORD,B))'])


# Execute the macro to extract values from the curves and store/update I/O parameters

Analyse2DCurve(curve2D=Curve2d['COGGING'])
Analyse2DCurve(curve2D=Curve2d['BN_AIRGAP'])
```

6. Select the input parameters:
   - Geometric parameters (defined with a numerical value, not a formula)
   - Physical parameters: I/O parameters (controlled by a scenario, not a formula)

> **Note:** Only the parameters available for selection are displayed.

HyperStudy will change the values of the selected input parameters (ranges are specified in HyperStudy) in order to explore the design (for instance, find the optimal values with respect to specific goals and constraints).

7. Select the output parameters:
   - Geometric parameters
   - Physical parameters: I/O parameters
   - Sensors

> **Note:** Only the parameters available for selection are displayed.

HyperStudy will recover automatically the results of the selected output parameters.

**8.** Click on the **OK** button

The following files are created in the working directory:

- the Flux project (which has been duplicated): `*.F2HST.FLU`
- the link file (which has been created): `*.F2HST`

In our example, the component name is `4HST`, and in the working directory, the following files are created:

- `4HST.F2HST.FLU`: duplicated Flux project
- `4HST.F2HST`: link file

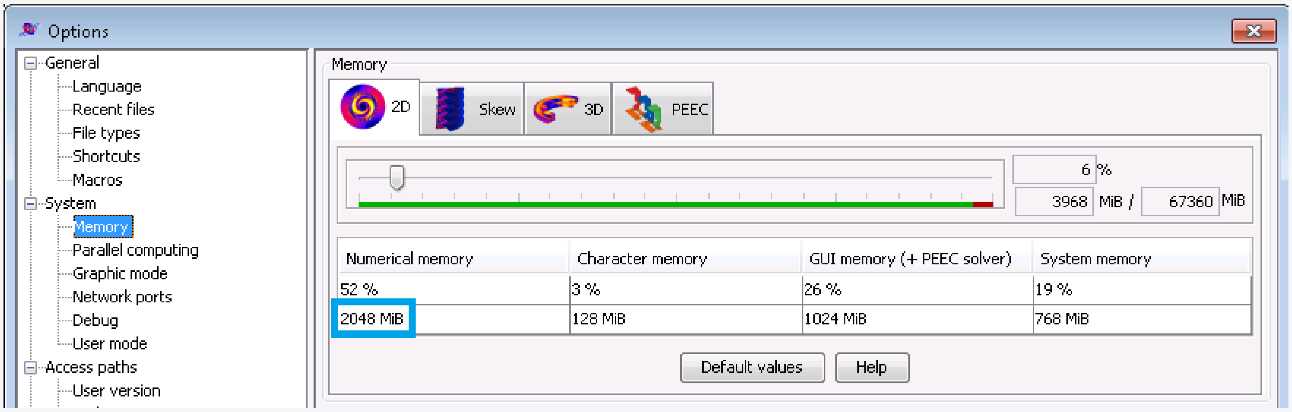
4HST.F2HST

