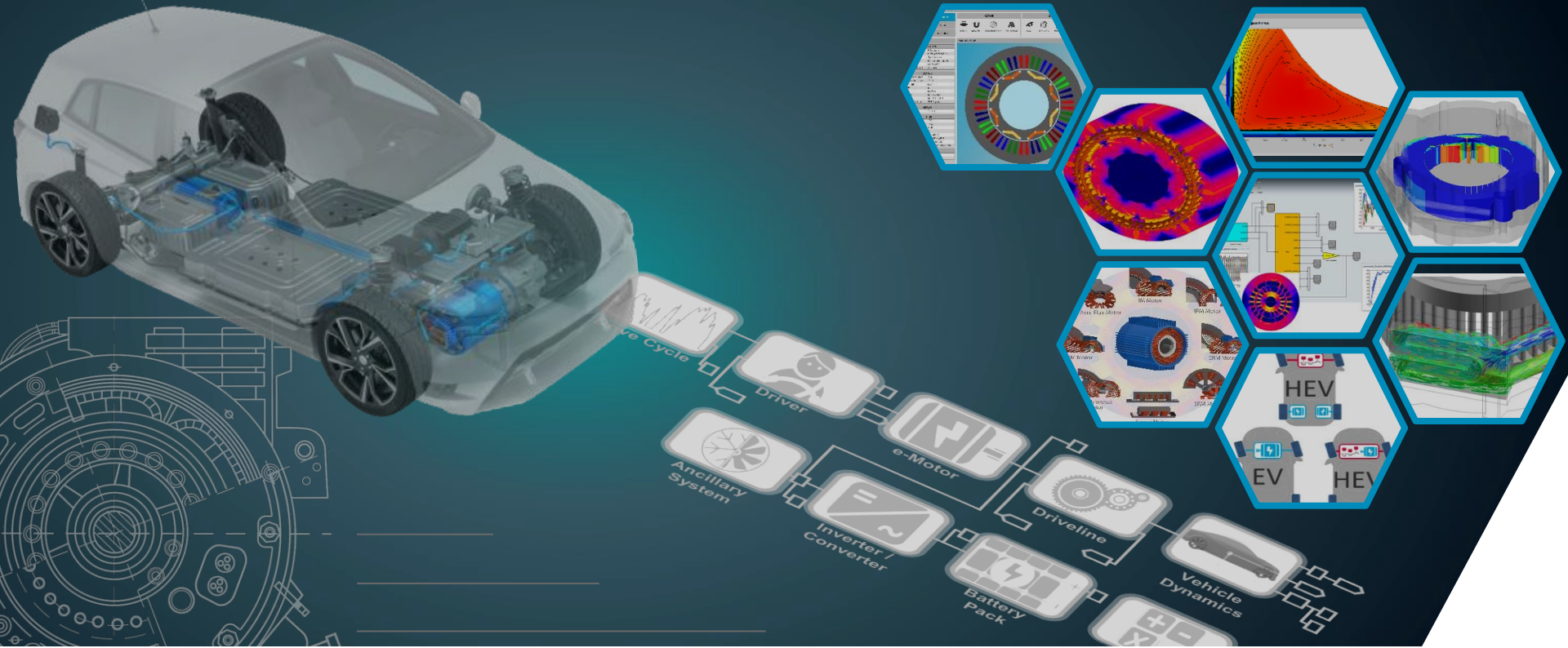


FLUX: MAGNETIC ANALYSIS FOR ELECTRIC MOTORS

ALTAIR MULTIDISCIPLINARY DESIGN OPTIMIZATION PLATFORM FOR ELECTRIC MOTORS

October 2021, Altair Flux / FluxMotor Valorization and Support Team

OBJECTIVES

Creating Flux magnetic analysis projects

- Base speed point simulation
- Specific speed point simulation

Motor performance analysis at base speed point

- Analyzing the motor performance (iron losses, magnet looses, efficiency, etc.)
- Creating HyperStudy connector for the MDO application

Motor performance analysis at specific speed point

- Analyzing the motor performance (iron losses, magnet looses, efficiency, etc.)
- Creating HyperStudy connector for the MDO application

OUTLINE

Creation of magnetic simulation scenarios

Magnetic analysis: base speed point

Magnetic analysis: specific operating point

Input file

- **Python script for model generation in Flux**
  EMOTOR_Flux_Model_Generation.py

Software

- Altair Flux 2021.2.1 (or later versions)

Output files

- **Flux 2D projects**
  - EMOTOR_MAG_BASE_POINT.FLU
  - EMOTOR_MAG_SPECIFIC_POINT.FLU
- **Flux / HyperStudy connectors**
  - Connector_Fx_Hst_BasePoint.F2HST
  - Connector_Fx_Hst_OperatingPoint.F2HST

Automation script file

- EMOTOR_Magnetic_Analysis_Project.py

ALTAIR

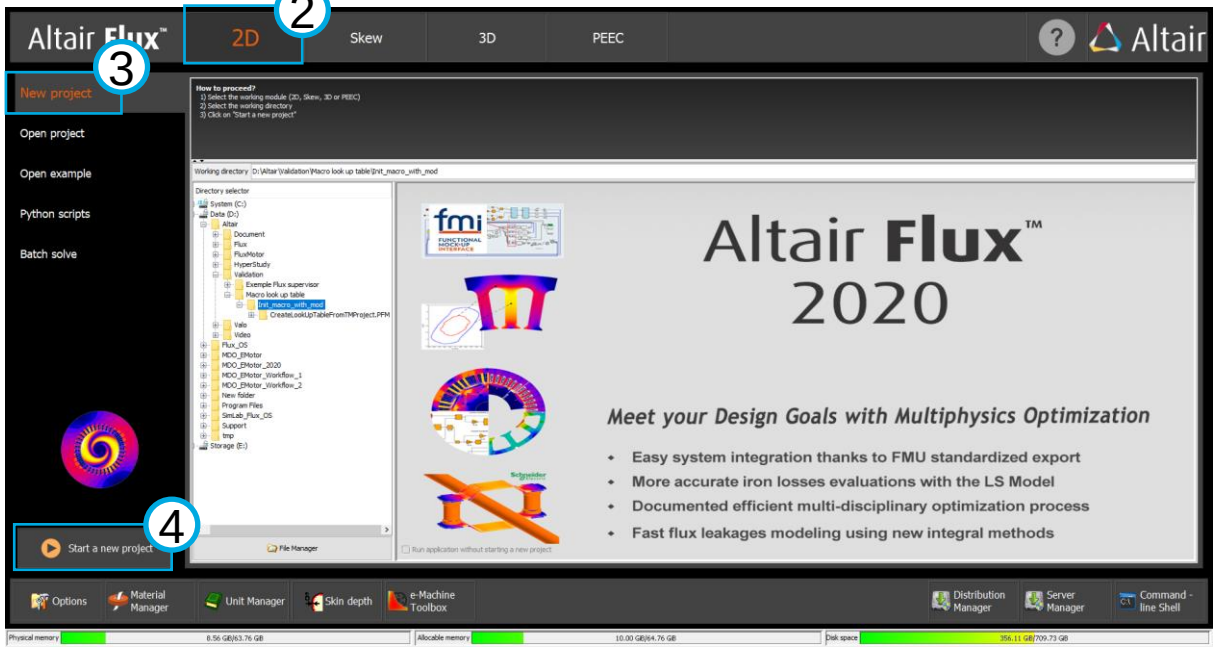
I. CREATION OF MAGNETIC SIMULATION SCENARIOS

CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project initiation
  - Start a new Flux 2D project

Step	Action
1	Open Flux supervisor
2	Select the [2D] simulation context
3	Click on [New project]
4	Click on [Start a new project]

Working path	~\MDO_EMotor\Flux\FluxProjects\MAG_BASE_POINT

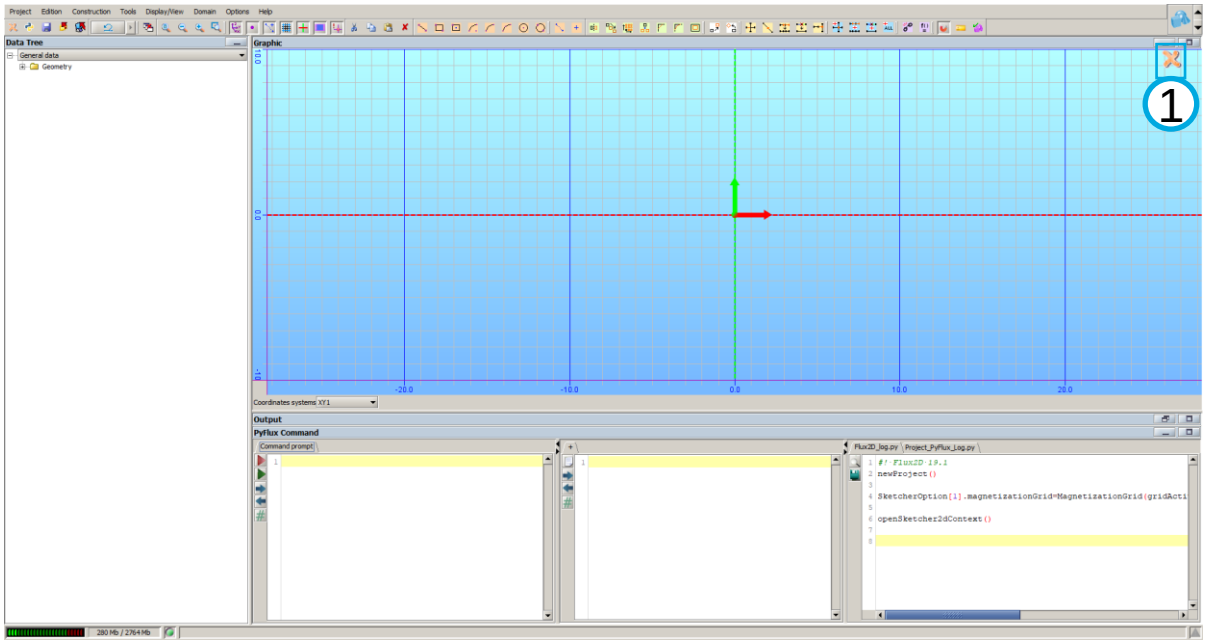


ALTAIR

CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project initiation
  - Close Flux Sketcher 2D Context

Step	Action
1	Click on the icon to close Flux Skecher 2D Context

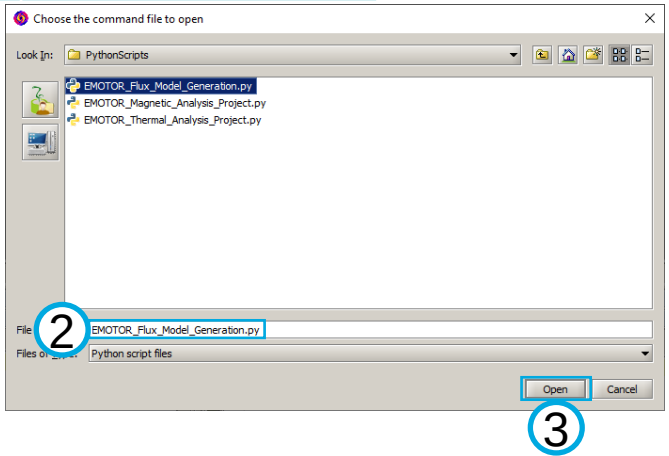


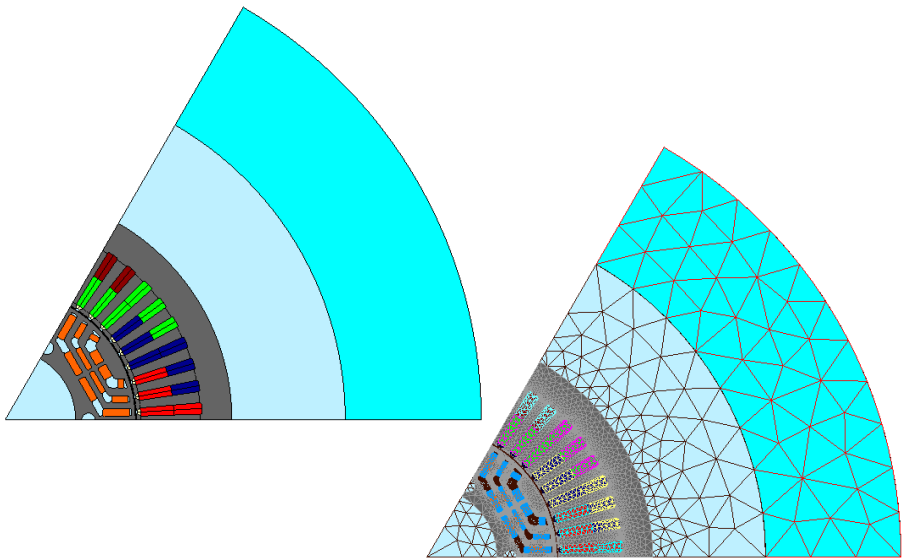
ALTAIR

CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project initiation
  - Import the motor model (by executing the Python script generated by FluxMotor)

Step	Action
1	Click on the icon
2	Select the file “EMOTOR_Flux_Model _Generation.py”
3	Click on [Open]



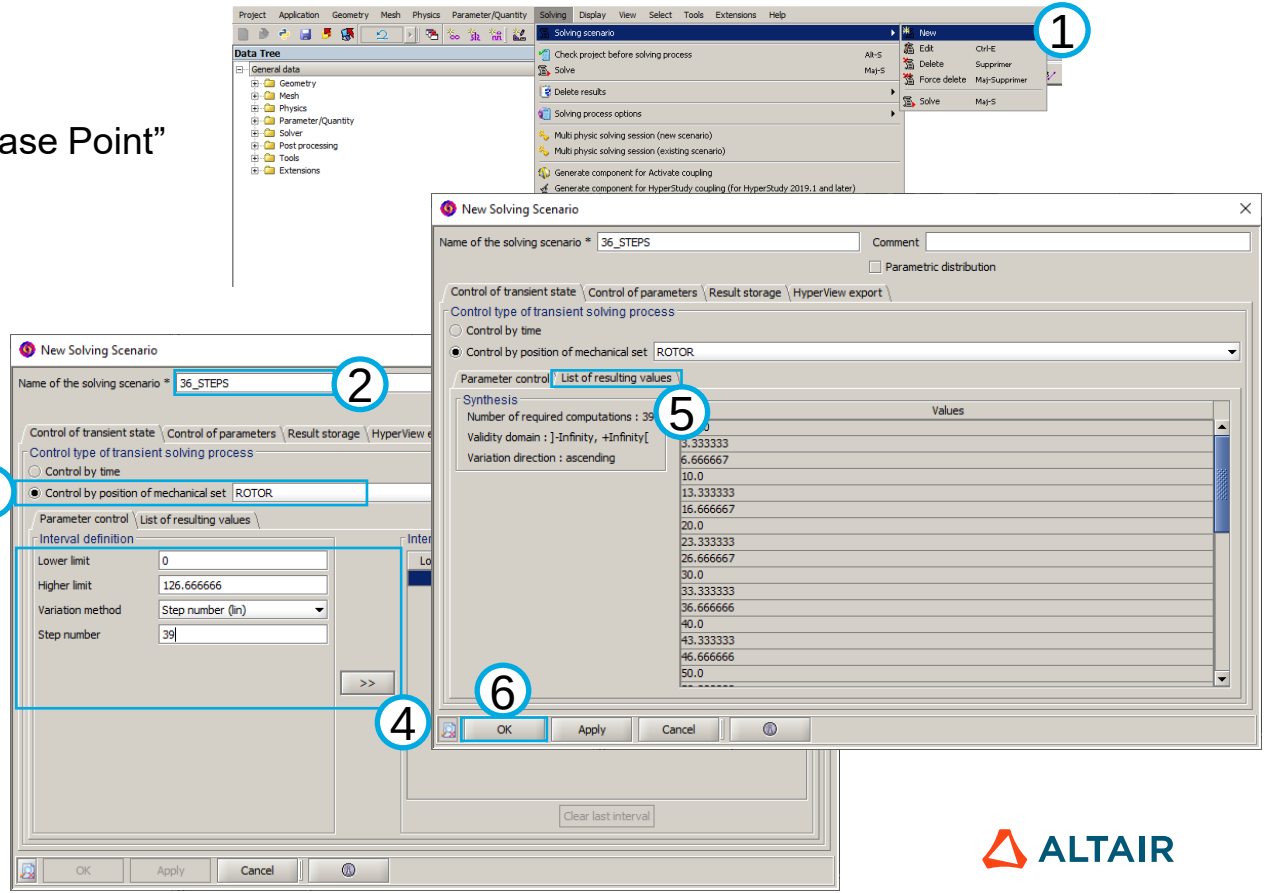


ALTAIR

CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project initiation
  - Create solving scenario “Base Point”

Step	Action
1	Create a Solving scenario by clicking on [Solving] – [Solving scenario] – [New]
2	Define the scenario name as “36_STEPS”
3	Select “Control by position of mechanical set – Rotor”
4	In parameter control tab, define the simulation step as “Step number (lin)” - Lower limit: 0.0 - Higher limit: 126.666666 - Step number: 39 Click on [>>]
5	Verify the step values in the [List of resulting values] tap
6	Click on [OK]

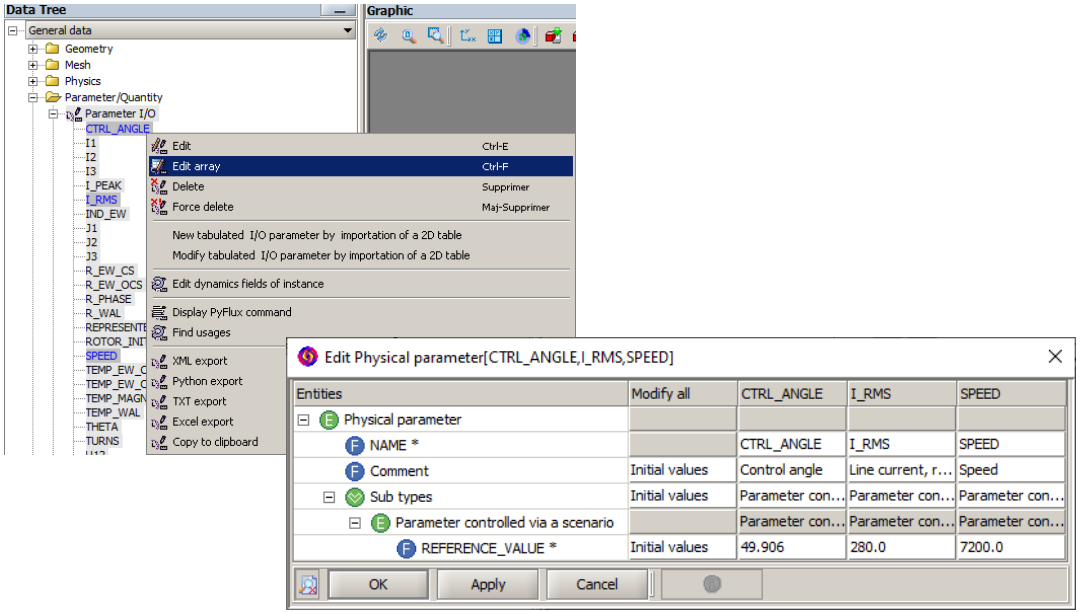


CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project specification for “Base Speed Point” application
  - Update I/O parameter values