```xml
<?xml version="1.0" encoding="UTF-8"?>
<HW_PDD schema="altair_pdd" schemaVersion="hstp_v_3">
   <!--
      COPYRIGHT (c) 1996-2019 Altair Engineering, Inc.
      Email: hstsupport@altair.com
      Phone: +1 (248) 614-2400
      http://www.altair.com
   -->
  <DocumentProperties>
     <FileDate>2019-05-07 16:55:41</FileDate>
     <Revision>0.0</Revision>
     <Author></Author>
     <System>Windows(64 bits)</System>
     <CreatedBy>Flux 2019.0</CreatedBy>
  </DocumentProperties>
  <ConnectionList>
     <Connection>
        <ItemList>
           <Item label="TCORE" name="" usage="input">
              <State>true</State>
              <Comment>GEOM</Comment>
              <ModelParameter>TCORE</ModelParameter>
              <InitialValue>20</InitialValue>
              <LowerBound>18</LowerBound>
              <UpperBound>22</UpperBound>
              <DataType>Real</DataType>
              <DataMode>Continuous</DataMode>
              <LevelList></LevelList>
              <ValueList></ValueList>
```

**Warning:**

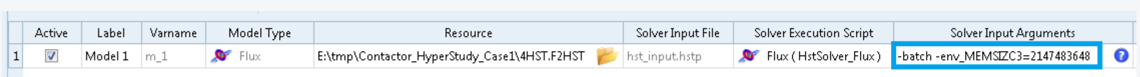
The memory values specified manually or by default in the Flux Supervisor options are written in the `F2HST` file when generating the component for HyperStudy from Flux. By default, HyperStudy will launch Flux with these values.

If these values are unnecessary high, it can limit the multiple Flux launching from HyperStudy. So, before generating the component, a suggestion is to tune the memory values to the minimum required to solve your model, in the **Flux Supervisor options**:




Note that if you change the memory values in the Flux Supervisor options, after generating the component, the new values will not be communicated to HyperStudy.

If you would like to change the memory values after generating the component, you can do it directly in HyperStudy using the solver arguments for Flux in the **Solver Input Arguments** field:



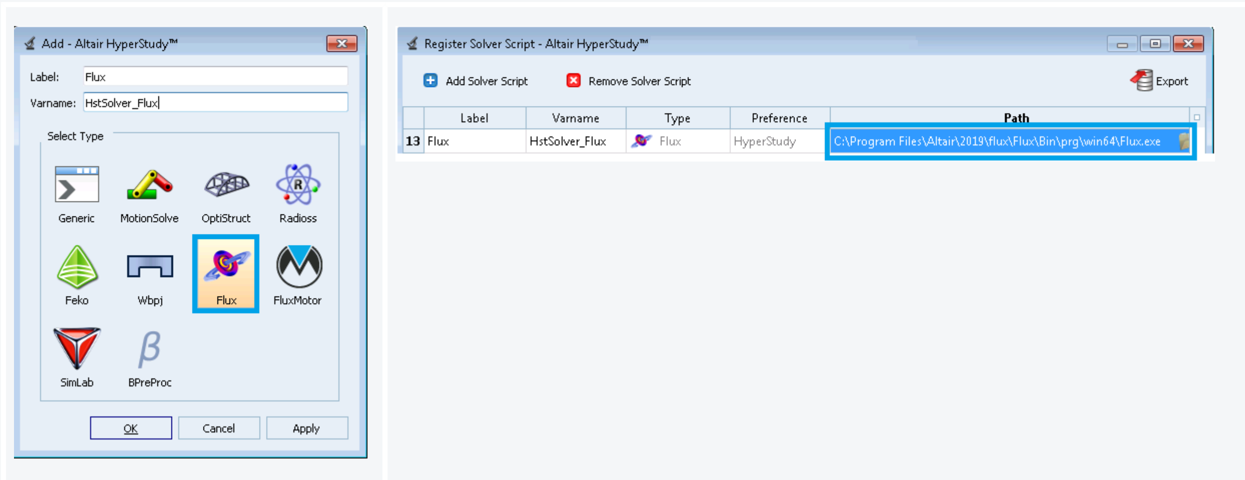
## In HyperStudy

**Attention:** For the first use of HyperStudy with Flux, you need to define the solver installation path

i.e. on Windows: `<Flux installation directory>\flux\Flux\Bin\prg\win64\Flux.exe`

(by default `<Flux installation directory>=C:\Program Files\Altair\2019` if Flux 2019.1 was installed on Windows with an administrator account),

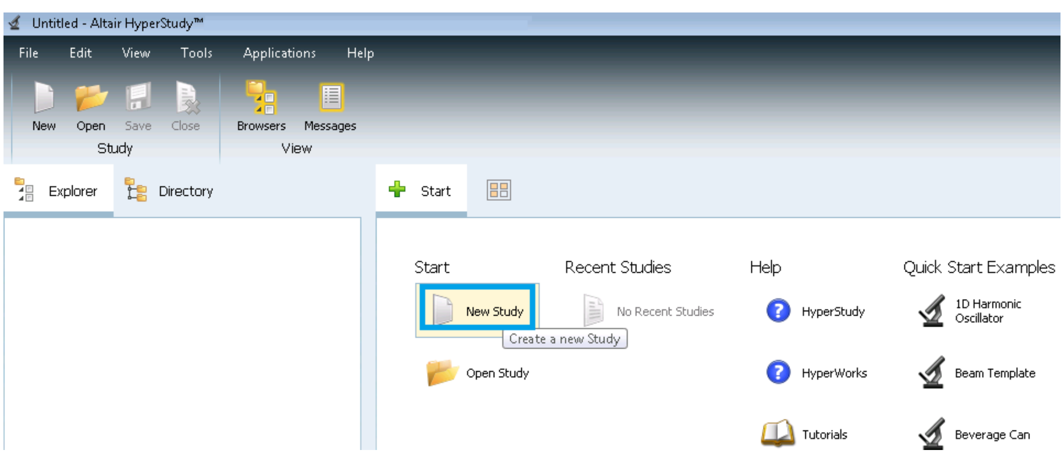
selecting the menu **Edit** > **Register Solver Script** > **Add Solver Script** button



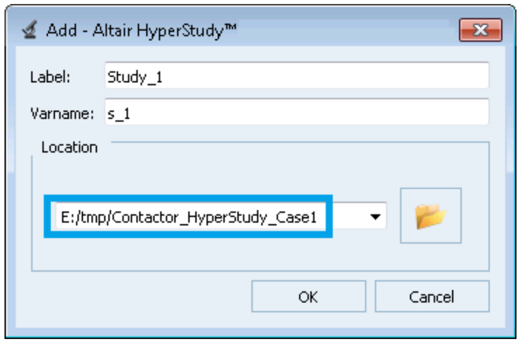

In HyperStudy, start a new study and create a connection with Flux model:

1. a. To start a new study:

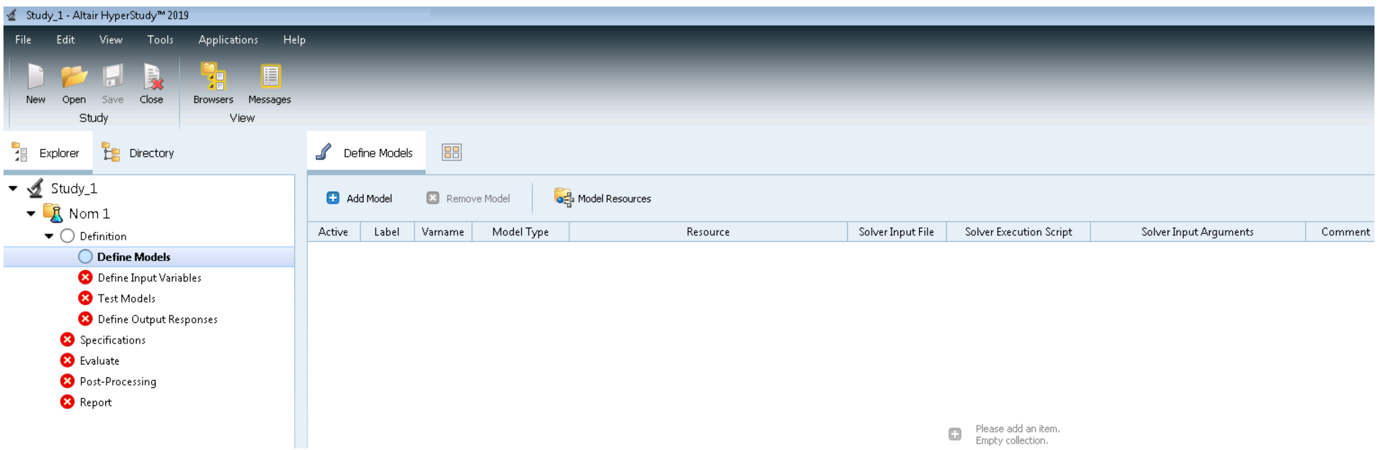
   Click on **Start** > **New Study**




   b. In the **Location** field, select the directory in which the Flux project `*.F2HST.FLU` and the link file `*.F2HST` are located




The new study is created:

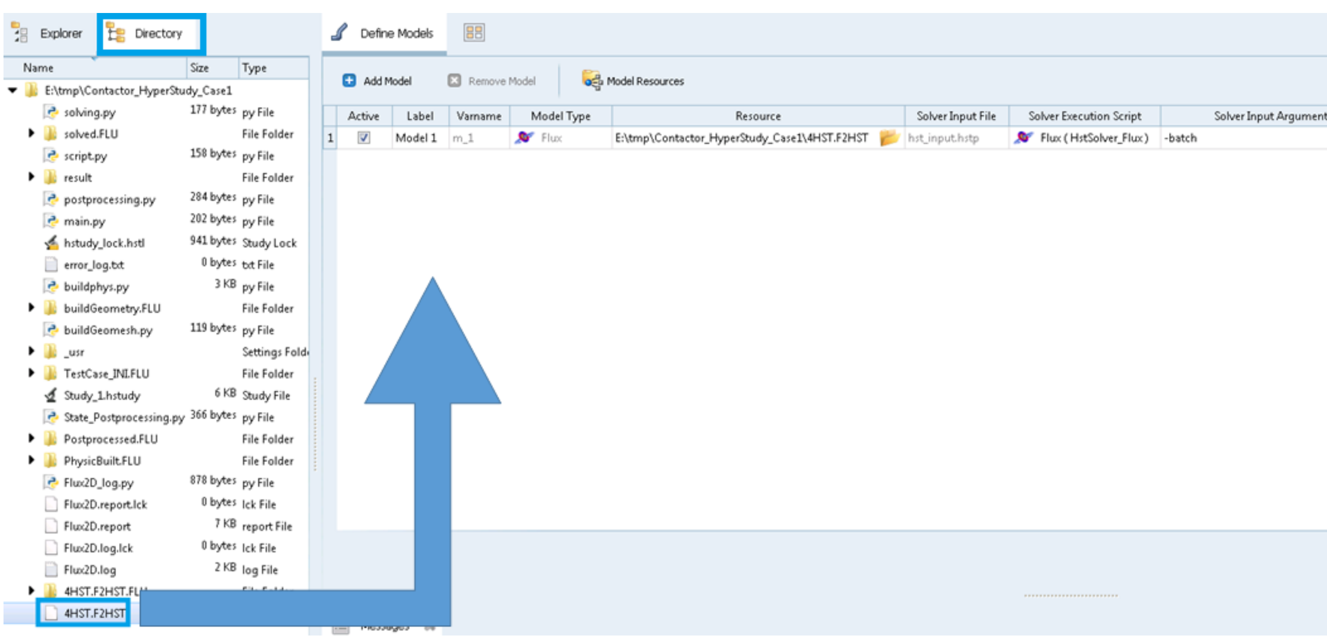



**2.** To create a connection with Flux model, there are two possibilities:

- A possibility is:

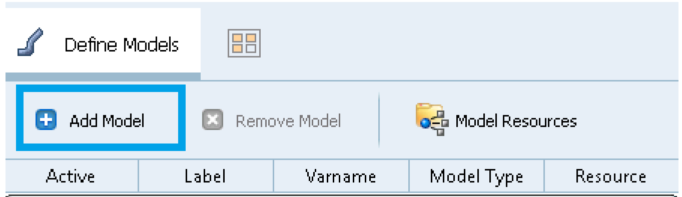
  Click on **Directory**, and **Drag & Drop the F2HST file** into the graphical interface.

  It creates automatically a Flux model and populates the fields.



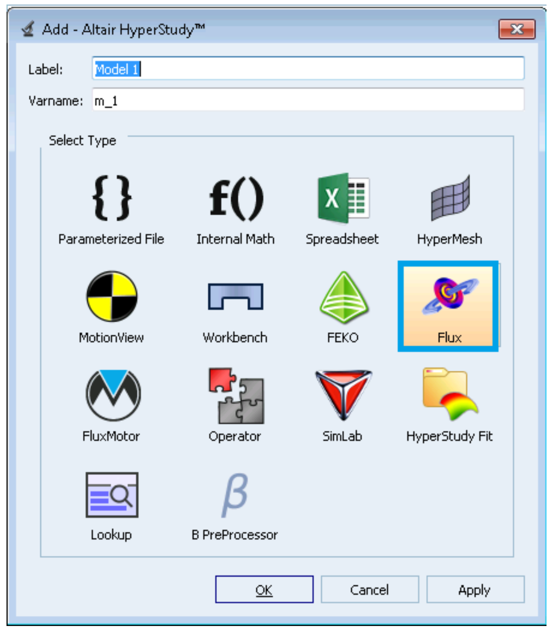
- Another possibility is:

**a.** Click on **Define Models** > **Add Model**



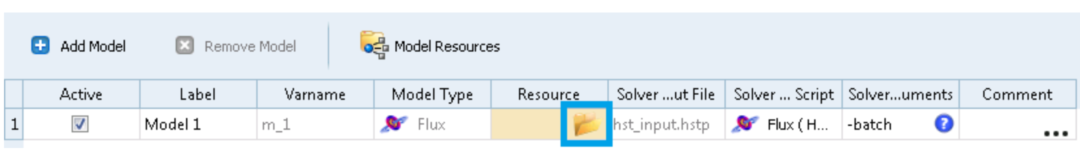

**b.** **i.** Select **Flux**

**ii.** Click on the **OK** button

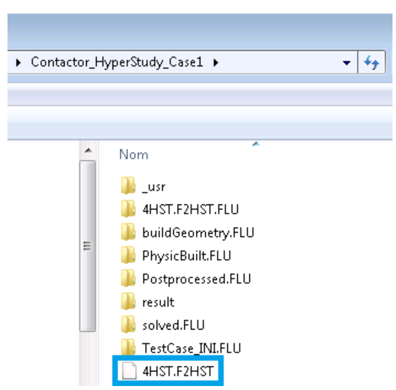



**c.** **i.** In the **Resource** field:

Click on 



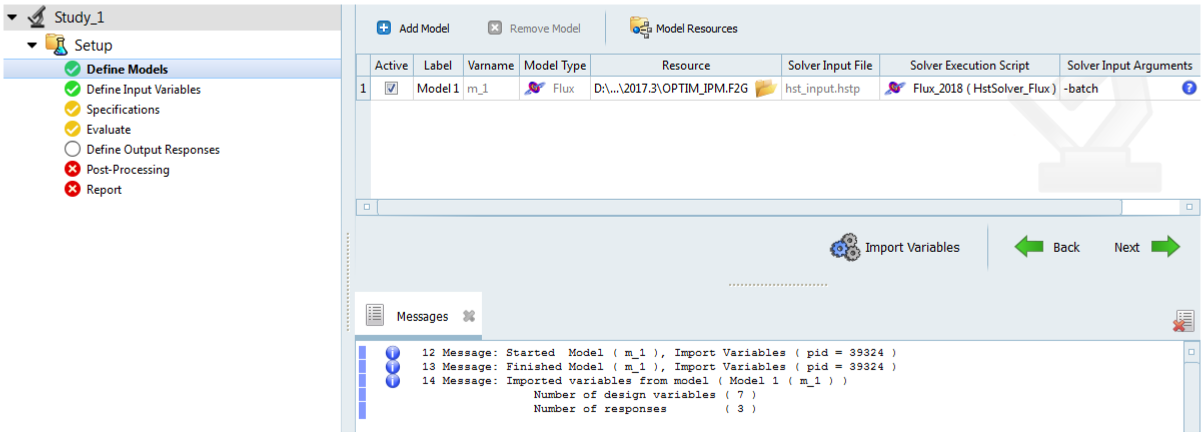

**ii.** Browse the directory and **select the F2HST file**



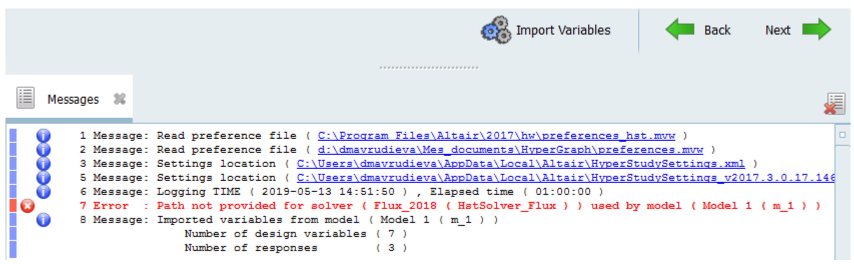
**3.** Whatever the creation option used, once the Flux model has been added:

Click on the **Import Variables** button (at the bottom right)

Your screen should look like this:




Note that if the Flux installation path has not been defined in HyperStudy, the following error message is displayed:




in this case, please refer to the Attention at the beginning of this section In HyperStudy.

For more details regarding HyperStudy use, please refer to HyperStudy user guide.

# 2.5 Flux e-Machine Toolbox

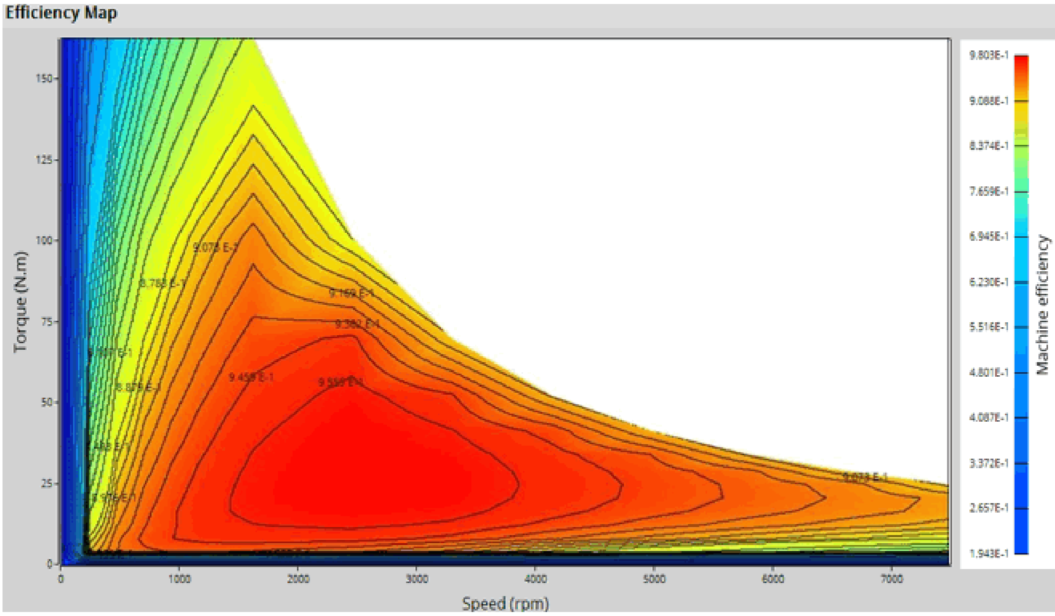
**Flux e-Machine Toolbox**is a new application in Flux dedicated for Electric Machine. This tool permits to compute and postprocess results dedicated for e-Machine like **efficiency maps**. It's a first version. It is planned to enrich the possibilities of postprocessing in future versions.

### Content

This chapter deals with following topics:

# 2.5.1 Flux e-Machine Toolbox: About

### Introduction




Starting from the 2019.1 version, Flux has a new tool called **Flux e-Machine Toolbox**. This coupling is available in 2D, 3D and Skew for the magnetic transient application.

The aim of Flux e-Machine Toolbox is to characterize the behavior of the machine in the "Torque-Speed" area and allows to postprocess various quantities in this area.

Input parameters like the **maximum phase voltage RMS** , the **maximum phase current RMS** and the desired **Maximum speed** of the machine are considered.

Two types of command modes are available: The **Maximum Torque Per Volt** command mode (MTPV) and the **Maximum Torque Per Amps** command mode (MTPA).

### Type of machine

In this first version, only one machine is taken into account by Flux e-Machine Toolbox. It is the Synchronous machine, permanent magnet in three phases (Inner rotor or outer rotor) with "integral slot-winding" (the stator must have the same number of periodicities as the number of poles pair in the rotor).

New types of machine will be implemented in next versions.

### About losses

Different losses are computed and postprocessed:

- **Stator Winding Joule Losses**: on each coil conductor
- **Rotor Eddy Current Losses**: on solid conductor regions (magnet, fret, can,…)

- **Iron losses**:
  - In 2D and 3D:
    - Only computed on laminated non-conducting magnetic regions
    - In Flux, it is advised to choose a material with losses, assigned to these regions (the user has the choice to use LS or Bertotti model)
    - If losses are not defined on the material, default Bertotti coefficients are automatically applied

      (K1=151.88 ; K2=0.07 ; K3=1.19 ; A1=2 ; A2=2 ; A3=1.5)
  - In Skew:
    - Only computed on non-conducting magnetic regions , because the laminated regions do not exist in Skew
    - Default Bertotti coefficients are automatically applied after the Flux solving process

      (K1=151.88 ; K2=0.07 ; K3=1.19 ; A1=2 ; A2=2 ; A3=1.5)

## Process

Here is the process of the Flux e-Machine Toolbox coupling:

| Step | Software | Description |
|---|---|---|
| 1 | Flux | Preparation of the Flux project:<br>• standard description: geometry, mesh and physics<br>• specific description: losses, number of poles, circuit information...<br>**Note:** A classic workflow is to use FluxMotor to pre-design the electrical machine and export it in a python srcipt. Then the python script is run in Flux to obtain a Flux 2D project ready to generate the coupling component |
| 2 | Flux | Generation of the Flux e-Machine Toolbox coupling component<br>**Note:** In the options of the generation of the coupling component, it is possible to choose to go directly at the Step 5. After the generation of the coupling component, Flux e-Machine Toolbox is launched (equal to step 3) and the generated component is directly opened (equal to step 4) |
| 3 | Flux Supervisor | Opening of Flux e-Machine Toolbox starting from Flux |
| 4 | e-Machine Toolbox | Opening of the Flux e-Machine Toolbox coupling component generated by Flux |

| Step | Software | Description |
|---|---|---|
| 5 | e-Machine Toolbox | Choice of input parameters |
| 6 | e-Machine Toolbox | Launching of the simulation<br>• The solving process of Flux is launched in batch mode<br>• Once the solving process is done, a postprocessing script is run to build the performance mapping |