Step	Action
1	Select the parameters “CTRL_ANGLE”, “I_RMS” and “SPEED” from the Data Tree [Parameter/Quantity] – [Parameter I/O]
2	Right click, and click on [Edit array] to modify the values as shown in the following table, click on [OK]

Parameter	Value	Unit
CTRL_ANGLE	49.906	degree
I_RMS	280.0	A
SPEED	7200.0	RPM

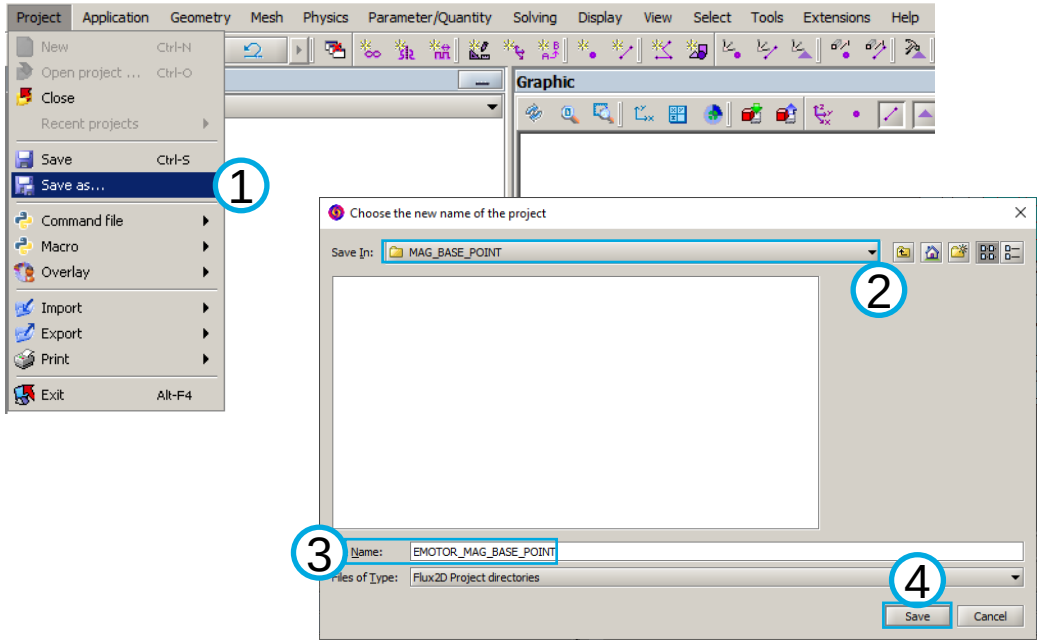


ALTAIR

CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project specification for “Base Speed Point” application
  - Save the Flux 2D project

Step	Action
1	Click on [Project] – [Save as]
2	Select the saving folder
3	Define the project name
4	Click on [Save]



Project name	EMOTOR_MAG_BASE_POINT.FLU
Saving folder	~/MDO_EMOTOR/Flux/FluxProjects/ MAG_BASE_POINT

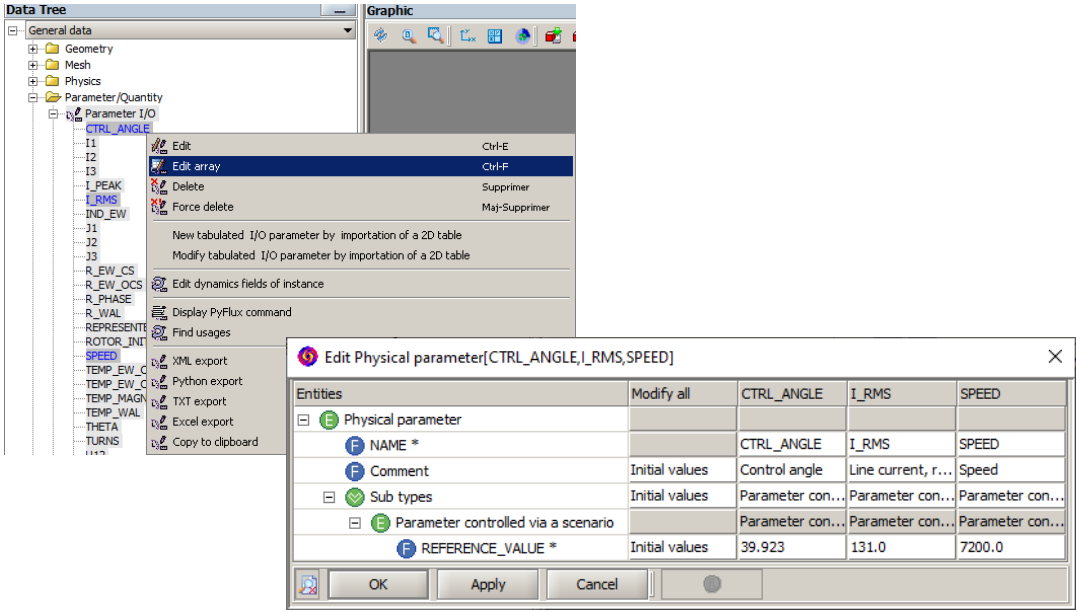
ALTAIR

CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project specification for “Specific Operating Point” application
  - Update I/O parameter values

Step	Action
1	Select the parameters “CTRL_ANGLE”, “I_RMS” and “SPEED” from the Data Tree [Parameter/Quantity] – [Parameter I/O]
2	Right click, and click on [Edit array] to modify the values as shown in the following table, click on [OK]

Parameter	Value	Unit
CTRL_ANGLE	39.923	degree
I_RMS	131.0	A
SPEED	7200.0	RPM

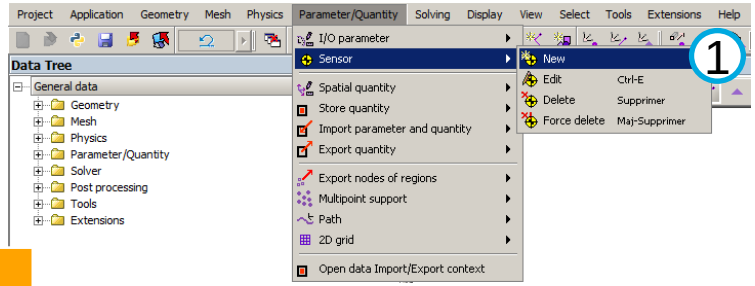


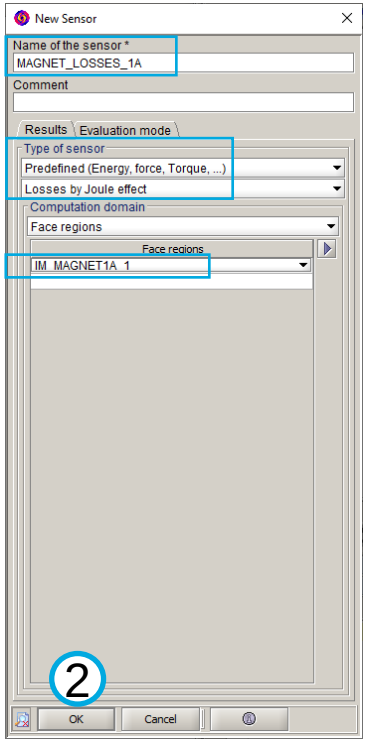
ALTAIR

CREATION OF MAGNETIC SIMULATION SCENARIOS

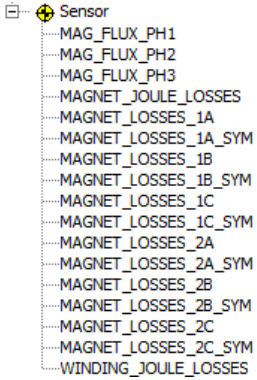
- Flux 2D project specification for “Specific Operating Point” application
  - Create sensors for each magnet to measure magnet losses

Step	Action
1	Click on [Parameter / Quantity] – [Sensor] – [New]
2	Create the 12 sensors in the following table





Sensor names	Corresponding face region names
MAGNET_LOSSES_1A	IM_MAGNET1A_1
MAGNET_LOSSES_1A_SYM	IM_MAGNET1A_SYM_1
MAGNET_LOSSES_1B	IM_MAGNET1B_1
MAGNET_LOSSES_1B_SYM	IM_MAGNET1B_SYM_1
MAGNET_LOSSES_1C	IM_MAGNET1C_1
MAGNET_LOSSES_1C_SYM	IM_MAGNET1C_SYM_1
MAGNET_LOSSES_2A	IM_MAGNET2A_1
MAGNET_LOSSES_2A_SYM	IM_MAGNET2A_SYM_1
MAGNET_LOSSES_2B	IM_MAGNET2B_1
MAGNET_LOSSES_2B_SYM	IM_MAGNET2B_SYM_1
MAGNET_LOSSES_2C	IM_MAGNET2C_1
MAGNET_LOSSES_2C_SYM	IM_MAGNET2C_SYM_1

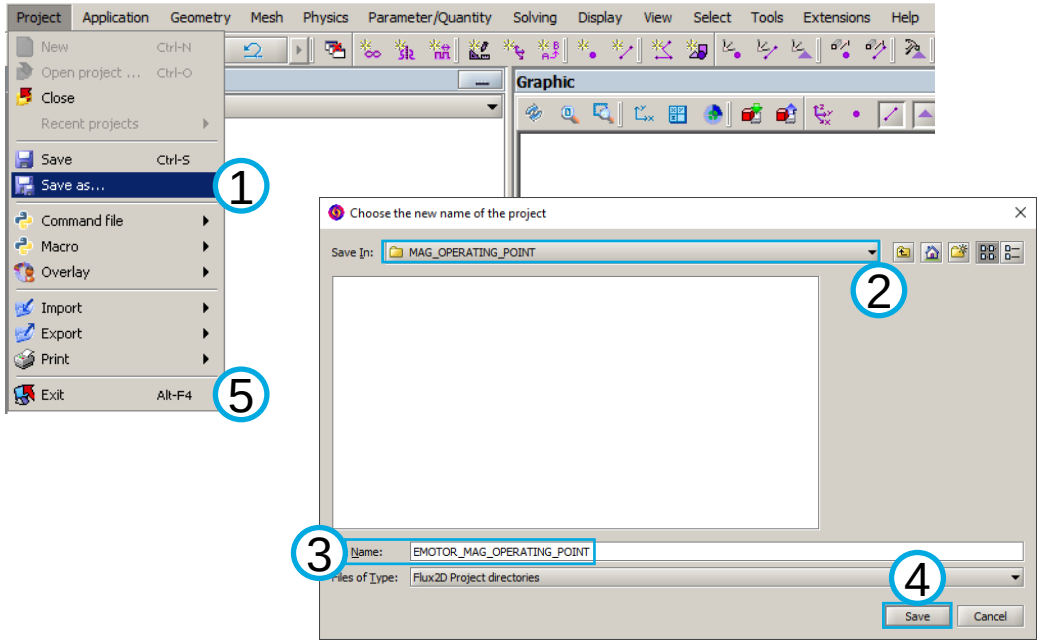


ALTAIR

CREATION OF MAGNETIC SIMULATION SCENARIOS

- Flux 2D project specification for “Specific Operating Point” application
  - Save the Flux 2D project

Step	Action
1	Click on [Project] – [Save as]
2	Define the project name
3	Select the saving folder
4	Click on [Save]
5	Click on [Project] – [Exit]



Project name	EMOTOR_MAG_OPERATING_POINT.FLU
Saving folder	~/MDO_EMOTOR/Flux/FluxProjects/ MAG_OPERATING_POINT

ALTAIR

MAGNETIC ANALYSIS: BASE SPEED POINT

ALTAIR

OUTLINE

Solving and postprocessing: base speed point

Generating HyperStudy connector: base speed point

Input file

- **Flux 2D project**
  - EMOTOR_MAG_BASE_POINT.FLU

Software

- Altair Flux 2019.1 (or later versions)

Output files

- **Flux / HyperStudy connector**
  Connector_Fx_Hst_BasePoint.F2HST
- **Flux project associated with the connector**
  Connector_Fx_Hst_BasePoint.F2HST.FLU
- **Python script for the postprocessing in HyperStudy**
  Connector_Fx_Hst_BasePoint.py

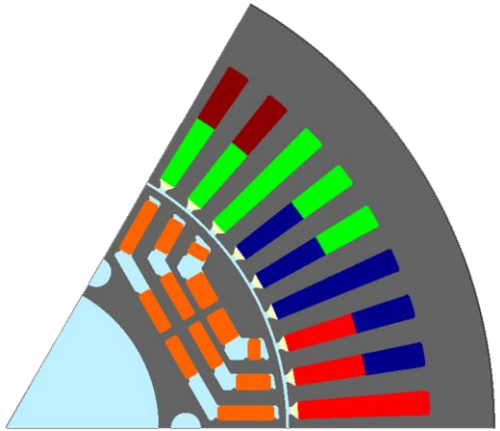
SOLVING AND POSTPROCESSING: BASE SPEED POINT

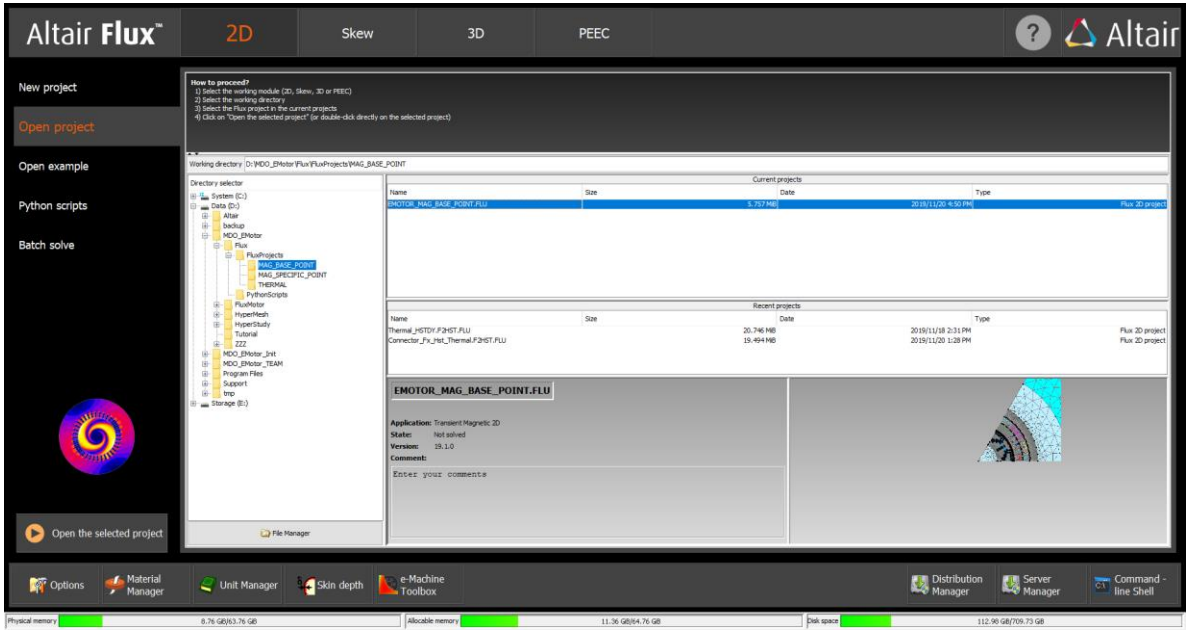
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Solving the magnetic problem at base speed point

- Open Flux 2D project

Step	Action
1	Open the project “EMOTOR_MAG_BASE_POINT.FLU”





Project name	EMOTOR_MAG_BASE_POINT.FLU
Project folder	~/MDO_EMOTOR/Flux/FluxProjects/MAG_BASE_POINT

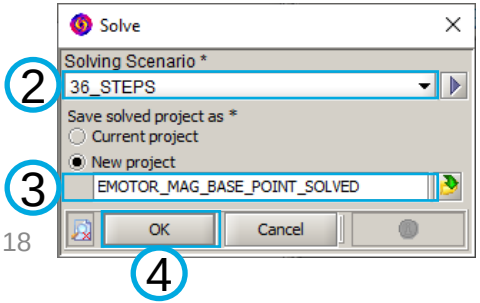
ALTAIR

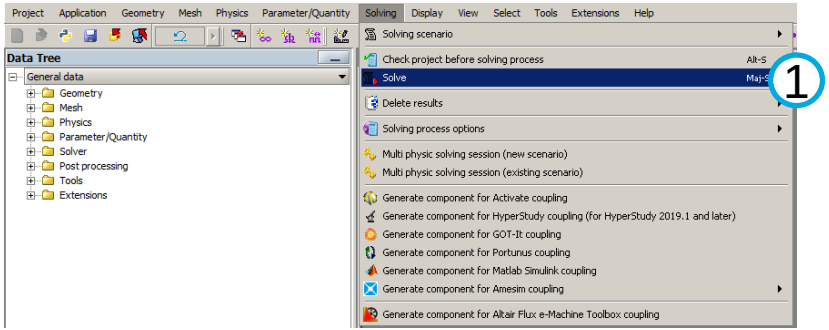
SOLVING AND POSTPROCESSING: BASE SPEED POINT

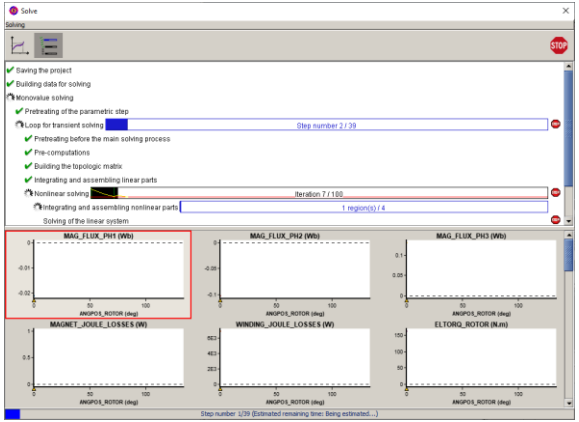
Solving the magnetic problem at base speed point