| 7 | e-Machine Toolbox | Postprocessing of results |

# 2.5.2 Flux e-Machine Toolbox: Coupling component

### Introduction

To achieve a Flux e-Machine Toolbox coupling, it is necessary to generate a coupling component from the modeled Flux project previously prepared. There are several prerequisites to respect in the construction of the Flux project to ensure this coupling.

### Preparation of the Flux project

Here are the prerequisites:

- Coupling available only in transient magnetic with 3-Phase circuit.
- I/O parameter piloted by a scenario, called SPEED and used in mechanical set in rotation with imposed speed.
- I/O parameter defined by formula, called POS_INI and used in mechanical set in rotation for the time position t=0s.
- A parameter with geometric type for the definition of the pole number.
- No need to define a solving scenario, Flux e-Machine Toolbox will do it automatically when running a test.
- For the Iron losses computation, it is mandatory to use non conducting laminated magnetic regions for the Rotor and the Stator (valid for 2D and 3D only).

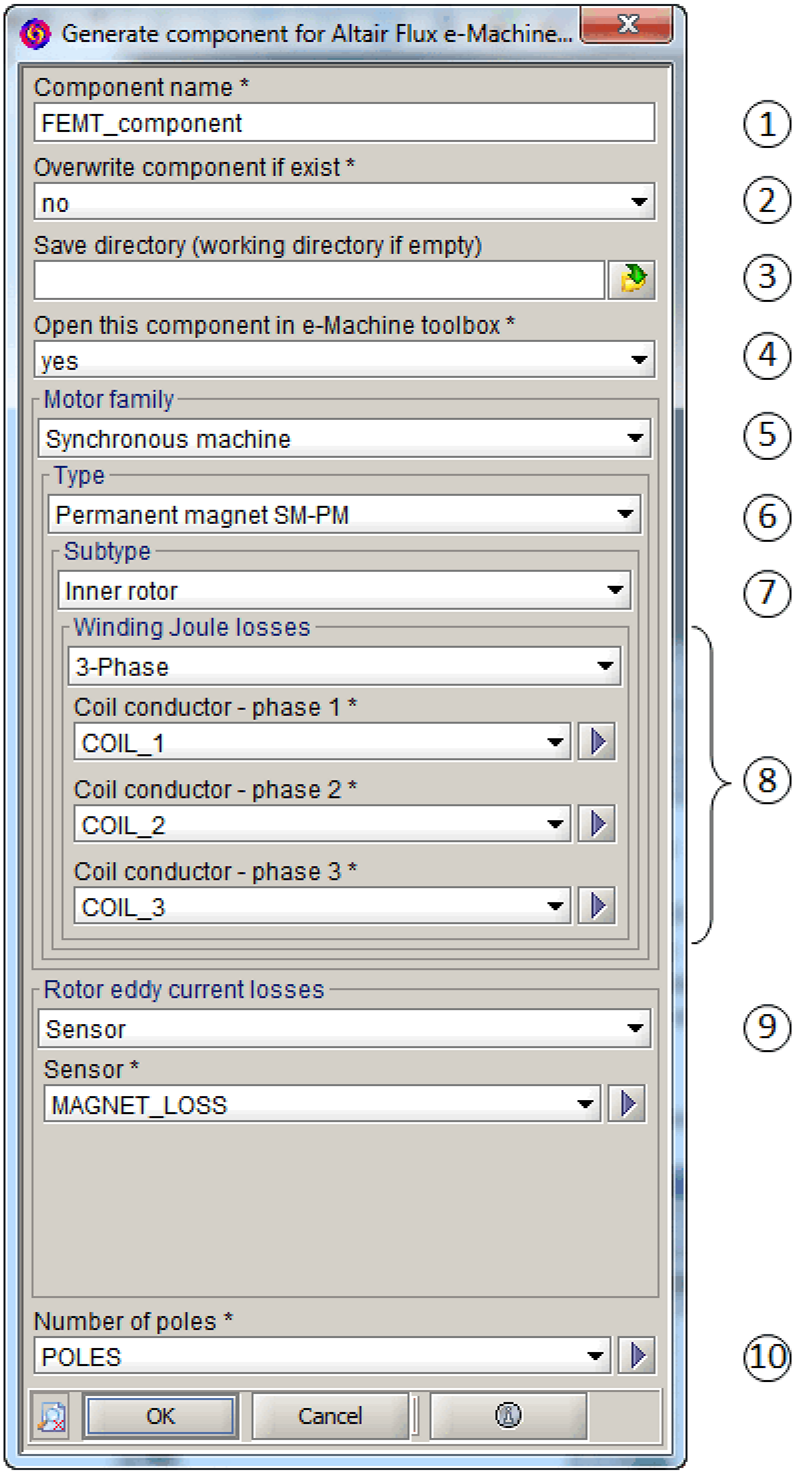
  > **Note:** In Skew, the non conducting laminated magnetic regions are not available. It is advised to use non conducting magnetic regions.

- For the Iron Losses computation, it is mandatory to define a material with losses model, assigned to non conducting laminated magnetic regions.
- It is mandatory to define in the application the transient initialization option to **initialized by static computation** in order to be able to compute the losses directly on the first period and to limit the number of steps to compute.
- **Only one pole section should be represented** (the machine should have integral-slot winding). Therefore, **a periodicity must be defined** in order to perform the machine simulation.

### Coupling component

The coupling component is necessary to ensure the transfer of information of the Flux project to Flux e-Machine Toolbox. This component is a file *.FEMT.

To access the generation of the coupling component, click on the menu **Solving** and click on **Generate component for Flux e-Machine Toolbox coupling**:




1. Name of the component saved in a directory **Name_of_the_component.FEMT**
2. Overwrite or not the component if it already exists
3. Save directory of the component (by default it's the current working directory)
4. Choice to launch or not Flux e-Machine Toolbox after the generation of the component and open the generated component
5. Motor family (only one available for the moment, **Synchronous machine**)

6. Motor type (only one available for the moment, **Permanent magnet SM-PM**)
7. Motor Sub-type (**Outer rotor** or **Inner rotor**)
8. Information needed to compute stator winding Joule losses
    a. Number of phases (3 phases only available for the moment)
    b. The coil conductor of each phase
9. Information needed to compute Rotor Eddy current losses
    a. Either a previously defined sensor, which computes Eddy current losses on magnet rotor regions
    b. Or the list of regions on which the rotor Eddy current losses must be computed (a sensor will be created automatically)
    c. Or the list of solid conductors on which the rotor Eddy current losses must be computed
10. Number of poles of the electrical machine (a geometric parameter is mandatory)

### Generated files

By generating the XXX.FEMT coupling component a XXX.FEMT directory is created in which there are:

- The saved Flux project **project.FLU**
- A directory **tests** in which tests run in Flux e-Machine Toolbox will be stored
- A file **config.xml** that contains all the information needed for the computation of perfomance mapping of the studied machine

# 2.5.3 Flux e-Machine Toolbox: The application

### Introduction

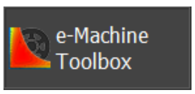
The application Flux e-Machine Toolbox allows to:

- Launch computations of performance mapping of an electrical machine starting from a coupling component generated in Flux