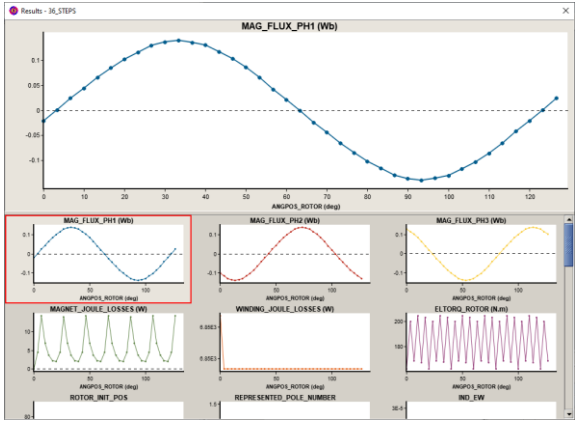
- Solve the project

Step	Action
1	Click on [Solving] – [Solve]
2	Select the solving scenario “36_STEPS”
3	Save the solved project as a new project “EMOTOR_MAG_BASE_POINT_SOLVED.FLU”
4	Click on [OK]









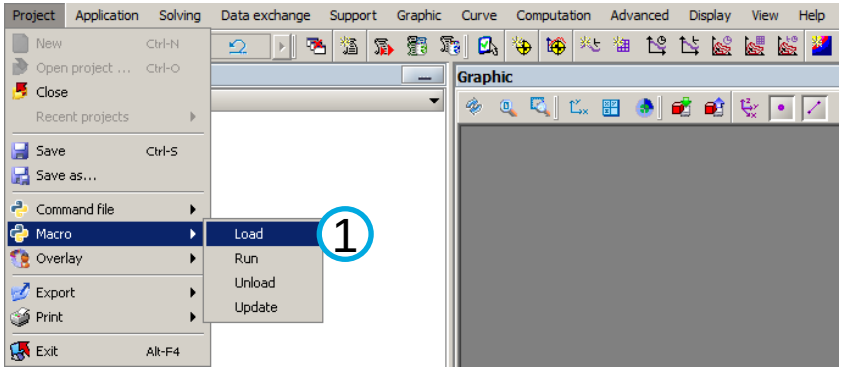
ALTAIR

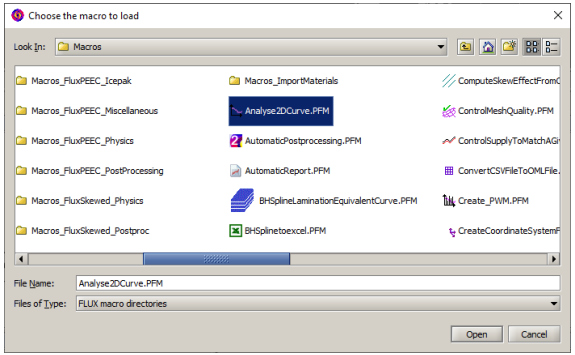
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: postprocessing initialization

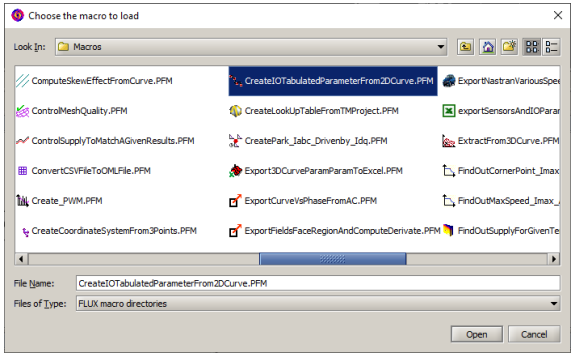
- Load Flux macros

Step	Action
1	Click on [Project] – [Macro] – [Load]
2	Load the following two macros: - Analyse2DCurve - CreateIOTabulated ParameterFrom2DCurve





This macro extracts different values on a 2D curve, and create corresponding I/O parameters



This macro intends to create a new tabulated I/O parameter from a 2D curve.

ALTAIR

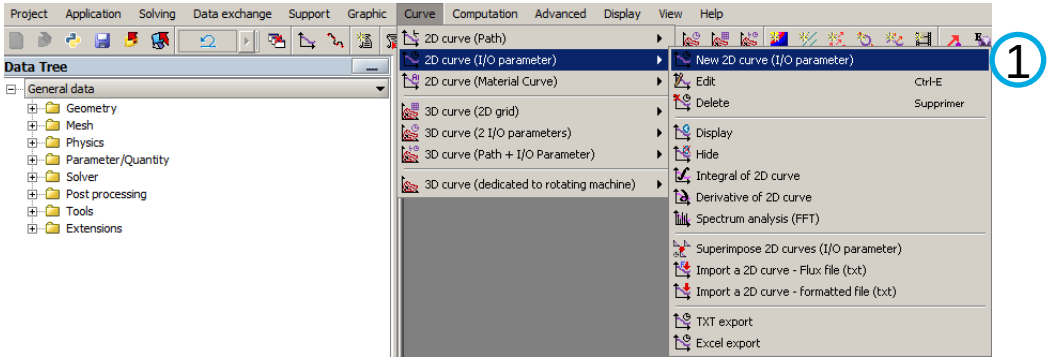
SOLVING AND POSTPROCESSING: BASE SPEED POINT

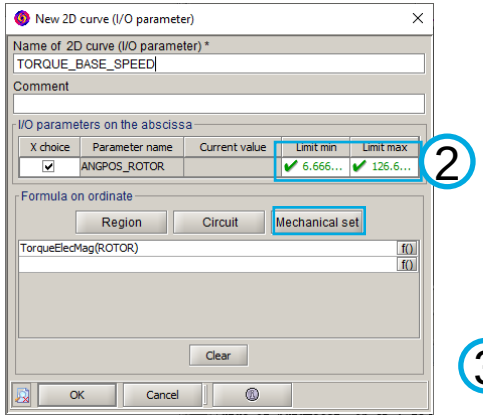
Magnetic analysis: magnetic torque at base speed point

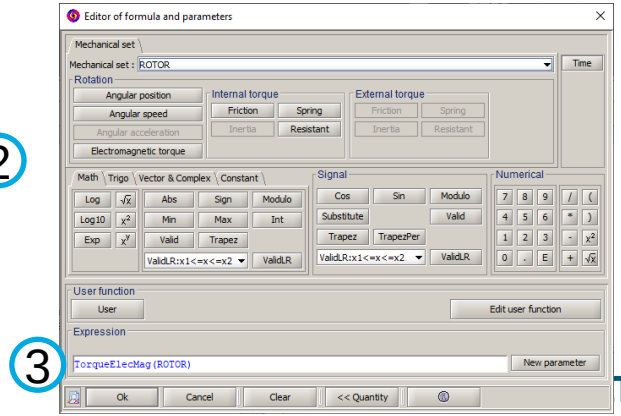
- Plot the magnetic torque curve

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [New 2D curve (I/O parameter)]
2	Create a new 2D curve TORQUE_BASE_SPEED Limit min: 6.666666 Limit max: 126.666666
3	Click on [Mechanical set], select “Rotor”, and click on [Electromagnetic torque], click on [OK]
4	Click on [OK]

Curve name	TORQUE_BASE_SPEED
Formula	TorqueElecMag(ROTOR)



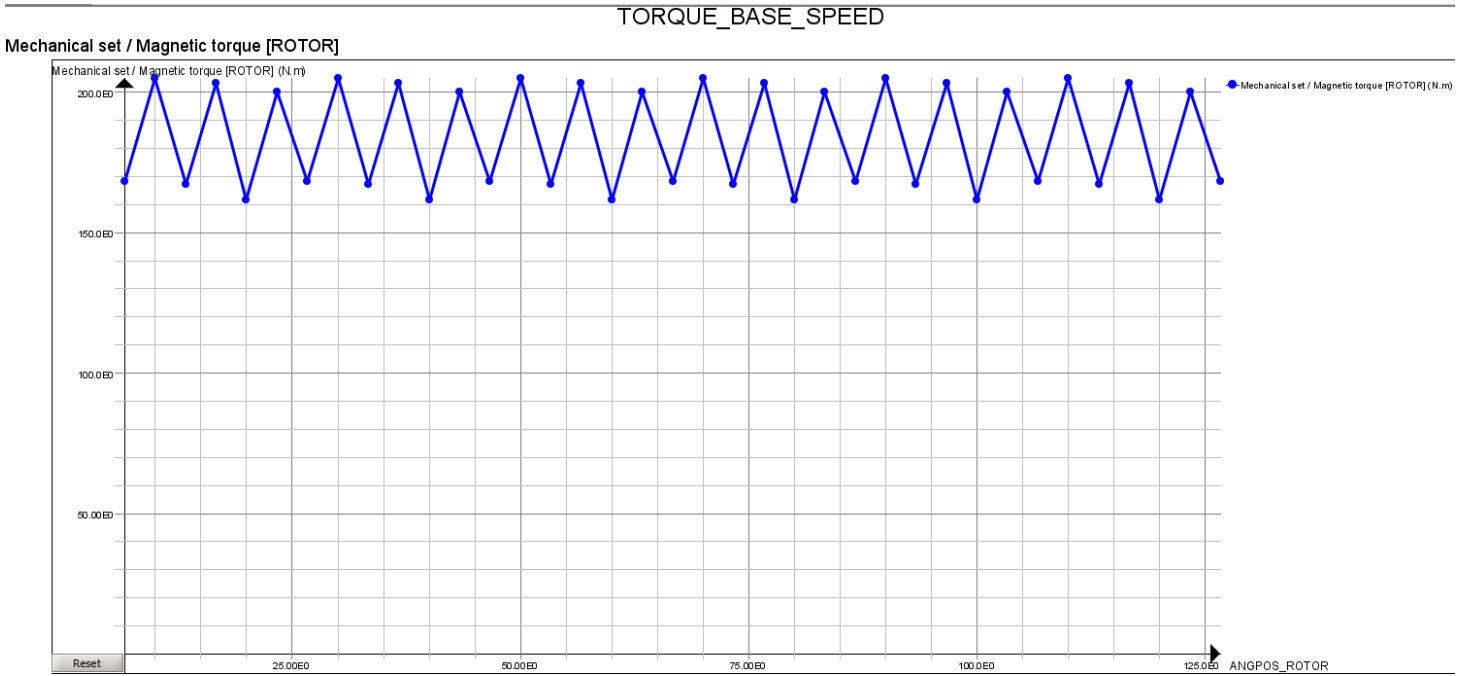




SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: magnetic torque at base speed point

- Plot the magnetic torque curve



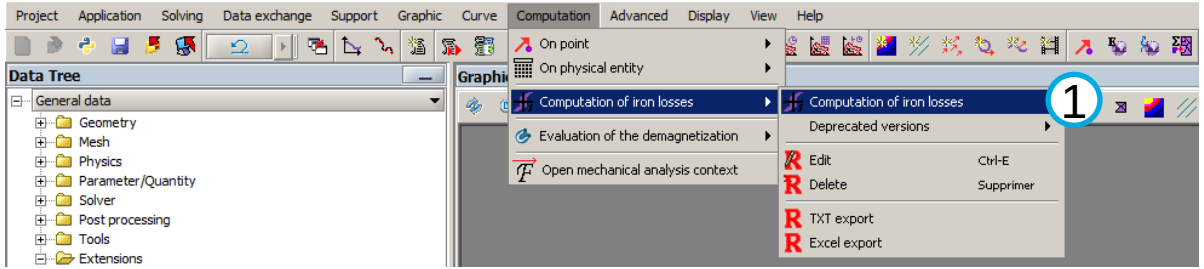
ALTAIR

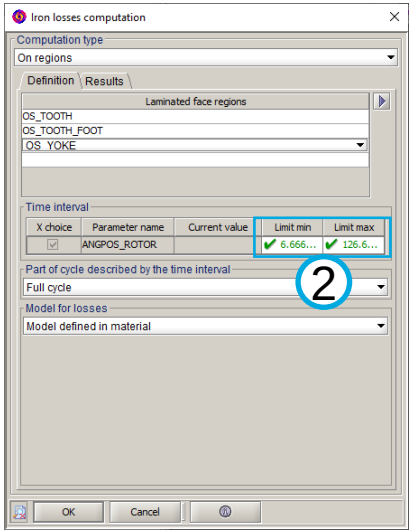
SOLVING AND POSTPROCESSING: BASE SPEED POINT

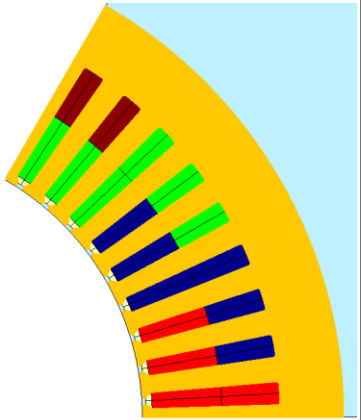
Magnetic analysis: stator iron losses at base speed point

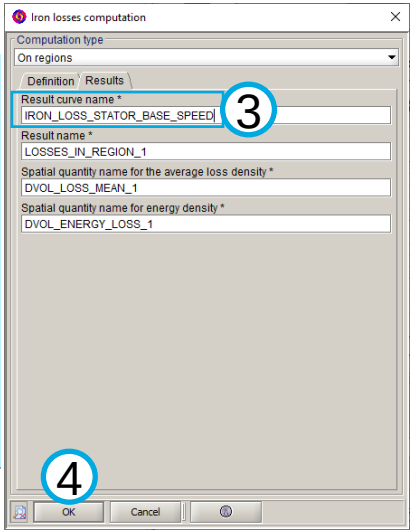
- Plot the stator iron loss curve

Step	Action
1	Click on [Computation] – [Computation of iron losses] – [Computation of iron losses]
2	Define the computation configuration in tab [Definition]: Face region: - OS_TOOTH - OS_TOOTH_FOOT - OS_YOKE Interval: [6.6666, 126.6666]
3	Define the computation configuration in tab [Results]: Result curve name: IRON_LOSS_STATOR_BASE_SPEED
4	Click on [OK]





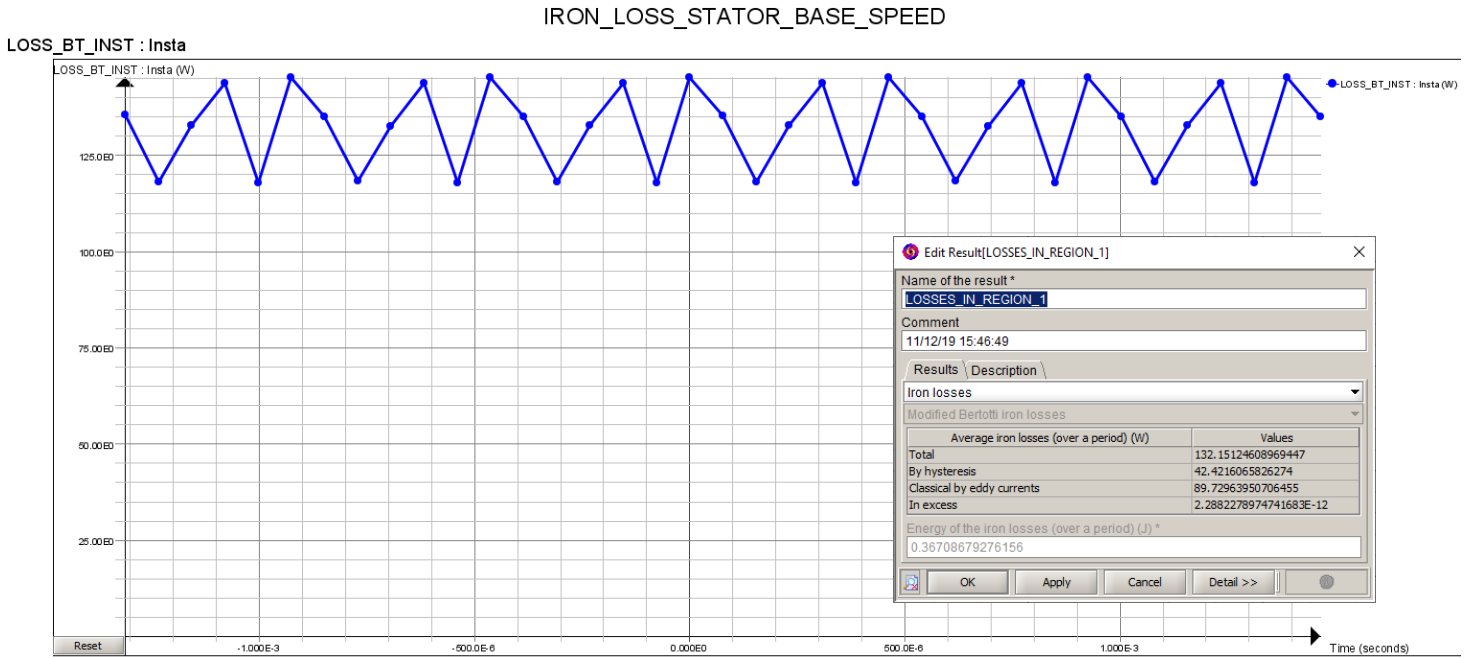




SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: stator iron losses at base speed point

- Plot the stator iron loss curve

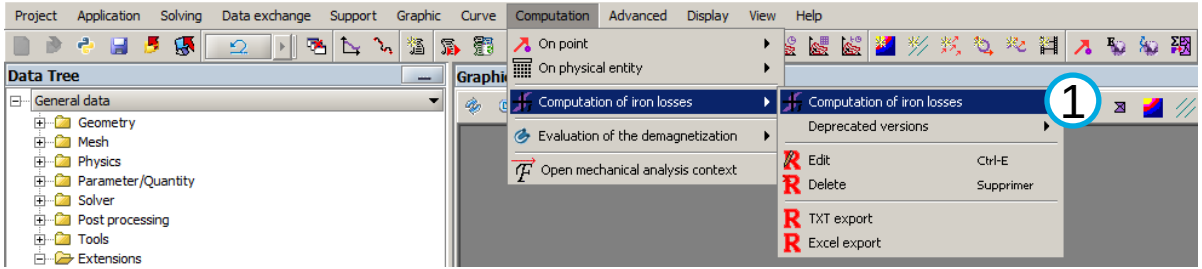


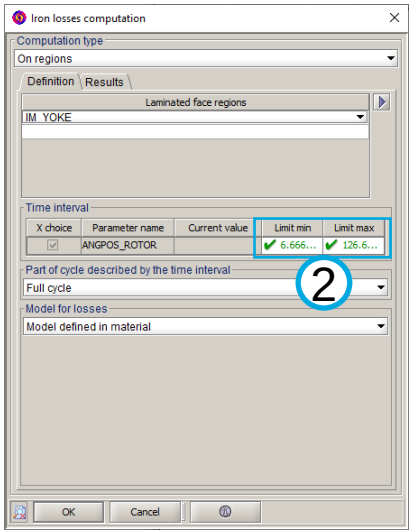
SOLVING AND POSTPROCESSING: BASE SPEED POINT