- Post-process these maps

### Launching of Flux e-Machine Toolbox

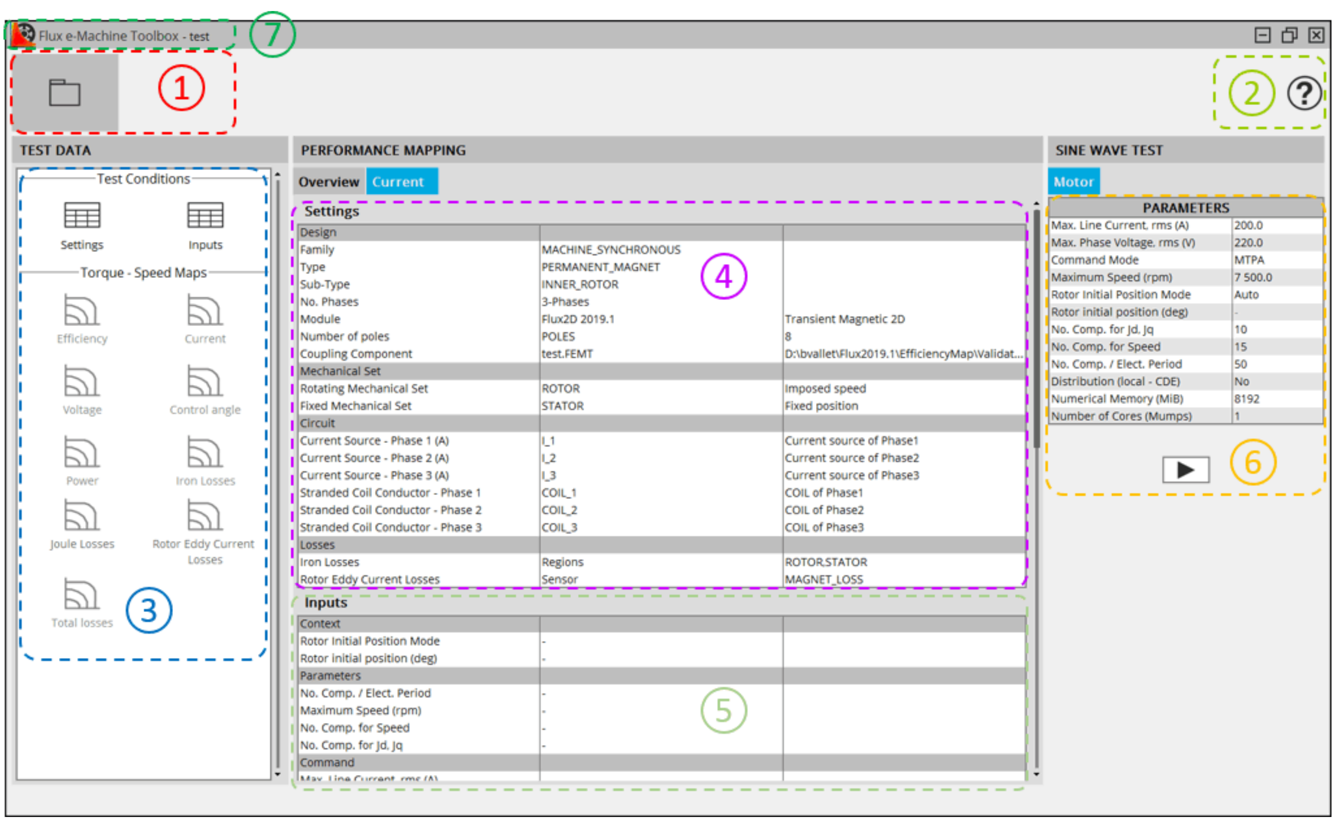
2 possibilities to access Flux e-Machine Toolbox application:

- Automatically after the generation of the coupling component (if the option has been chosen). The application is opened and the generated component is automatically opened
- Via the supervisor by clicking on:




### Environment

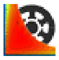
Here is the environment of Flux e-Machine Toolbox with an opened component without any test run.

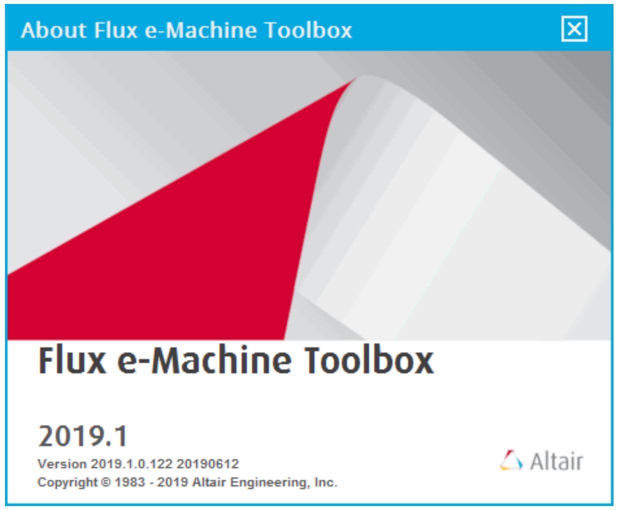



1. Allow the opening of a coupling component generated via Flux
2. Allow access to online documentation of Flux e-Machine Toolbox
3. Different filters for the display of the central zone (all filters are active once a test is done)

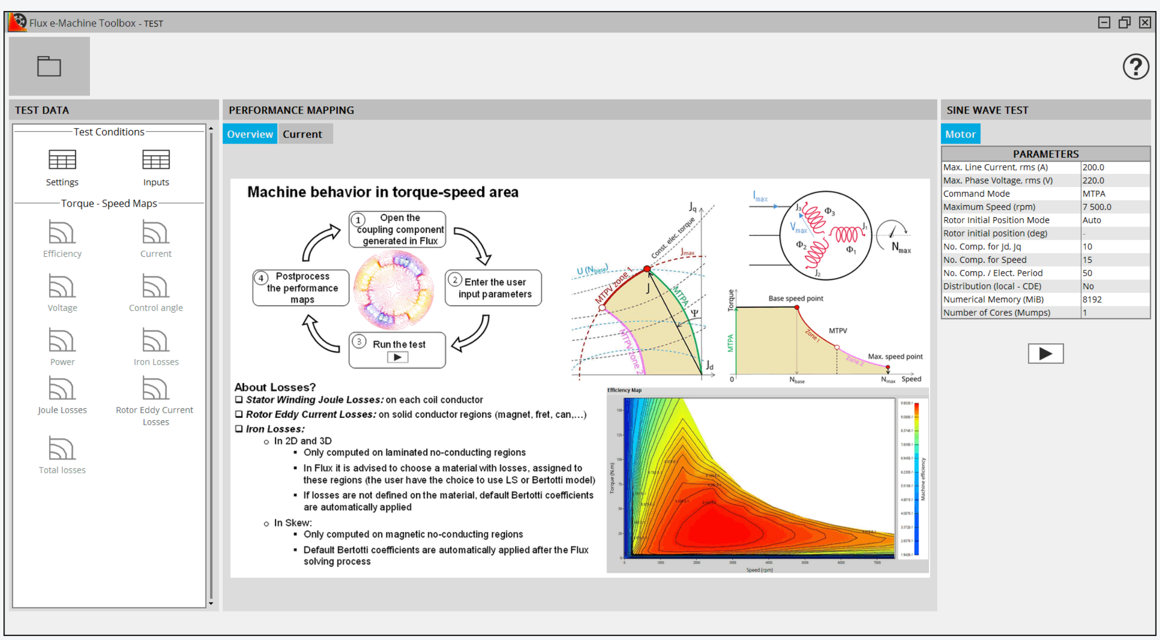
4. Display initial conditions which match the information in the configuration file of the coupling component
5. Display input parameters, chosen by the user, of the displayed test (empty if no test achieved)
6. Input parameters that the user must choose before running the test by clicking on:

7. Access on About box by clickin on the icon:






**Note:** The **Overview** tab corresponds to general information on the Flux e-Machine Toolbox application. This tab is independent of the opened component and independent of tests done.

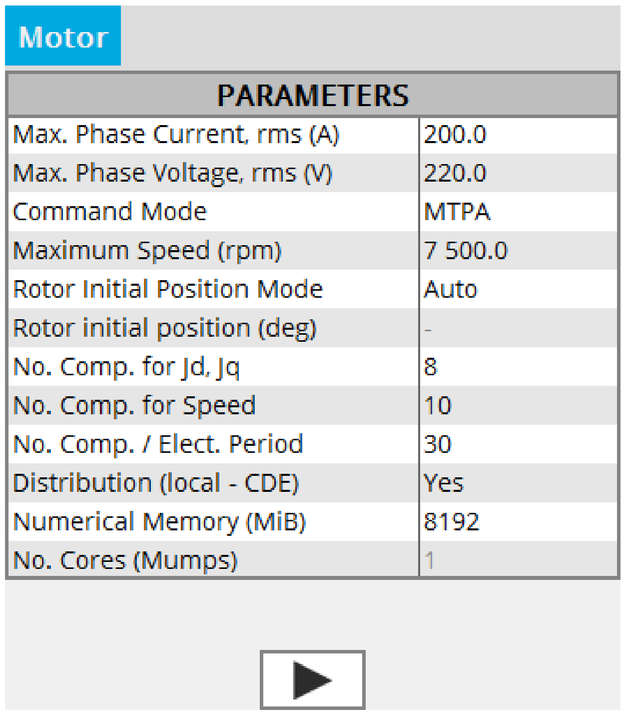


# 2.5.4 Flux e-Machine Toolbox: Input parameters

### Introduction

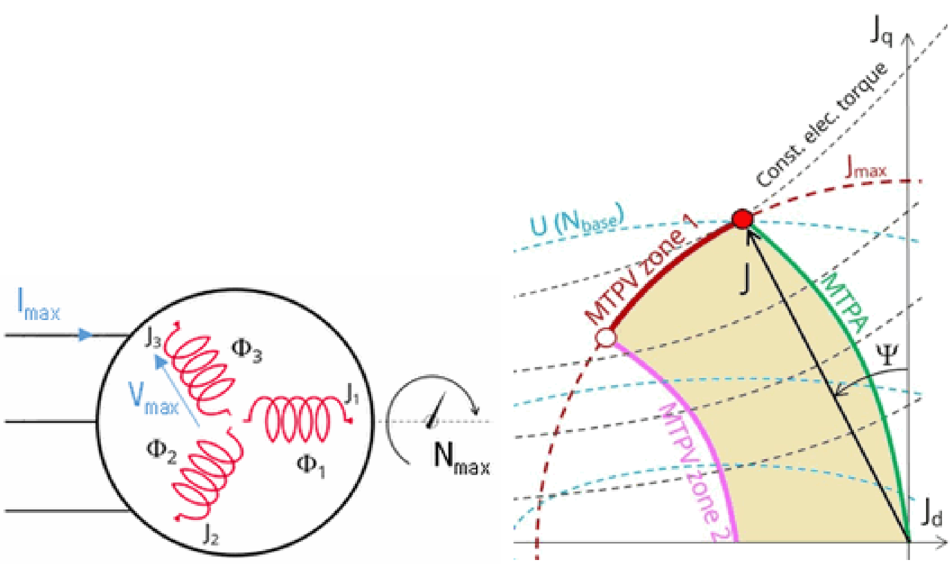
After having opened the coupling component in Flux, the user must define several input parameters befor running a test. These parameters allow to:

- pilot the solving of the Flux project
- specify solving options (distribution, memory)
- compute and build the different performance maps


Motor

| PARAMETERS | |
|---|---|
| Max. Phase Current, rms (A) | 200.0 |
| Max. Phase Voltage, rms (V) | 220.0 |
| Command Mode | MTPA |
| Maximum Speed (rpm) | 7 500.0 |
| Rotor Initial Position Mode | Auto |
| Rotor initial position (deg) | - |
| No. Comp. for Jd, Jq | 8 |
| No. Comp. for Speed | 10 |
| No. Comp. / Elect. Period | 30 |
| Distribution (local - CDE) | Yes |
| Numerical Memory (MiB) | 8192 |
| No. Cores (Mumps) | 1 |

### Input parameters : Wished data



- Maximum phase current RMS
- Maximum phase voltage RMS
- Command mode:
  - either **MTPV**: the Maximum Torque Per Volt
  - either **MPTA** : the Maximum Torque Per Amper
- Maximum allowed rotating speed for the electrical machine

### Input parameters: Rotor initial angle

The Rotor initial angle is a necessary variable in order to apply Park's transforms. Two possibilities:

- User: the user knows the initial angle of the rotor and can enter the value
- Auto: the value of the initial angle of the rotor is computed automatically. The goal is to find the rotor angle for t=0s in order to align it with the magnetic flux density in the airgap generated by the reference phase (e.g., phase A). Two simulations are performed in order to compute rotor initial angle. A path in the middle of the airgap is automatically defined in order to compute the magnetic flux density within it.

### Input parameters: Number of computations requested

- Number of computation for Jd and Jq (8 by default)
- Number of computations for the speed (10 by default)
- Number of computations by electrical period (30 by default)

> **Note:** The total number of computations is the product of the 4, so with the default values there are 19200 computation points. It is necessary to have enough points to obtain a good Torque-Speed envelope. So the calculation time can therefore be quite long. The choice of values is a compromise between accuracy of results and solving time. The default values allow this compromise.
>
> As a reference, for 20000 computation points on a server with 22 cores with 8 Mio (GB) of numerical memory, a 2D project is solved in 5h. With this same project, with 75000 computation points (Id and Iq at 10, Speed at 15 and Nb of computations by electrical period at 50), results will be more accurate but with a solving time around 24h.

### Input parameters: Options of distribution and memory

- Activate or not the local distribution CDE (Computing Distribution Engine). The number of cores must be defined in the configuration of CDE

  > **Attention:** Make sure the service is installed and enabled. If this is not the case even with the option to **Yes**, the solving will be sequential and not distributed.

  > **Note:** The computing distribution is not implemented in Skew.

- Numerical memory allocated to the solving of Flux project
- Number of cores used by the Mumps solver for the initial project

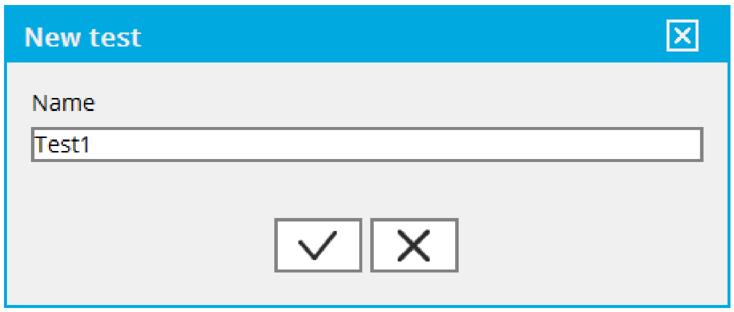
- If the local distribution via CDE is **Yes**, the value of the number of cores used by the Mumps solver is fixed at 1, and non accessible for the user, (in this case the parameter is grayed out, which means that the indicated value is not taken into account)
- If the local distribution via CDE is **No**, the user value of the number of cores is taken into account by the Mumps solver. By default this value is 4.
- In the case where the local distribution via CDE is **Yes**, but CDE is not correctly configured, the number of cores entered will be taken into account by the Mumps solver.

# 2.5.5 Flux e-Machine Toolbox: Tests

### Running a test

Once the input parameters have been defined, the user can run a test by clicking on

A dialog box appears in which the user must choose the name of the test and click on .




### List of tests

Once the test running is done, the test appears at the bottom right. Several tests can be run on a same component, in this case they are listed:

| TESTS | |
|---|---|
| ✓ test1 | 2019-06-11 01:56:15 |
| ✕ test2 | 2019-06-11 01:56:44 |
| test3 | 2019-06-11 01:57:00 |

### Actions on a test

The possible actions on a test run are:

- **Display:**

  Display the test results by double-clicking on it or by right click → **Display**

  A tab with the name of the test is added in the central area, and it is possible to navigate easily using the filters on the left

- **Close:**

  Close the display of a test by clicking on the cross of the tab or by right-click Overview Test1 or by right-click → **Close**

- **Delete:**

  Delete a test by selecting it in the list and by clicking on or by a right-click → **Delete**

- **Export:**

  Export a test by selecting it in the list and by clicking on or by a right-click → **Export**

### Test states

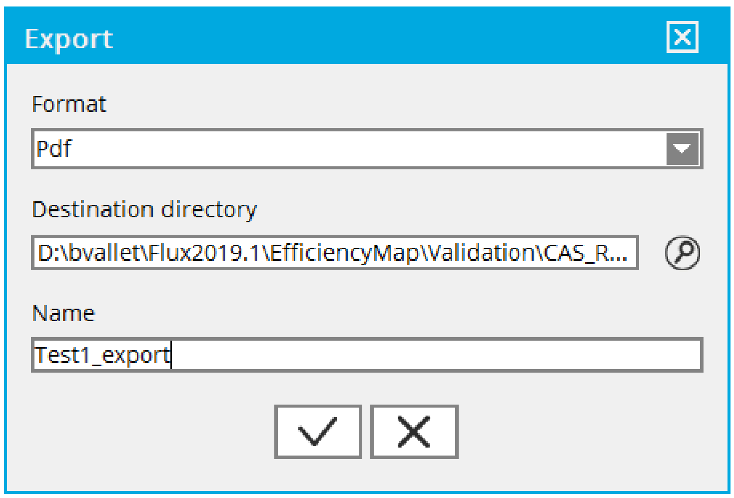
A test can have 3 states:

- ✔ : the test is successful
- : the test is in progress
- ✖ : the test is failed

### Export a test

A test can be exported in 3 different formats:

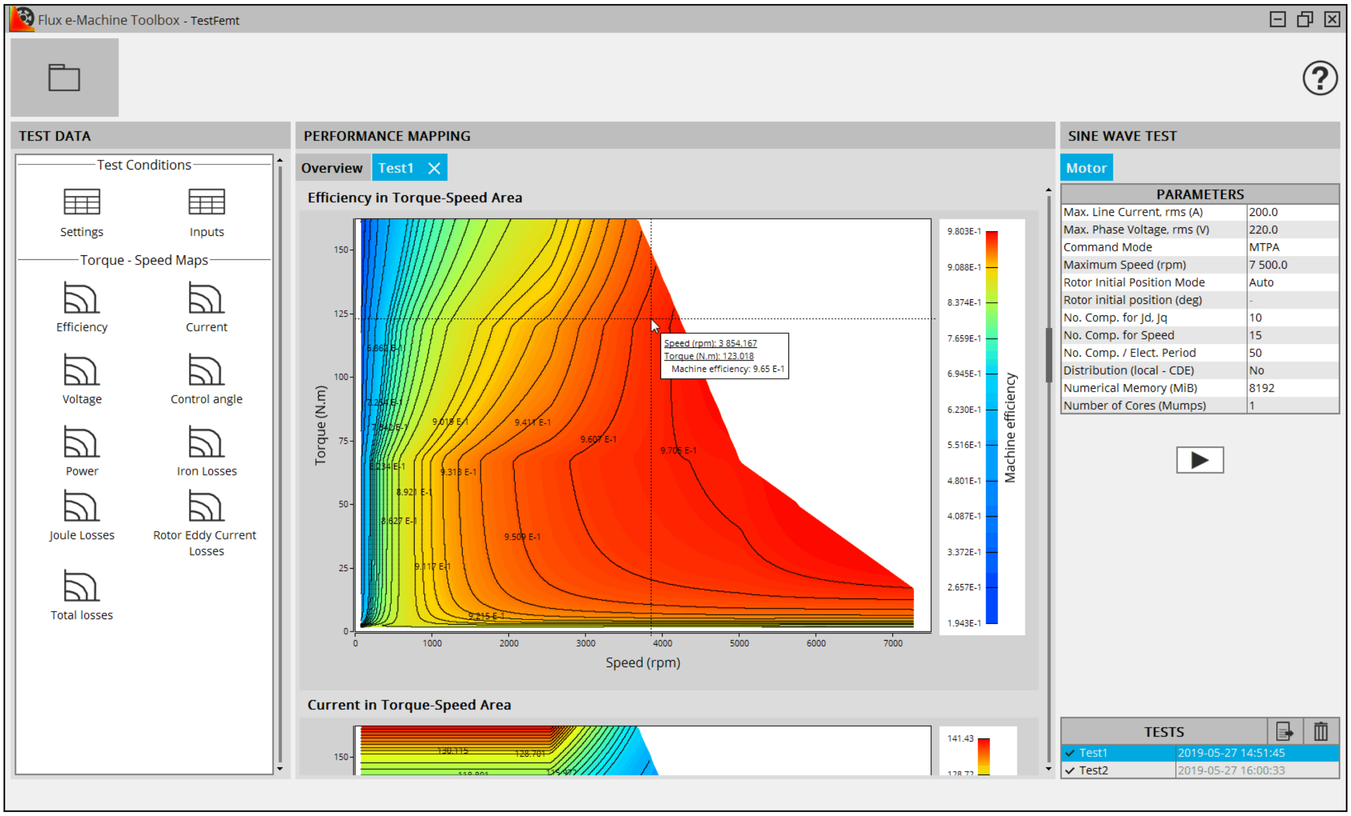
- **pdf** format for a report (equivalent to all the content of the central area)
- **html** format for a report (equivalent to all the content of the central area)
- **txt** format to postprocess data (a text file by quantity)



# 2.5.6 Flux e-Machine Toolbox: Postprocessing

## Introduction

Once the test running is completed, performance maps are displayed in the central area. The filters on the left allow to display directly the perfomance map wished. The user has the possibility to use a cursor to display values by pointing on the map.




## Performance maps

Several quantities are postprocessed in the Torque-Speed area:

- Efficiency
- Phase current
- Phase voltage
- Control angle
- Mechanical power
- Iron losses in rotor and stator
- Stator winding Joule losses
- Rotor Eddy current losses
- Total losses: the sum of these 3 types of losses

### Skew specificity

In Flux, the description of a Skew model is done in the 2D environment. The 3D device is then rebuilt for the postprocessing. The face regions become volume regions.