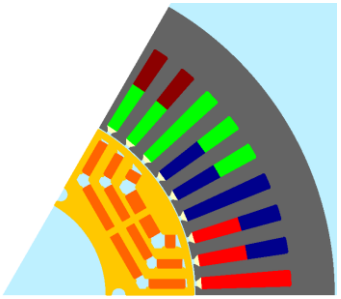
Magnetic analysis: rotor iron losses at base speed point

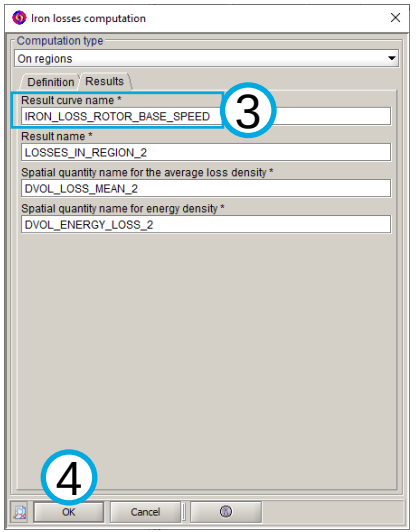
- Plot the rotor iron loss curve

Step	Action
1	Click on [Computation] – [Computation of iron losses] – [Computation of iron losses]
2	Define the computation configuration in tab [Definition]: Face region: - IM_YOKE Interval: [6.6666, 126.6666]
3	Define the computation configuration in tab [Results]: Result curve name: IRON_LOSS_ROTOR_BASE_SPEED
4	Click on [OK]





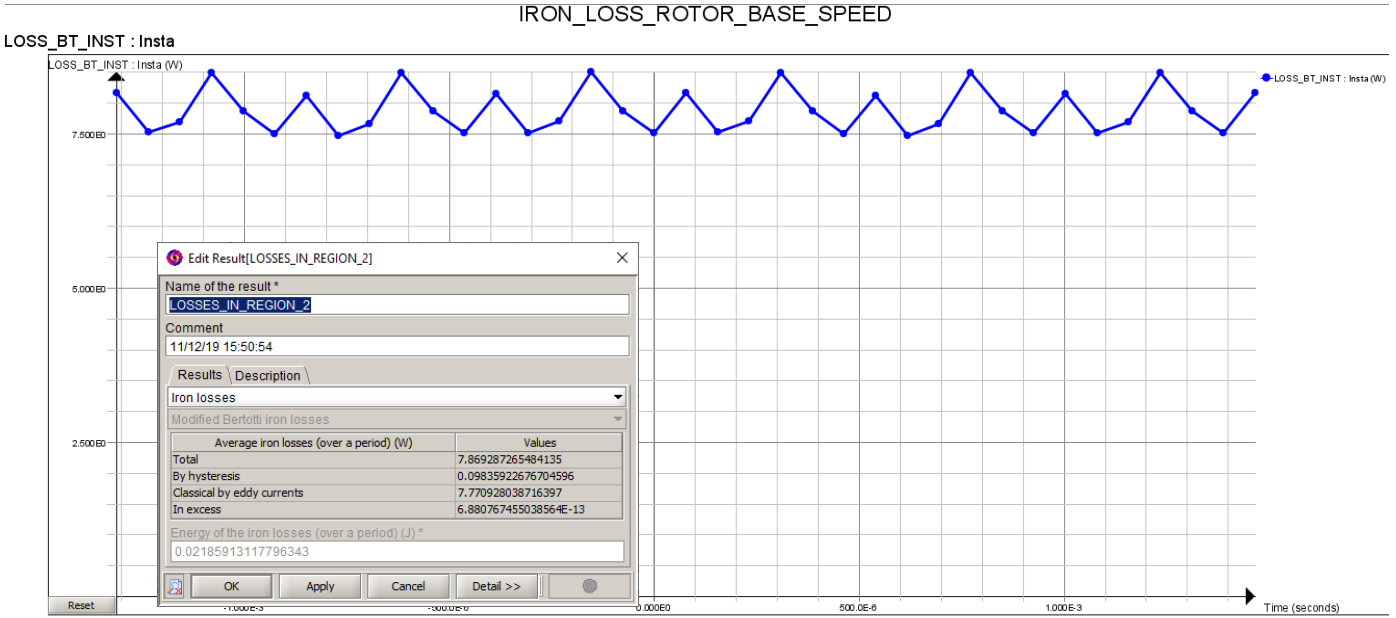




SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: rotor iron losses at base speed point

- Plot the rotor iron loss curve



ALTAIR

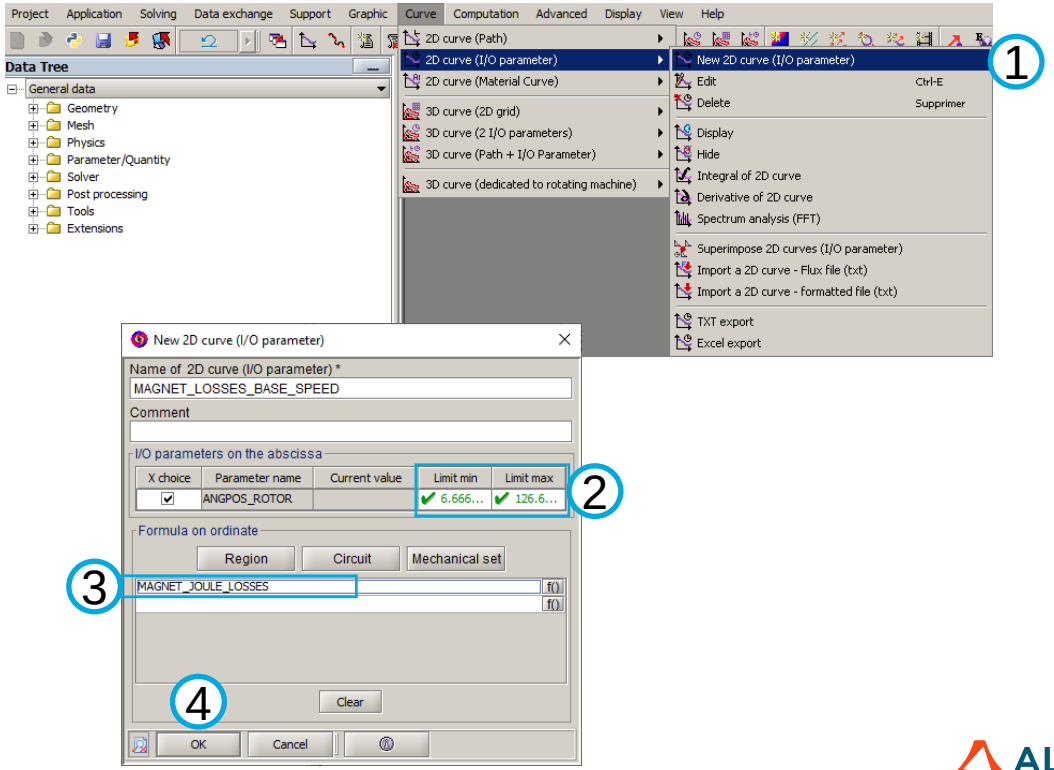
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: magnet losses at base speed point

- Plot the magnet loss curve

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [New 2D curve (I/O parameter)]
2	Create a new 2D curve MAGNET_LOSSES_BASE_SPEED Limit min: 6.666666 Limit max: 126.666666
3	Define the formula as the predefined sensor MAGNET_JOULE_LOSSES
4	Click on [OK]

Curve name	MAGNET_LOSSES_BASE_SPEED
Formula	MAGNET_JOULE_LOSSES

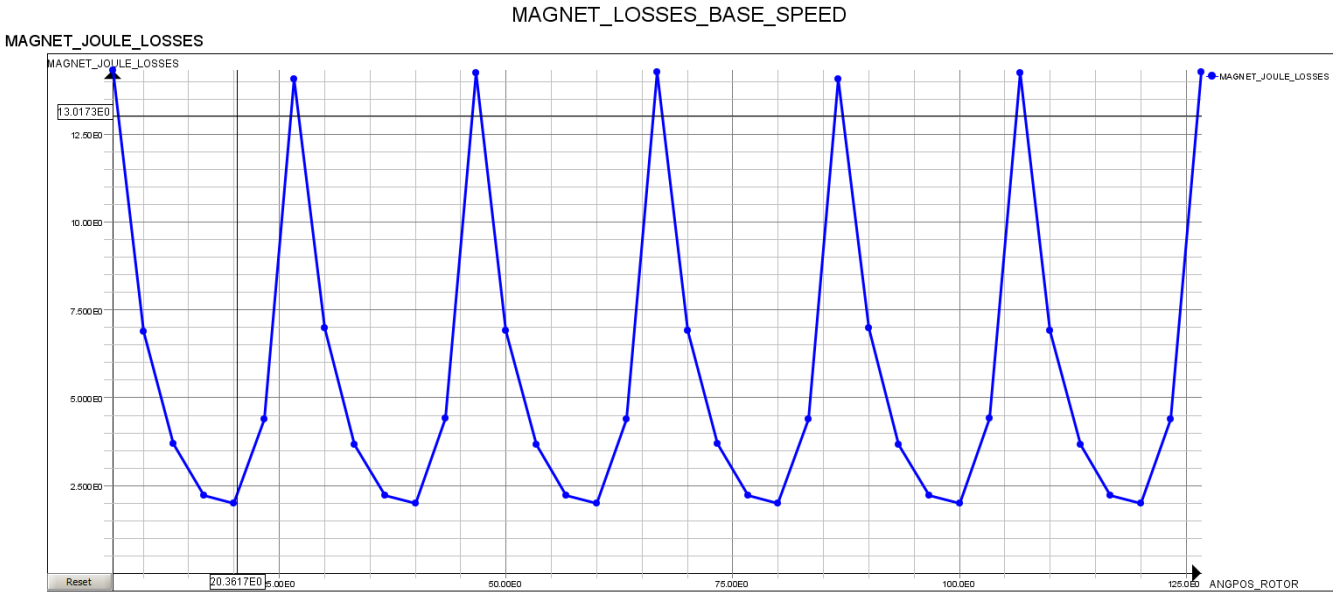


ALTAIR

SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: magnet losses at base speed point

- Plot the magnet loss curve



ALTAIR

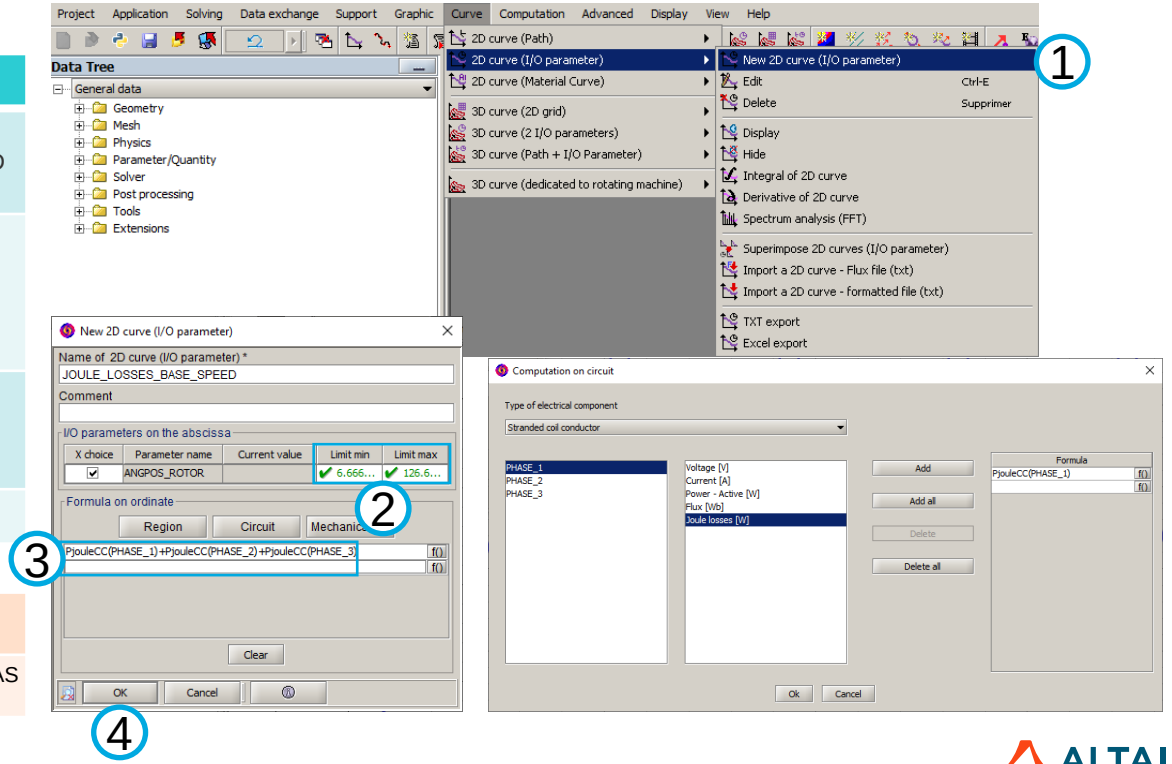
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: Joule losses at base speed point

- Plot the Joule loss curve

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [New 2D curve (I/O parameter)]
2	Create a new 2D curve JOULE_LOSSES_BASE_SPEED Limit min: 6.666666 Limit max: 126.666666
3	Define the plotting formula as PjouleCC(PHASE_1)+PjouleCC(PHASE_2)+PjouleCC(PHASE_3)
4	Click on [OK]

Curve name	JOULE_LOSSES_BASE_SPEED
Formula	PjouleCC(PHASE_1)+PjouleCC(PHASE_2)+PjouleCC(PHASE_3)

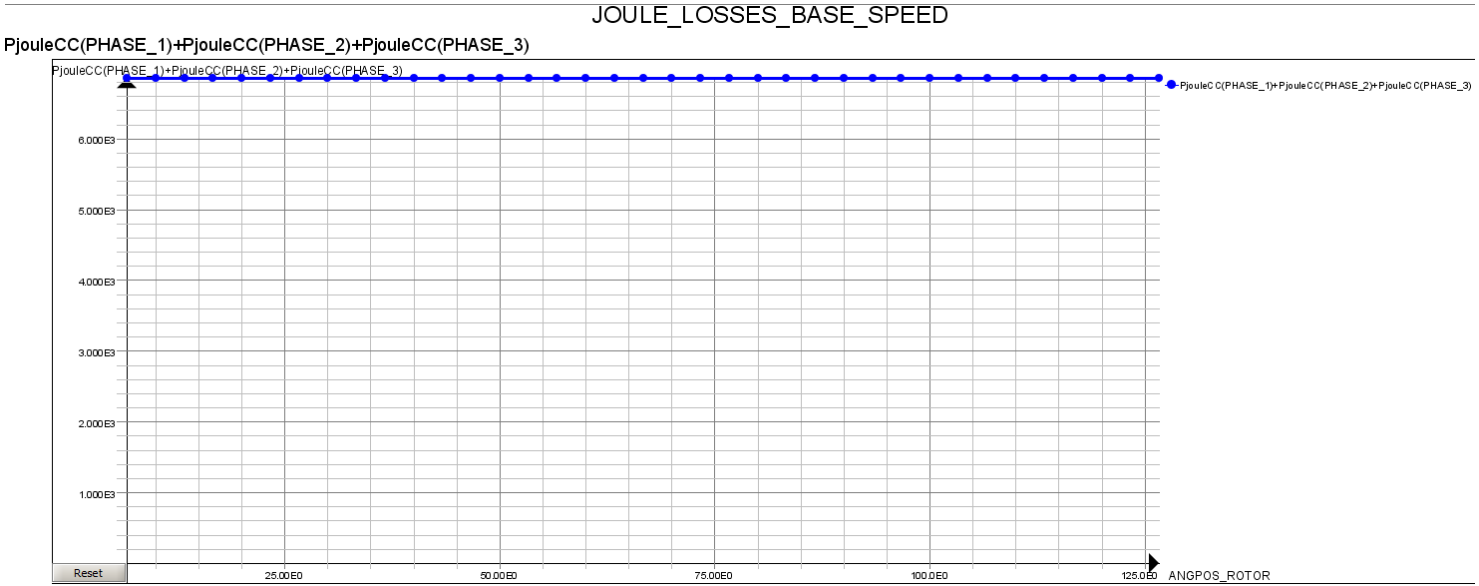


ALTAIR

SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: Joule losses at base speed point

- Plot the Joule loss curve



ALTAIR

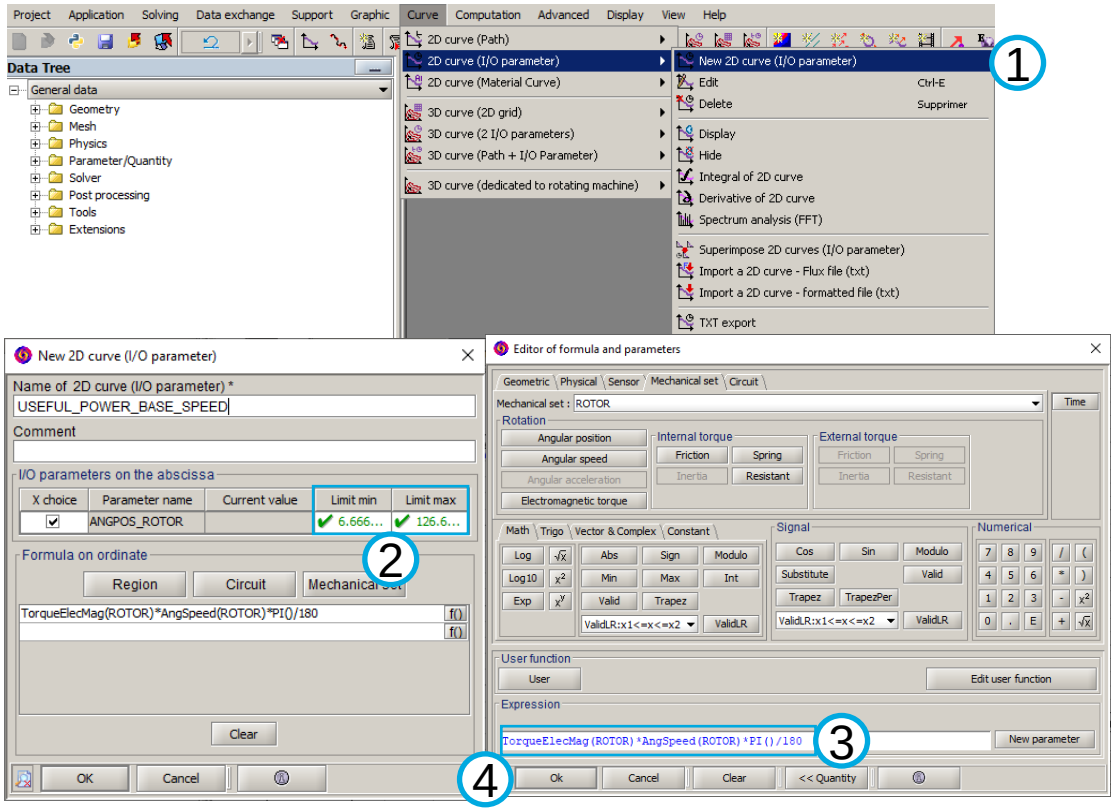
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: useful power at base speed point