The iron losses are only calculated on the non-conductive magnetic regions, because the laminated regions do not exist in Skew. In this case, the default Bertotti coefficients are applied after the end of the Flux solving process.

(K1=151.88 ; K2=0.07 ; K3=1.19 ; A1=2 ; A2=2 ; A3=1.5)

# 2.5.7 Flux e-Machine Toolbox: Workflows

## Generate a coupling component for Flux e-Machine Toolbox

Here are the steps to run a computation via Flux e-Machine Toolbox:

1. Open the project in Flux starting from the Flux Supervisor

   → *The description of the electrical machine is considered done in this project*
2. Check the prerequisites in the Flux project for Flux e-Machine Toolbox coupling
3. Open the dialog box of the generation of Flux e-Machine Toolbox coupling component

   Click on the **Solving** menu and click on **Generate component for Flux e-Machine Toolbox coupling**
4. Choose the name of the component
5. Choose to overwrite **Yes** or **No** the component if already existing (**No** by default)
6. Choose the save directory (the working directory by default)
7. Choose to run **Yes** or **No** the Flux e-Machine Toolbox application and open the generated component (**Yes** by default)
8. Choose the electrical machine to study (family, type and sub-type of the motor)
9. Choose the prerequisites for the computation of stator winding Joule losses
   a. The number of phases
   b. The coil conductor for each phase
10. Choose the prerequisites for the computation of rotor Eddy current losses
    a. Choose the support type (Sensor or Region or Solid conductor)
    b. Choose the support
11. Choose the geometric parameter defining the number of poles of the machine
12. validate by clicking on **OK**

    → *The coupling component XXX.FEMT has been created*

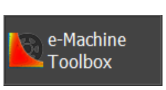
    → *Flux e-Machine Toolbox is launched and the coupling component is opened*

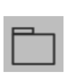
## Run a test in Flux e-Machine Toolbox

Here are the steps to run a test in Flux e-Machine Toolbox:

> **Note:** The coupling component is considered already generated in Flux.

1. Open Flux e-Machine Toolbox by clicking in the supervisor on:

   
   
2. Open the coupling component by clicking on the icon:

    and choose the coupling component XXX.FEMT and validate

→ *The component is opened in Flux e-Machine Toolbox*

3. Define different user parameters
4. Run the test

   Click on 

   → *A progress bar indicates that the test is in progress*


   → *The Flux solving is run in batch mode*

   → *Once the Flux solving is done, a postprocessing script is run*

   → *Once the postprocessing script is completed, perfomance maps are loaded in the central area of Flux e-Machine Toolbox*
5. Postprocess the maps

   → *It is possible to export the results of a test for a report (HTML and PDF format) or to postprocess data (TXT format)*

### Variant: Run a test from an electrical machine model obtained by FluxMotor

It is possible to do a Flux e-Machine Toolbox coupling from a motor described in FluxMotor and thus benefit from the pre-design done with FluxMotor and the modeling of the losses in Flux to obtain the performance maps in Flux e-Machine Toolbox:

1. In FluxMotor, generate a python script corresponding to the studied motor
2. In Flux:
   a. Run this python script to create the Flux project
   b. Save the project
   c. Generate the Flux e-Machine Toolbox coupling component
3. In Flux e-Machine Toolbox, run the test

# 2.5.8 Flux e-Machine Toolbox: Command Line

## Introduction

The command line of Flux e-Machine Toolbox is a program called `emtcli` which allows to invoke certain features of Flux e-Machine Toolbox from a script.

This command line can be launched from the **emtcli.exe** file located in the directory:

%FLUX_INSTALL_DIR%/Flux/EMachineToolbox/bin

Currently `emtcli` allows to run an performance map computation from an existing Flux e-Machine Toolbox component and / or to export the results of a previous perfomance test.

## The different syntaxes

- Running an performance map computation:

```
emtcli --project <FeMT component path> --inputs <input file path>
```

- Export results of an existng computation in txt format:

```
emtcli --project <FeMT component path> --export <test name> <format> <output path>
```

- Running a computation and export it at the end:

```
emtcli --project <FeMT component path> --inputs <input file path> --export <test
 name> <format> <output path>
```

## Options

- `--project`: the path of the Flux e-Machine Toolbox coupling component from which the computation or the export must be run
- `--inputs`: the path to the text file which contains the input parameters of the computation to run. This file must contain one parameter by line, each parameter being defined by: `Name=Value`.

The supported parameters correspond to the input parameters displayed in the Flux e-Machine Toolbox application and are described below:

```
TestName=<Name of the test to create to do a computation>
```

```
LineCurrent=<corresponds to the RMS phase current maximal (A)>
```

```
LineVoltage=<corresponds to the RMS phase voltage maximal (V)>
```

```
CommandMode=MTPA or MTPV <corresponds to the command mode>
```

```
MaxSpeed=<corresponds to the maximal speed (tr/min)>
```

```
InitialAngleComputation=USER or AUTO <corresponds tothe definition of the initial
 angle>
```

```
InitialAngle=<corresponds to the rotor intial angle (deg, to define only if
 InitialAngleComputation is fied at AUTO>
```

```
NbIdIq=<corresponds to the number of computation for Jd, Jq>
```

```
NbSpeed=<corresponds to the number of computation for the speed>
```

```
NbCompElectPeriod=<corresponds to the computation by electrical period>
```

```
DistributionEnabled=yes or no <corresponds to the distribution (local - CDE)
```

```
CoreNumber=1 <corresponds to the number of cores (Mumps)>
```

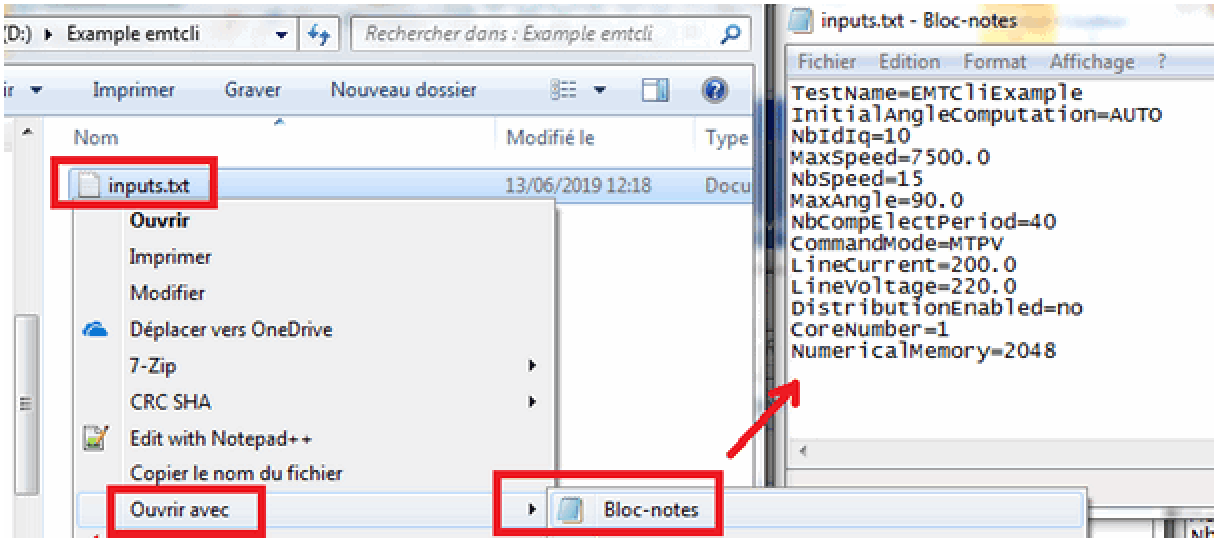
```
NumericalMemory=2048 <corresponds tot the numerical memory (Mio) >
```

- `--export`: must be followed by exactly three arguments:
    - `<test name>`: The name of the test present in the coupling component, for which the results must be exported. The name will also be used to name the export directory in which the text files will be exported
    - `<format>`: The format of exported files. Currently only the string "TXT" is supported and allows to export the results in text format
    - `<output path>`: The path of a directory in which the export directory will be created. If this path does not point to an existing directory the command line will try to create the missing directories

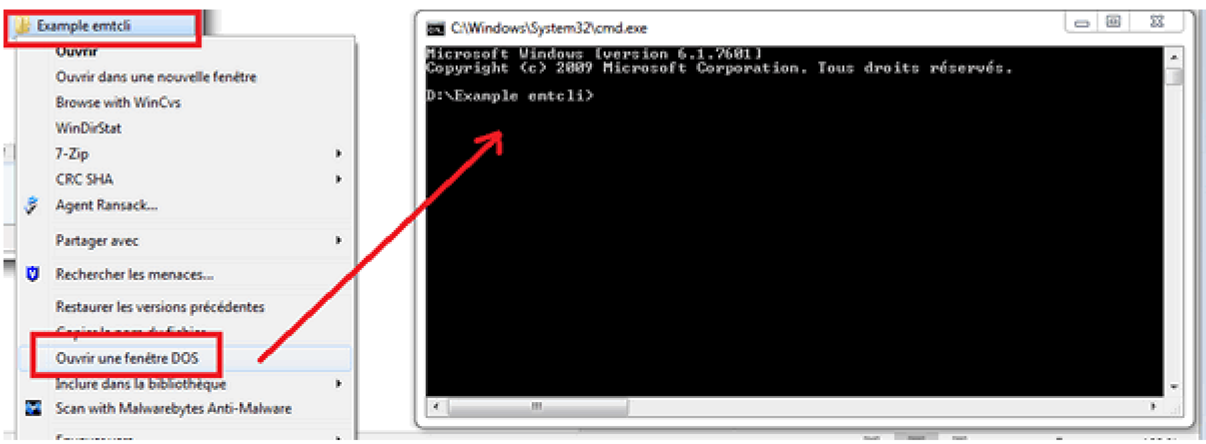
## Example of use via the DOS console

It is necessary to have generated a FeMT coupling component from a Flux project and have the path to this component. For instance d:\Example emtcli\Example.FEMT. Here are the steps to use the command line via DOS:

1. In a user working directory, create a new text file named « inputs.txt »




2. Open the « inputs.txt » file in a text editor and enter wished values for input parameters of test to run:




3. From a Windows Explorer, open a DOS window in the working directory
4. In the DOS window, update the environment variable « Path » by adding the directory « bin » of EMachineToolkit to be able to run easily **emtcli**:

```
> setPATH=%PATH%;C:\Altair\Flux_2019.1\EMachineToolbox\bin
```

5. Then execute the command « emtcli » to run the test and export results:

```
> emtcli --project Example.FEMT --inputs inputs.txt –export EMTCliExample TXT
"d:\Example emtcli"
```