- Plot the useful power curve

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [New 2D curve (I/O parameter)]
2	Create a new 2D curve USEFUL_POWER_BASE_SPEED Limit min: 6.666666 Limit max: 126.666666
3	Define the plotting formula as TorqueElecMag(ROTOR)*Ang Speed(ROTOR)*PI()/180
4	Click on [OK]

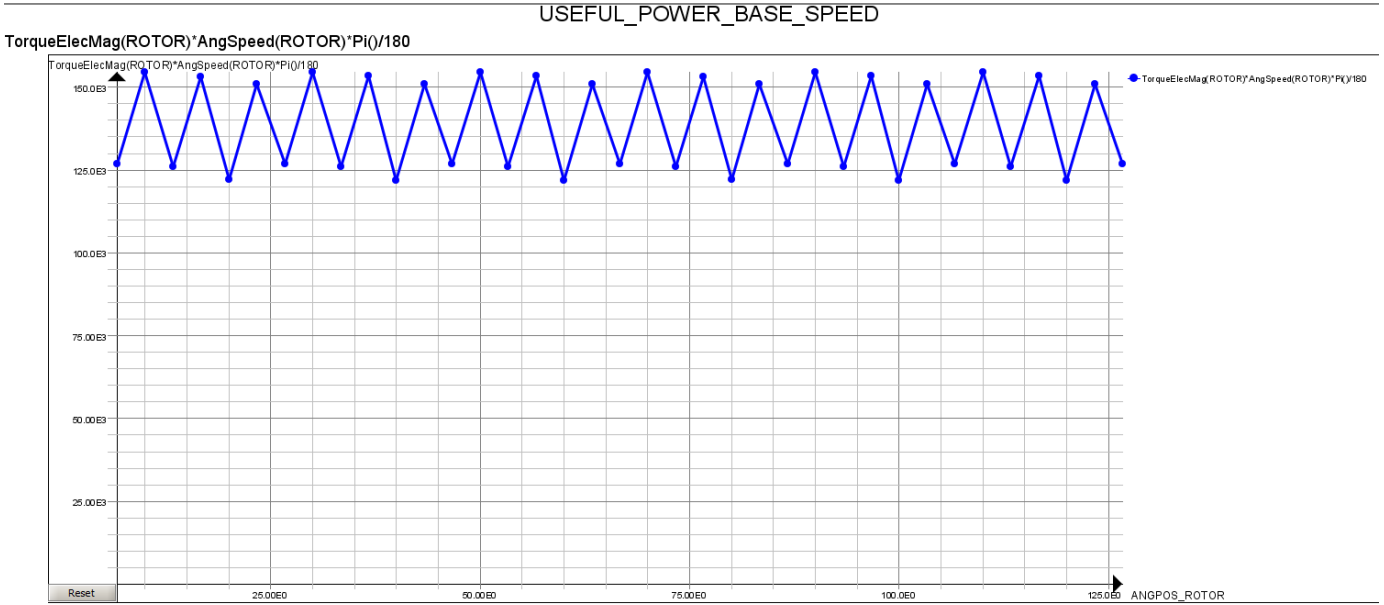
Curve name	USEFUL_POWER_BASE_SPEED
Formula	TorqueElecMag(ROTOR)*AngSpeed(ROTOR)*PI()/180



SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: useful power at base speed point

- Plot the useful power curve

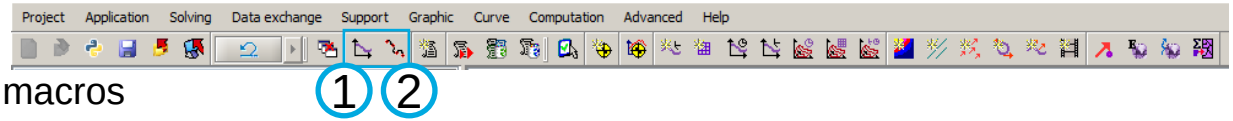


ALTAIR

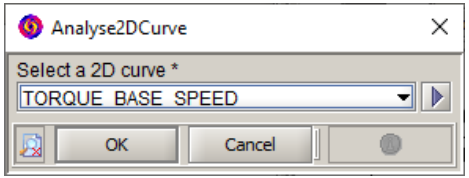
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis

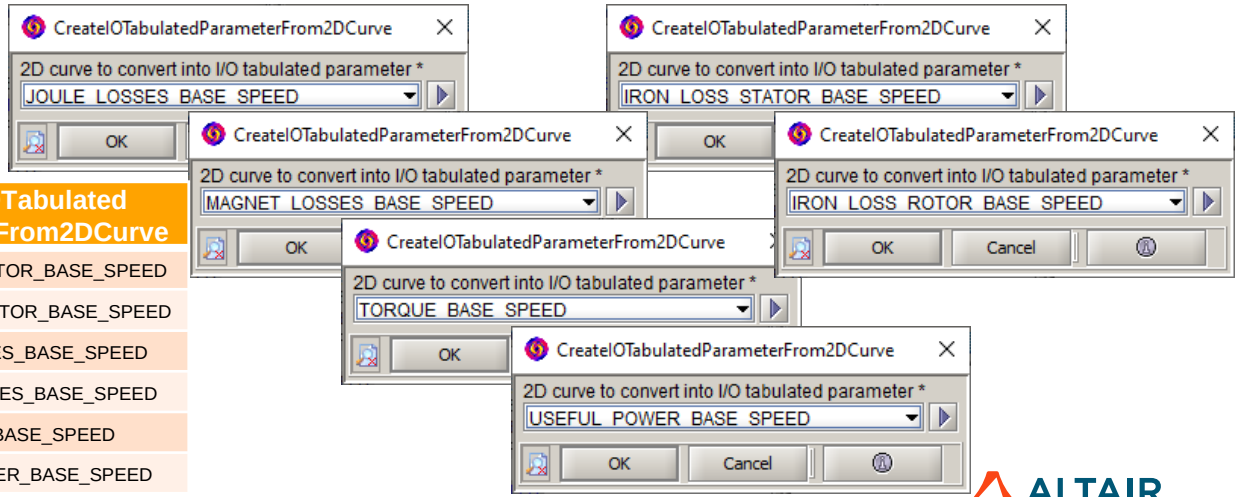
- Analyze the curves with macros



Step	Action
1	Click on the icon to run the macro "Analyse2DCurve"
2	Click on the icon to run the macro "Analyse2DCurve"



Tips: the curve analysis by predefined macros in Flux will generate parameters to calculate motor efficiency



Analyse2DCurve	CreateIOTabulated ParameterFrom2DCurve
TORQUE_BASE_SPEED	IRON_LOSS_ROTOR_BASE_SPEED
	IRON_LOSS_STATOR_BASE_SPEED
	JOULE_LOSSES_BASE_SPEED
	MAGNET_LOSSES_BASE_SPEED
	TORQUE_BASE_SPEED
	USEFUL_POWER_BASE_SPEED

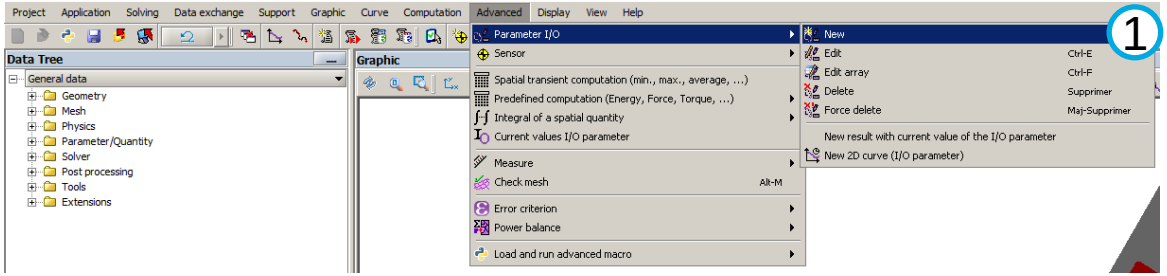
ALTAIR

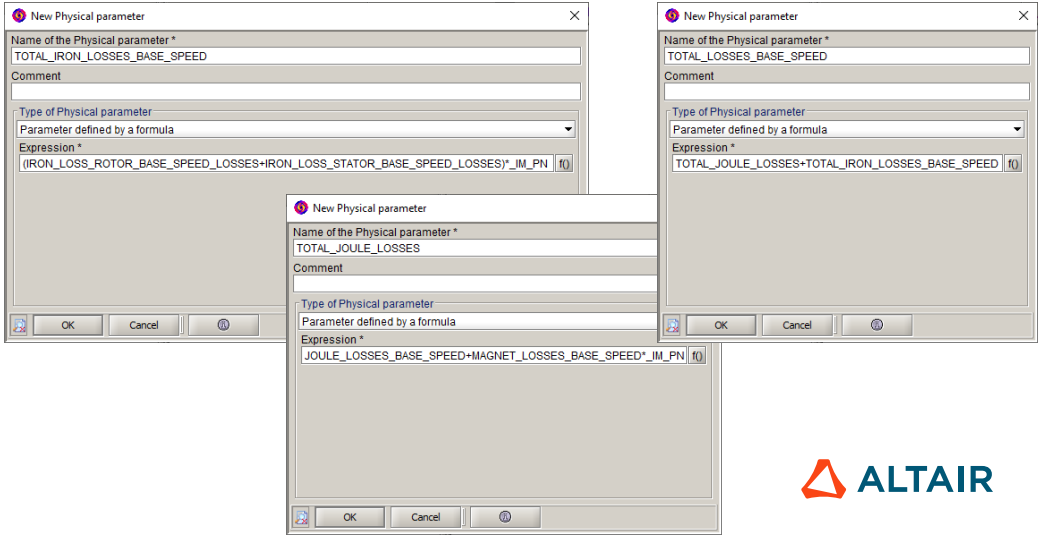
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: motor efficiency at base speed point

- Define variation parameters

Step	Action
1	Click on [Advanced] – [Parameter I/O] – [New]
2	Create three new variation parameters from previous curves - TOTAL_IRON_LOSSES_BASE_SPEED - TOTAL_JOULE_LOSSES - TOTAL_LOSSES_BASE_SPEED





Parameter name	Formula expression
TOTAL_IRON_LOSSES_BASE_SPEED	(IRON_LOSS_ROTOR_base_speed+IRON_LOSS_STATOR_base_speed)*_IM_PN
TOTAL_JOULE_LOSSES	JOULE_LOSSES_base_speed+MAGNET_LOSSES_base_speed*_IM_PN
TOTAL_LOSSES_BASE_SPEED	TOTAL_Joule_losses+Total_Iron_losses_base_speed

ALTAIR

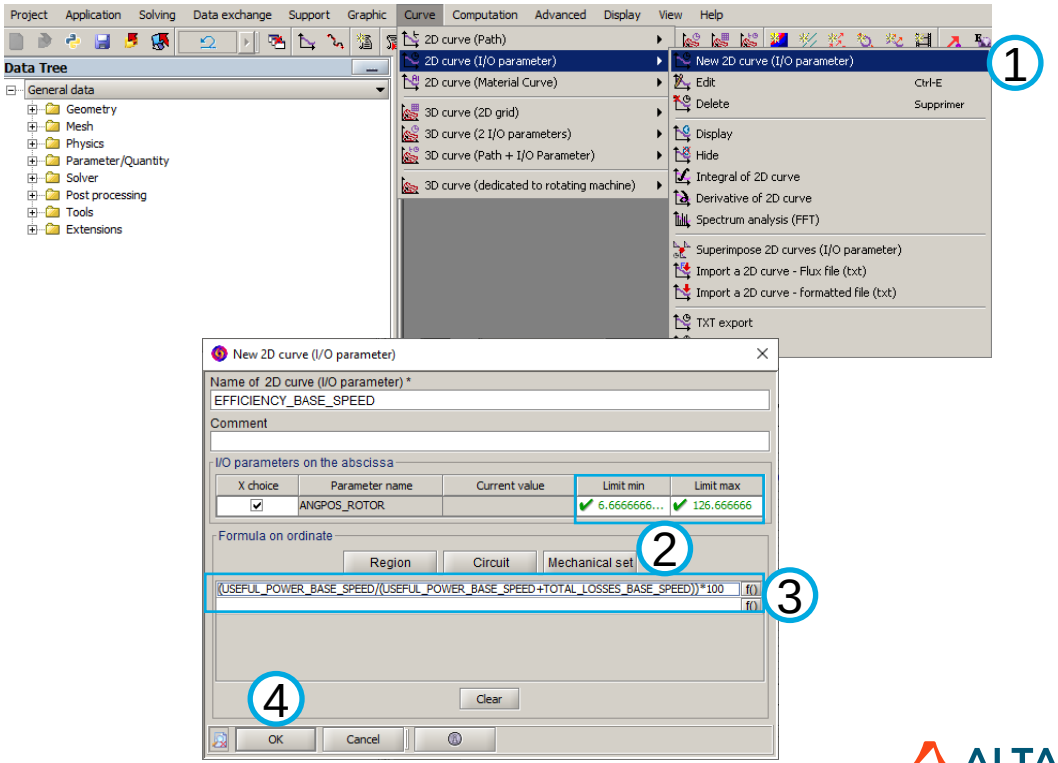
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: motor efficiency at base speed point

- Plot the motor efficiency curve

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [New 2D curve (I/O parameter)]
2	Create a new 2D curve EFFICIENCY_BASE_SPEED Limit min: 6.666666 Limit max: 126.666666
3	Define the plotting formula as (USEFUL_POWER_BASE_SPEED/(USEFUL_POWER_BASE_SPEED+TOTAL_LOSSES_BASE_SPEED))*100
4	Click on [OK]

Curve name	EFFICIENCY_BASE_SPEED
Formula	(USEFUL_POWER_BASE_SPEED/(USEFUL_POWER_BASE_SPEED+TOTAL_LOSSES_BASE_SPEED))*100



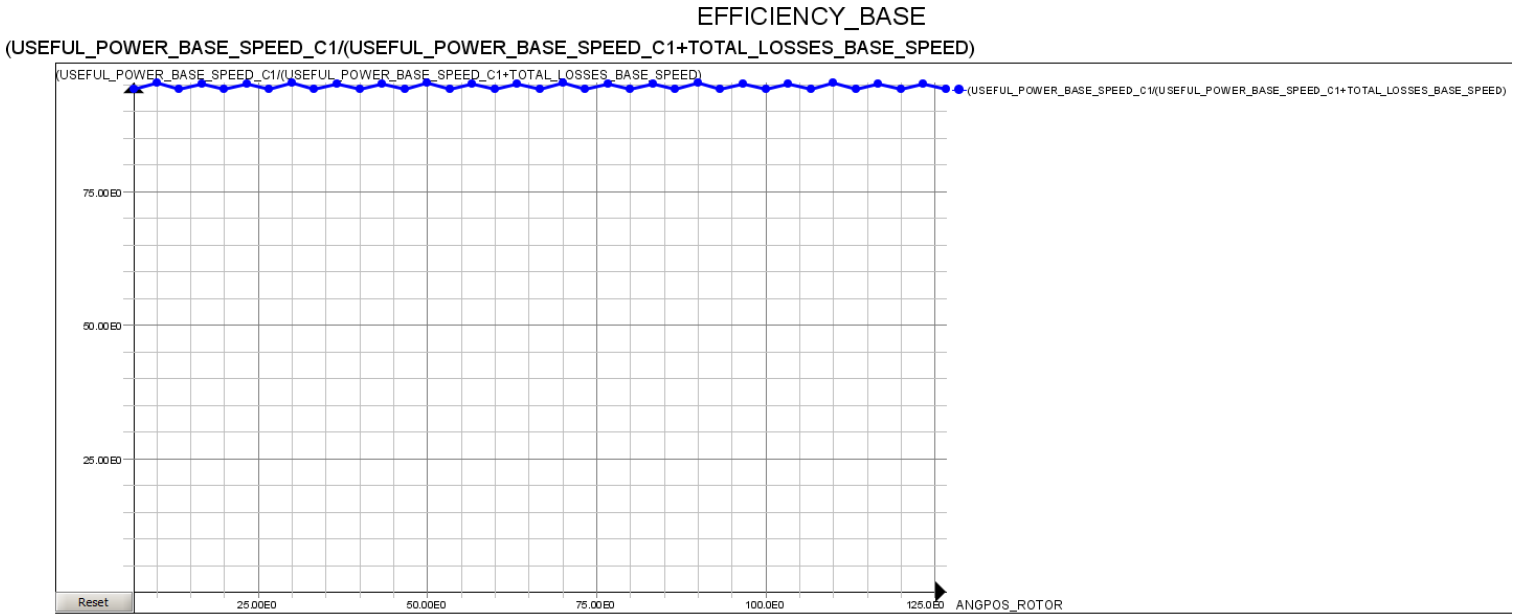
Tips: these three parameters are to calculate motor efficiency

ALTAIR

SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: motor efficiency at base speed point

- Plot the motor efficiency curve



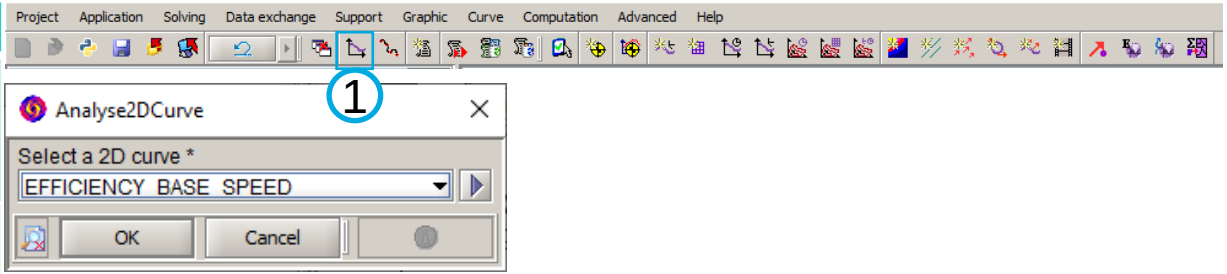
ALTAIR

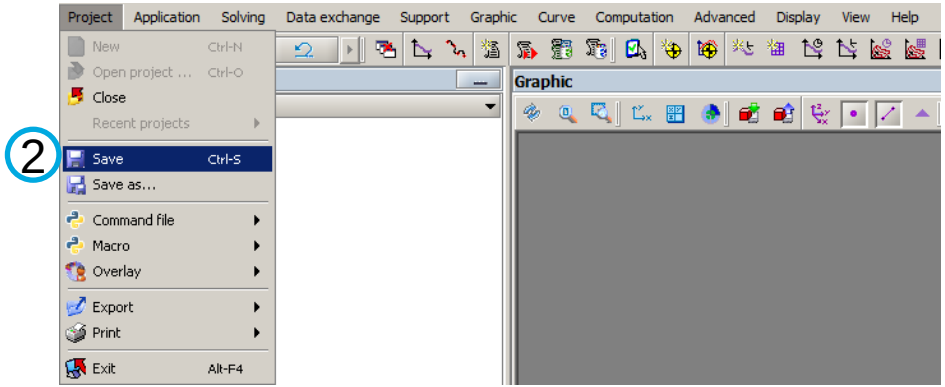
SOLVING AND POSTPROCESSING: BASE SPEED POINT

Magnetic analysis: motor efficiency at base speed point

- Analyze the efficiency curve with macros

Step	Action
1	Click on the icon to run the macro “Analyse2DCurve”, select “EFFICIENCY_BASE_SPEED”
2	Click on [Project] – [Save] to save the project



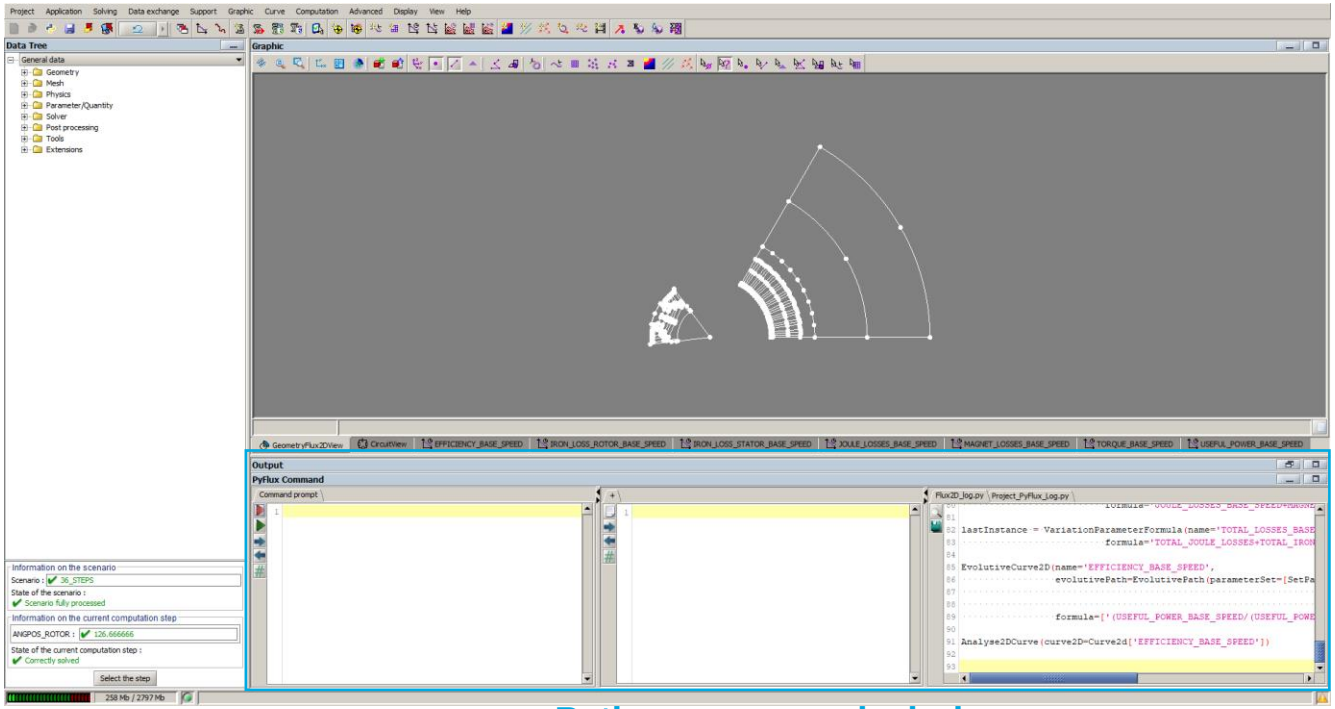


ALTAIR

GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Preparing post-processing Python script for HyperStudy connector



Python command windows

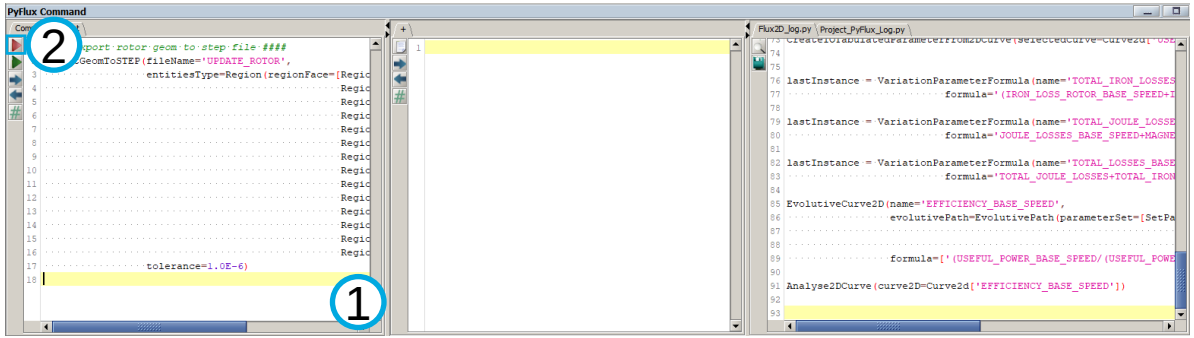
ALTAIR

GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Preparing post-processing Python script for HyperStudy connector

- Run scripts to export the rotor geometry file

Step	Action
1	Copy the following scripts and paste in the [Command prompt] windows
2	Click on the icon  to execute the scripts




```
selectCurrentStep(activeScenario=Scenario['36_STEPS'],
                  parameterValue=['ANGPOS_ROTOR=59.9999996842105'])

#### Export rotor geom to step file ####
exportGeomToSTEP(fileName='UPDATE_ROTOR',
                 entitiesType=Region(regionFace=[RegionFace['IM_YOKE'],
                                                 RegionFace['IM_EDGE'],
                                                 RegionFace['IM_MAGNET2A_1'],
                                                 RegionFace['IM_MAGNET2B_1'],
                                                 RegionFace['IM_MAGNET2C_1'],
                                                 RegionFace['IM_MAGNET1A_1'],
                                                 RegionFace['IM_MAGNET1B_1'],
                                                 RegionFace['IM_MAGNET1C_1'],
                                                 RegionFace['IM_MAGNET1C_SYM_1'],
                                                 RegionFace['IM_MAGNET1B_SYM_1'],
                                                 RegionFace['IM_MAGNET1A_SYM_1'],
                                                 RegionFace['IM_MAGNET2C_SYM_1'],
                                                 RegionFace['IM_MAGNET2B_SYM_1'],
                                                 RegionFace['IM_MAGNET2A_SYM_1']]),
                 tolerance=1.0E-6)
```

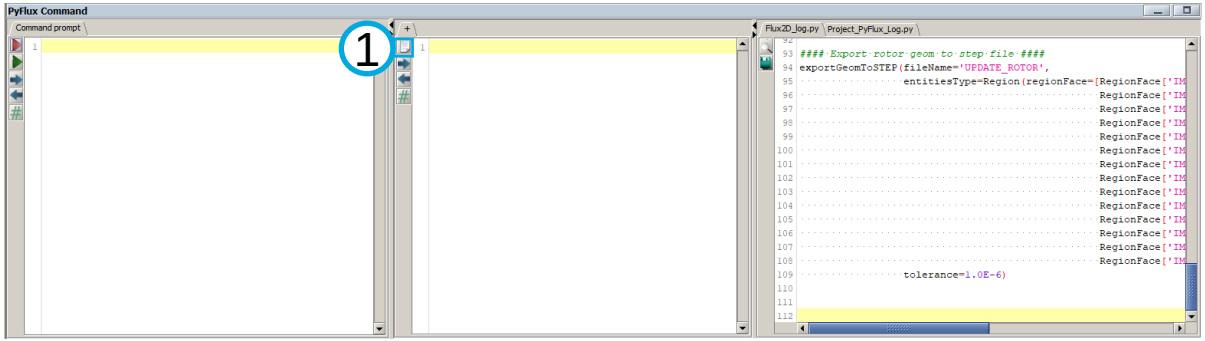

ALTAIR

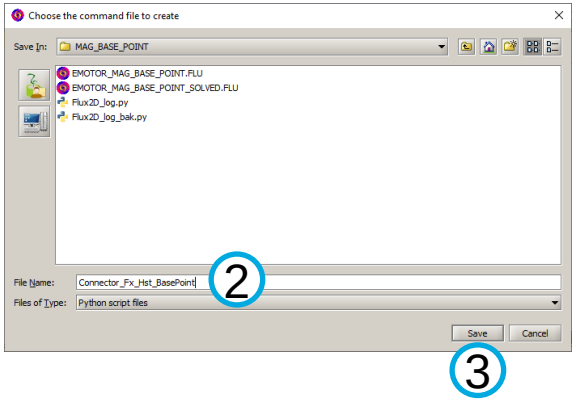
GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Preparing post-processing Python script for HyperStudy connector

- Create a new Python script

Step	Action
1	Click on the icon to create a new Python script file
2	Define the script name and the saving folder
3	Click on [Save]





Script name	Connector_Fx_Hst_BasePoint.py
Saving folder	~/MDO_EMOTOR/Flux/FluxProjects/ MAG_BASE_POINT

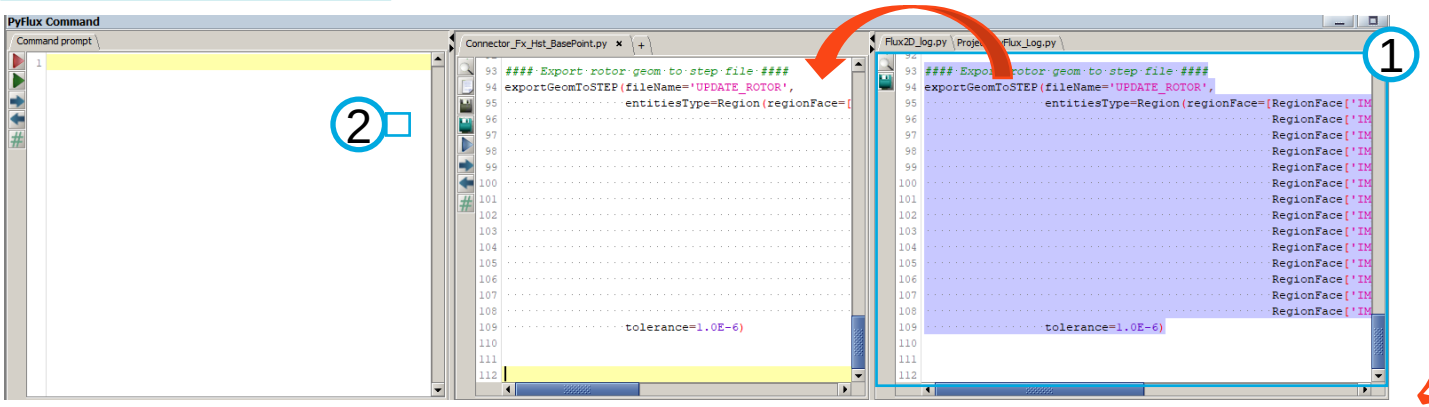
ALTAIR

GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Preparing post-processing Python script for HyperStudy connector

- Create backup of the log script

Step	Action
1	Copy all the script from the Flux2D_log.py file into the created Python script
2	Click on the icon  to save the script
3	Click on [Project] – [Exit]



ALTAIR

GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Preparing post-processing Python script for HyperStudy connector

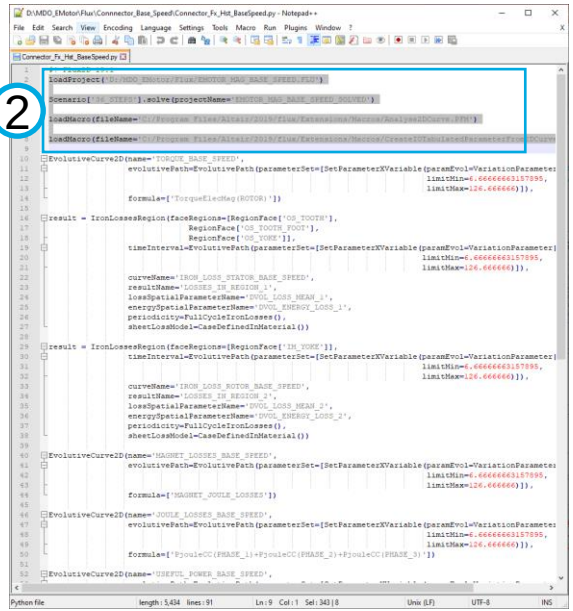
- Replace Flux macro import commands

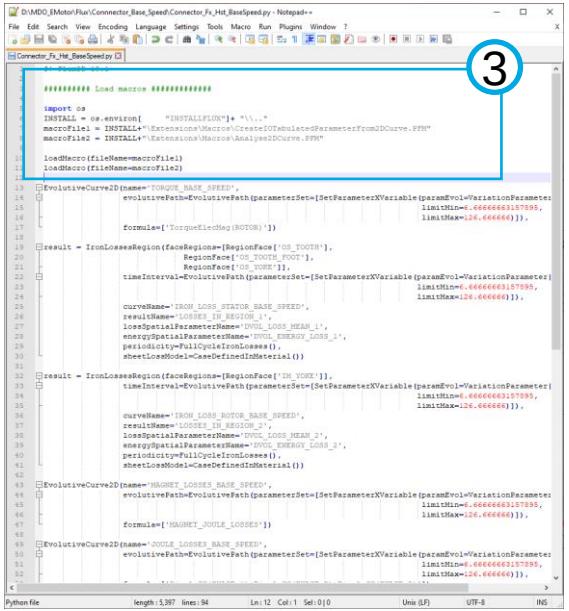
Step	Action
1	Open the Python script by Notepad editor
2	Delete the scripts before post-processing
3	Insert the following scripts before post-processing

########## Load macros ##############

```
import os
INSTALL = os.environ[     "INSTALLFLUX"]+ "\\.."
macroFile1 =
INSTALL+"\Extensions\Macros\CreateIOTabulatedParameterFrom2D
Curve.PFM"
macroFile2 = INSTALL+"\Extensions\Macros\Analyse2DCurve.PFM"

loadMacro(fileName=macroFile1)
loadMacro(fileName=macroFile2)
```





Attention: 1) the project will be solved automatically by HyperStudy;
2) the Flux macro location is depended on the Flux installation path. Therefore, these scripts should be replaced by a generic way to adapted to all the user.

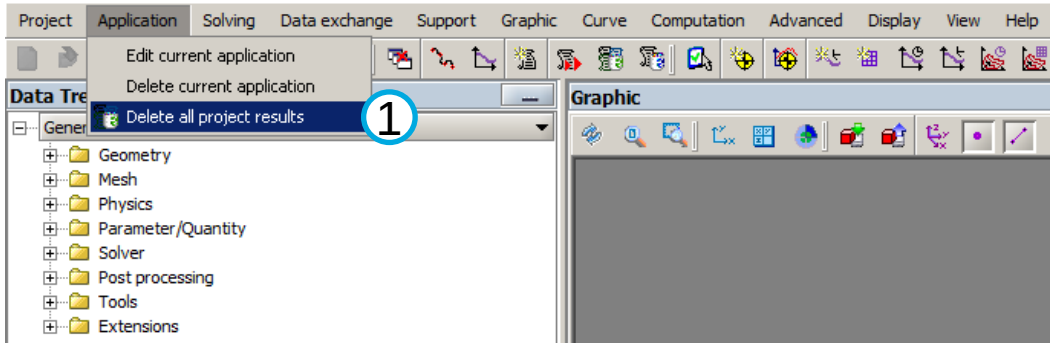
ALTAIR

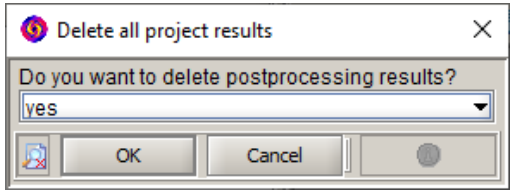
GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Generating HyperStudy connector

- Delete all project results

Step	Action
1	Click on [Application] – [Delete all project results]
2	Select [Yes], and click on [OK]





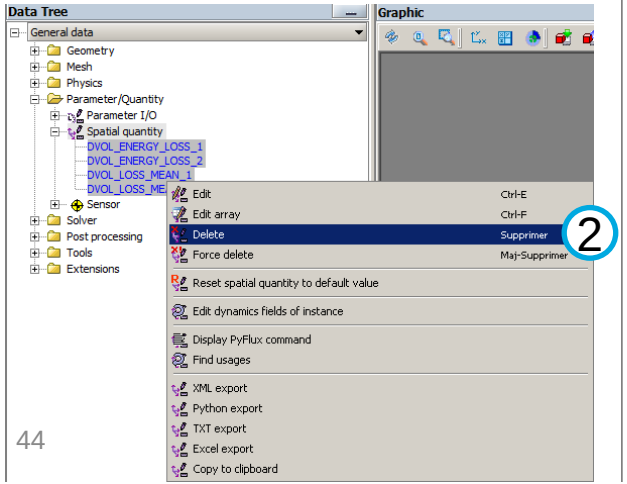
ALTAIR

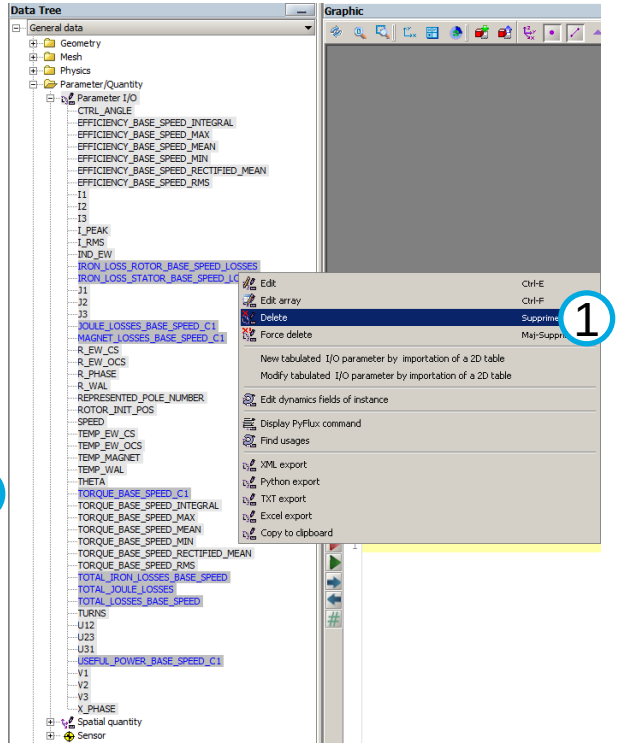
GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

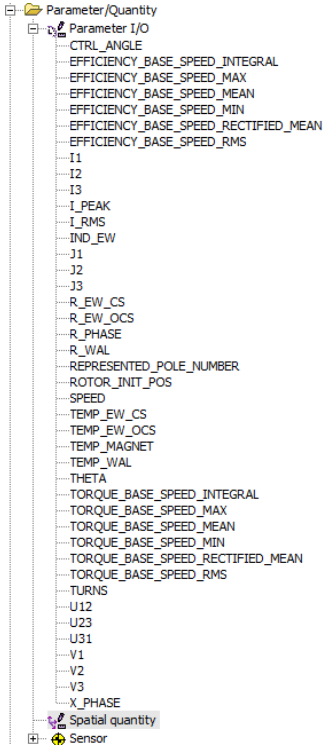
Generating HyperStudy connector

- Delete parameters and spatial quantities not needed in connector

Step	Action
1	Select the following 9 I/O parameters, right click and click on [Delete]
2	Select the 4 spatial quantities, right click and click on [Delete]







ALTAIR

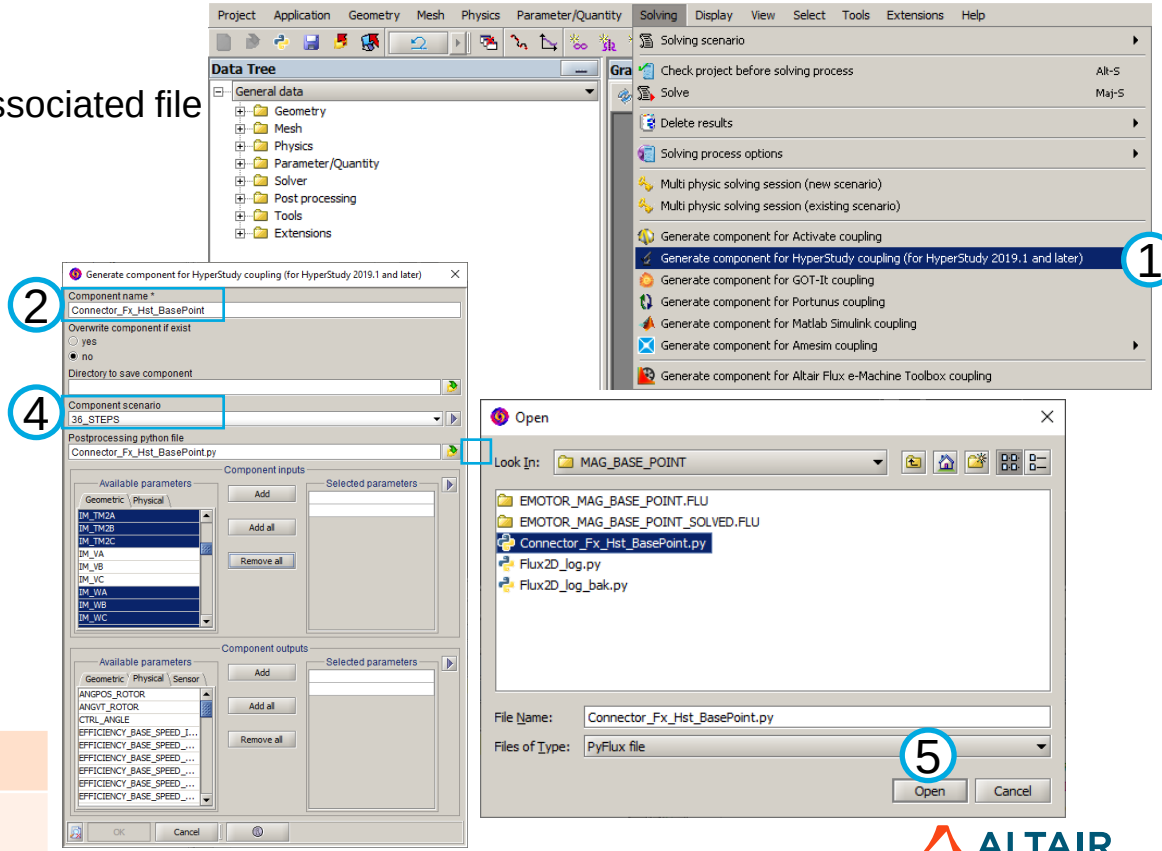
GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Generating HyperStudy connector

- Define connector name and associated file

Step	Action
1	Click on [Solving] – [Generate component for HyperStudy coupling (for HyperStudy 2019.1 and later)]
2	Define the component name as “Connector_Fx_Hst_BasePoint”
3	Directory by default
4	Define the component scenario “36_STEPS”
5	Select the “Connector_Fx_Hst_BasePoint.py” as the postprocessing Python file

Connector name	Connector_Fx_Hst_BasePoint
Postprocessing Python file	Connector_Fx_Hst_BasePoint.py



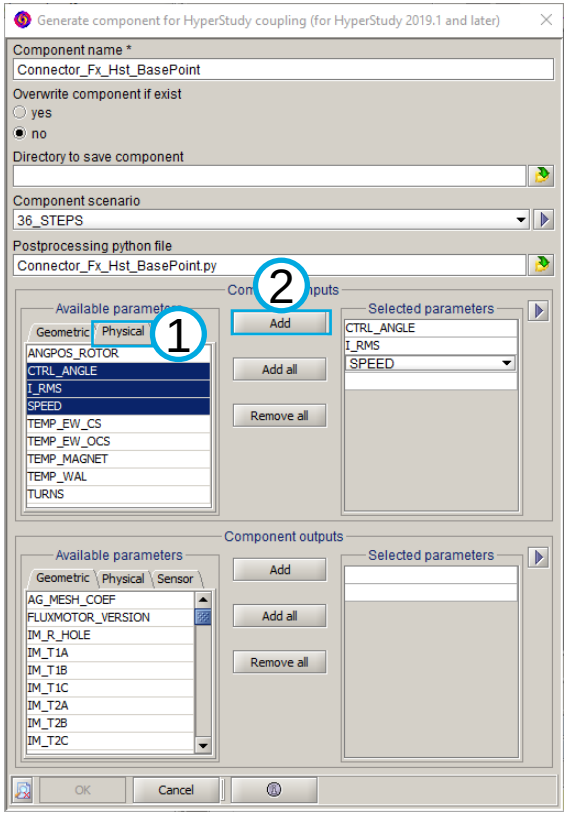
ALTAIR

GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Generating HyperStudy connector

- Define component inputs (physical)

Step	Action
1	Click on [Physical] tab, select the following three parameters: - CTRL_ANGLE - I_RMS - SPEED
2	Click on [Add]



ALTAIR

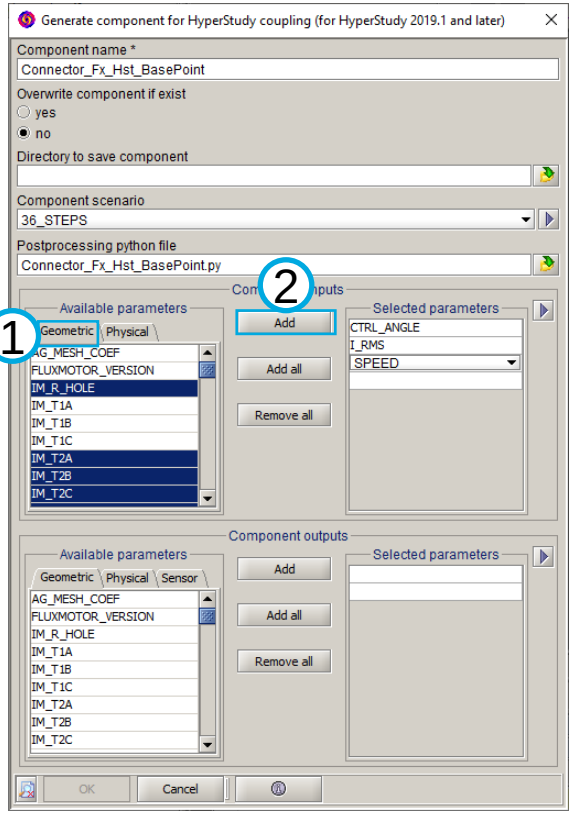
GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Generating HyperStudy connector

- Define component inputs (geometric)

Step	Action
1	Click on [Geometric] tab, select the following 22 parameters in the table
2	Click on [Add]

Imput geometric variables	
IM_R_HOLE	IM_TM2B
IM_T2A	IM_TM2C
IM_T2B	IM_WA
IM_T2C	IM_WB
IM_T3A	IM_WC
IM_T3B	IM_WM1A
IM_T3C	IM_WM1B
IM_TM1A	IM_WM1C
IM_TM1B	IM_WM2A
IM_TM1C	IM_WM2B
IM_TM2A	IM_WM2C



ALTAIR

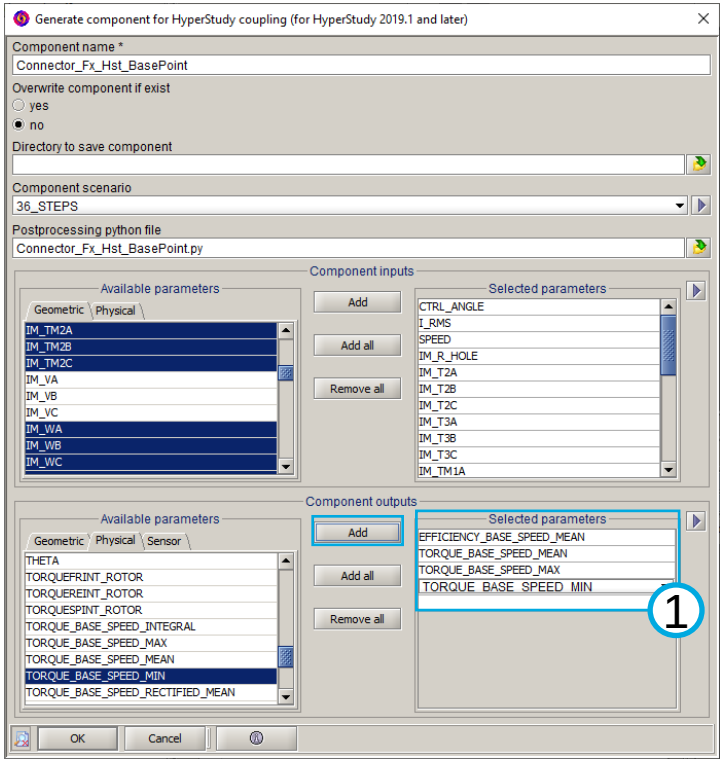
GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Generating HyperStudy connector

- Define component outputs

Step	Action
1	Click on [Physical] tab in the output option
2	Select the following parameters and click on [Add]

Output variables
EFFICIENCY_BASE_MEAN
TORQUE_BASE_SPEED_MEAN
TORQUE_BASE_SPEED_MAX
TORQUE_BASE_SPEED_MIN



ALTAIR

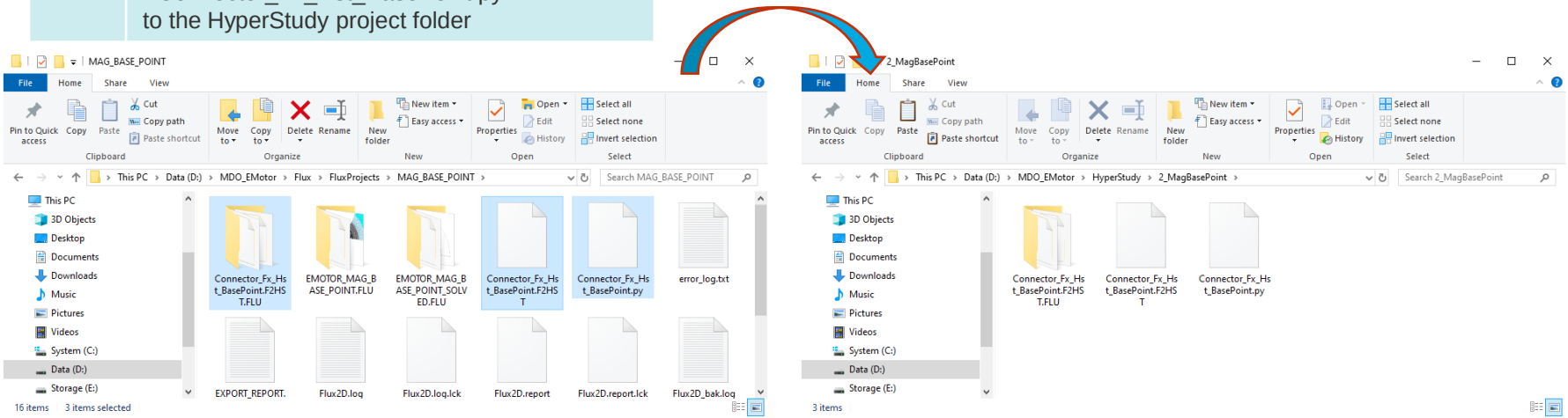
GENERATING HYPERSTUDY CONNECTOR: BASE SPEED POINT

Generating HyperStudy connector

- Exchange files for the Flux / HyperStudy connector

Step	Action
1	Copy and paste the connector files: - Connector_Fx_Hst_BasePoint.F2HST - Connector_Fx_Hst_BasePoint.F2HST.FLU - Connector_Fx_Hst_BasePoint.py to the HyperStudy project folder

Path for saving the Flux magnetic analysis connector 1	~\MDO_EMotor\HyperStudy\2_BasePoint



MAGNETIC ANALYSIS: SPECIFIC OPERATING POINT

OUTLINE

Solving and postprocessing: specific operating point

Generating HyperStudy connector: specific operating point

Input file

- **Flux 2D project**:
  - EMOTOR_MAG_OPERATING_POINT.FLU

Software

- Altair Flux 2019.1 (or later version)

Output documents

- **Flux / HyperStudy connector**:
  Connector_Fx_Hst_OperatingPoint.F2HST
- **Flux project associated with the connector**
  Connector_Fx_Hst_OperatingPoint.F2HST.FLU
- **Python script for the postprocessing in HyperStudy**
  Connector_Fx_Hst_OperatingPoint.py

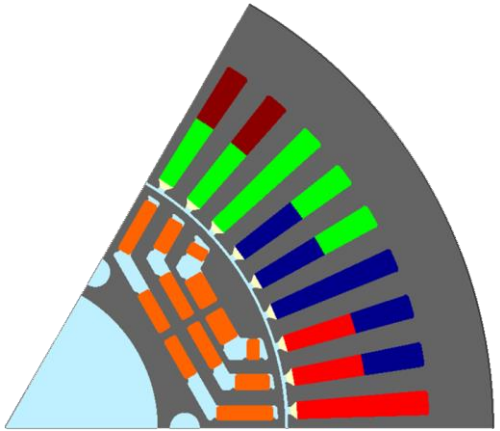
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

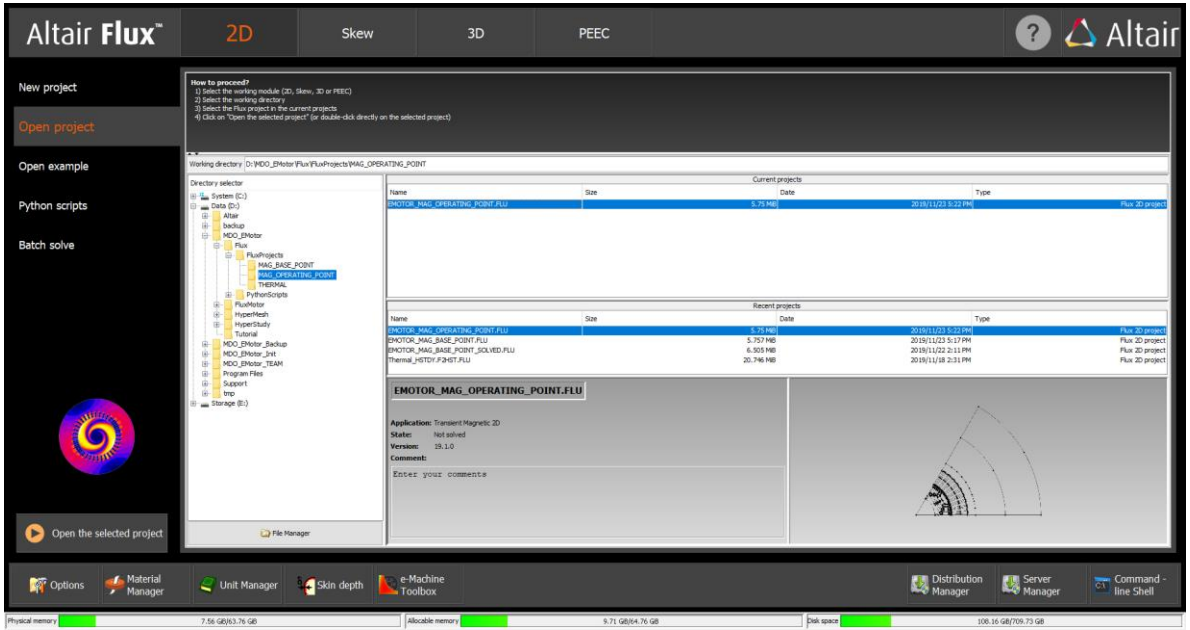
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Solving the magnetic problem at specific operating point

- Open Flux 2D project

Step	Action
1	Open the project “EMOTOR_MAG_OPERATING_POINT.FLU”





Project name	EMOTOR_MAG_OPERATING_POINT.FLU
Project folder	~/MDO_EMOTOR/Flux/FluxProjects/ MAG_OPERATING_POINT

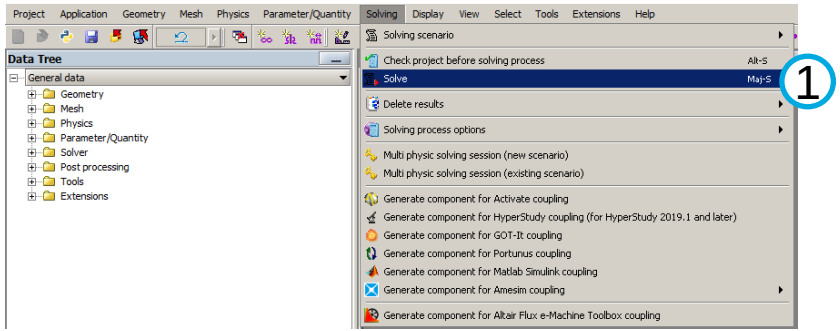
ALTAIR

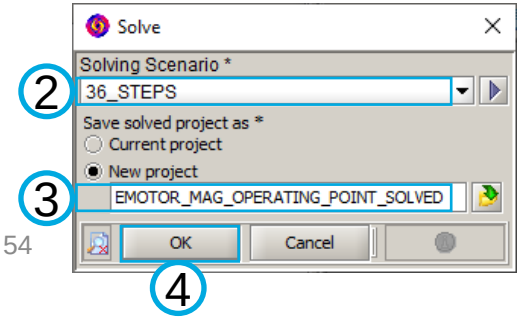
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

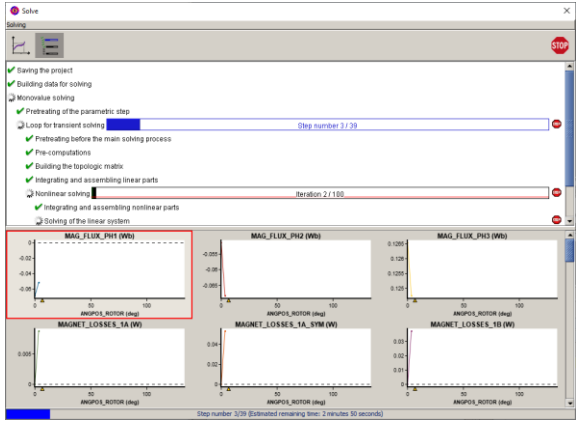
Solving the magnetic problem at specific operating point

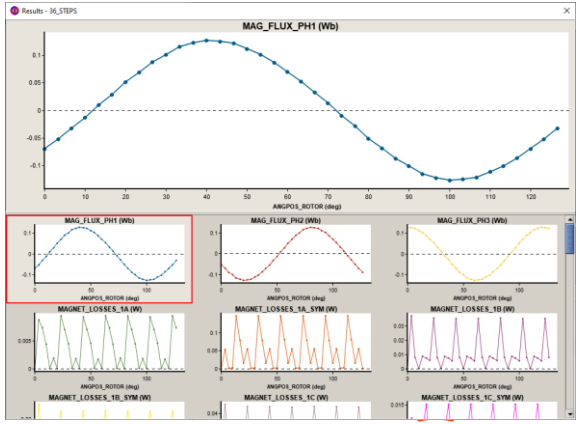
- Solve the project

Step	Action
1	Click on [Solving] – [Solve]
2	Select the solving scenario “36_STEPS”
3	Save the solved project as a new project “EMOTOR_MAG_OPERATING_POINT_SOLVED.FLU”
4	Click on [OK]







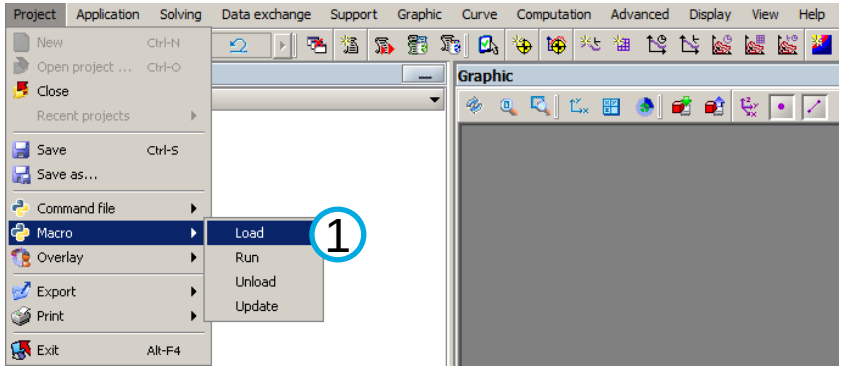


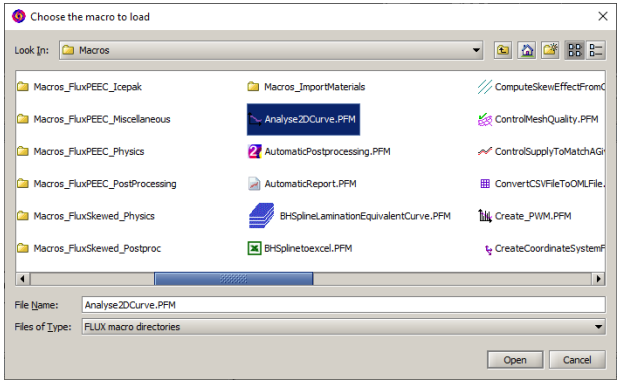
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

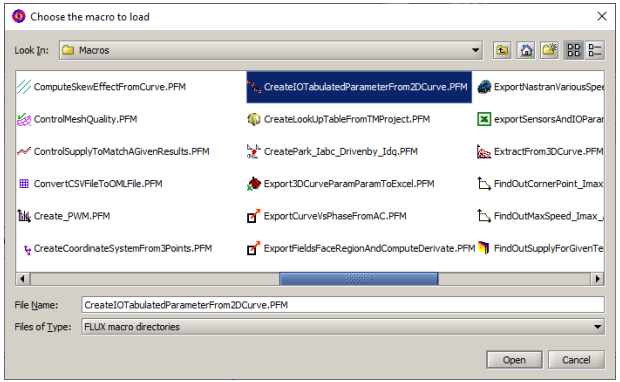
Magnetic analysis: postprocessing initialization

- Load Flux macros

Step	Action
1	Click on [Project] – [Macro] – [Load]
2	Load the following two macros: - Analyse2DCurve - CreateIOTabulated ParameterFrom2DCurve







ALTAIR

SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

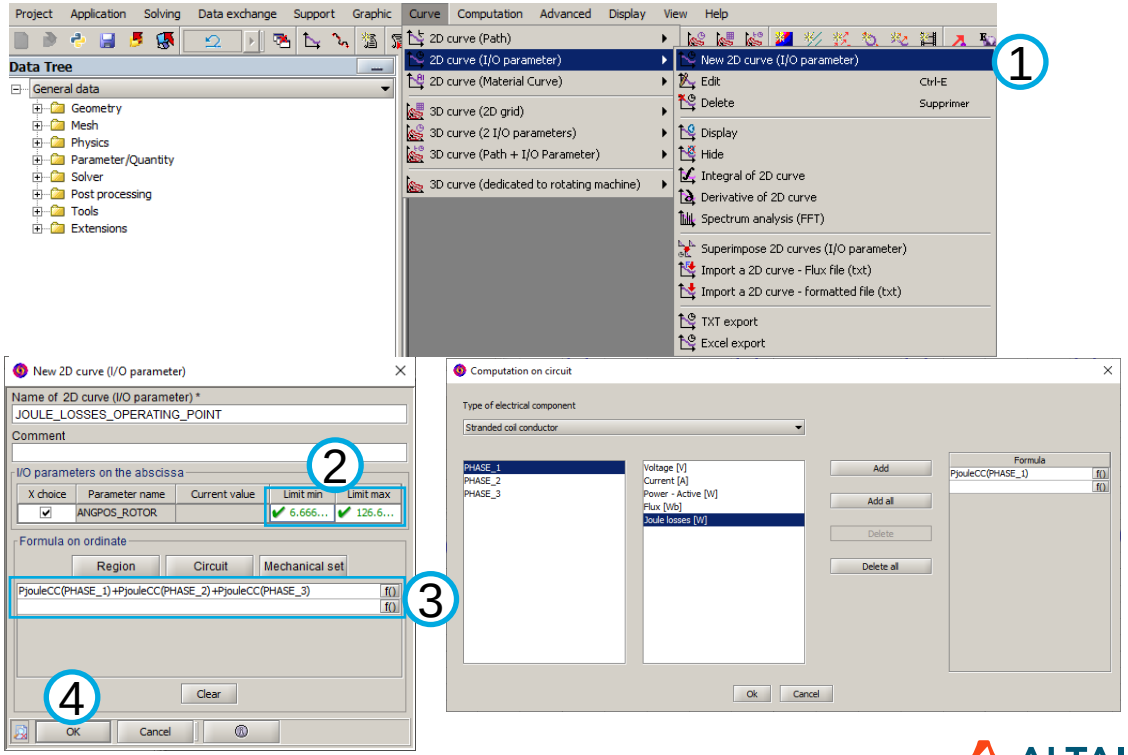
Joule losses | Stator iron losses | Rotor iron losses | Magnet losses

Magnetic analysis: Joule losses at the specific operating point

- Plot the Joule loss curve

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [New 2D curve (I/O parameter)]
2	Create a new 2D curve JOULE_LOSSES_OPERATING_POINT Limit min: 6.666666 Limit max: 126.666666
3	Define the plotting formula as PjouleCC(PHASE_1)+PjouleCC(PHASE_2)+PjouleCC(PHASE_3)
4	Click on [OK]

Curve name	JOULE_LOSSES_OPERATING_POINT
Formula	PjouleCC(PHASE_1)+PjouleCC(PHASE_2)+PjouleCC(PHASE_3)



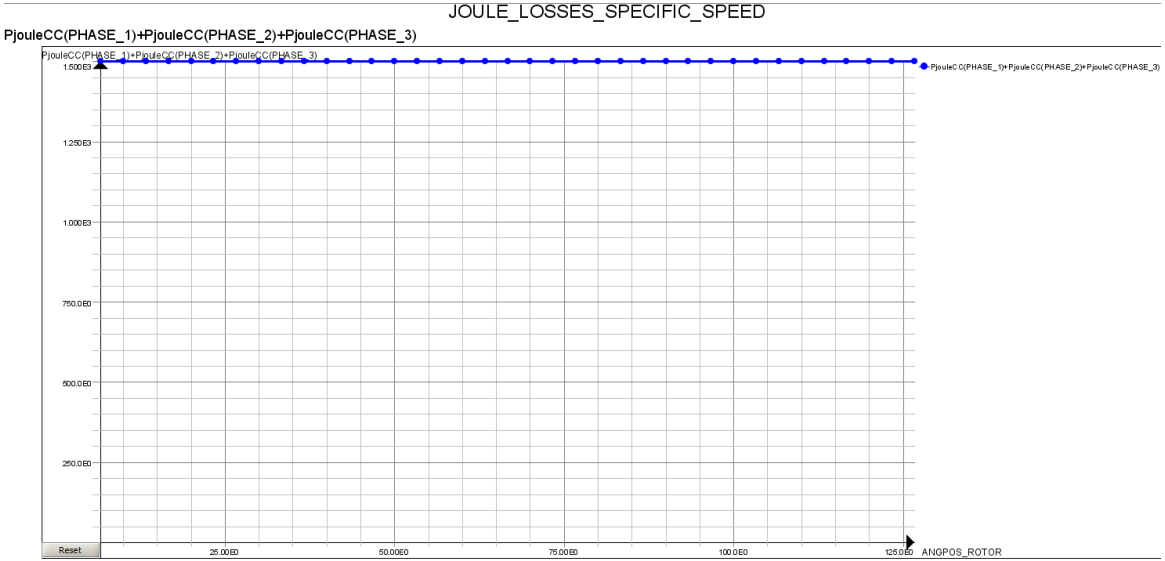
ALTAIR

SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Joule losses | Stator iron losses | Rotor iron losses | Magnet losses

Magnetic analysis: Joule losses at the specific operating point

- Plot the Joule loss curve



ALTAIR

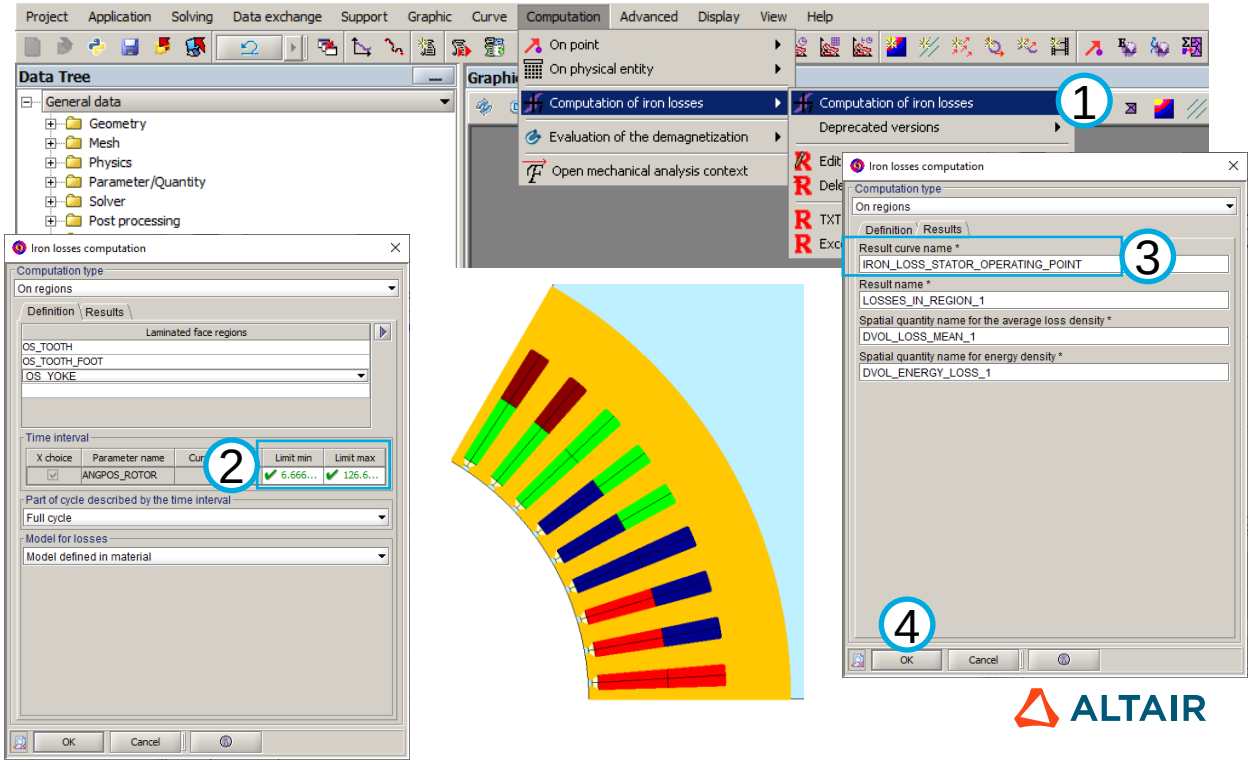
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Magnetic analysis: stator iron losses at the specific operating point

Joule losses | Stator iron losses | Rotor iron losses | Magnet losses

- Plot the stator iron loss curve

Step	Action
1	Click on [Computation] – [Computation of iron losses] – [Computation of iron losses]
2	Define the computation configuration in tab [Definition]: Face region: - OS_TOOTH - OS_TOOTH_FOOT - OS_YOKE Interval: [6.6666, 126.6666]
3	Define the computation configuration in tab [Results]: Result curve name: IRON_LOSS_STATOR_OPERATING_POINT
4	Click on [OK]



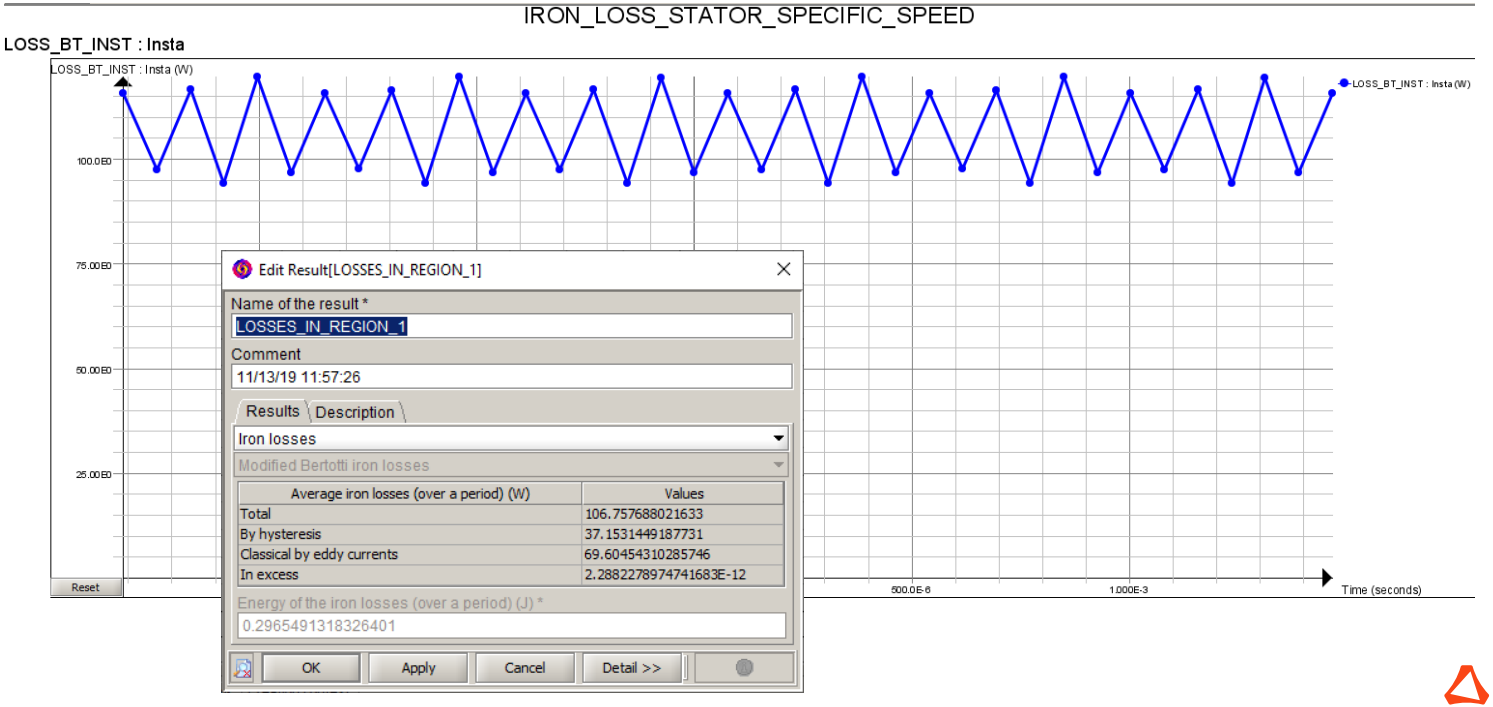
ALTAIR

SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Magnetic analysis: stator iron losses at the specific operating point

Joule losses | Stator iron losses | Rotor iron losses | Magnet losses

- Plot the stator iron loss curve

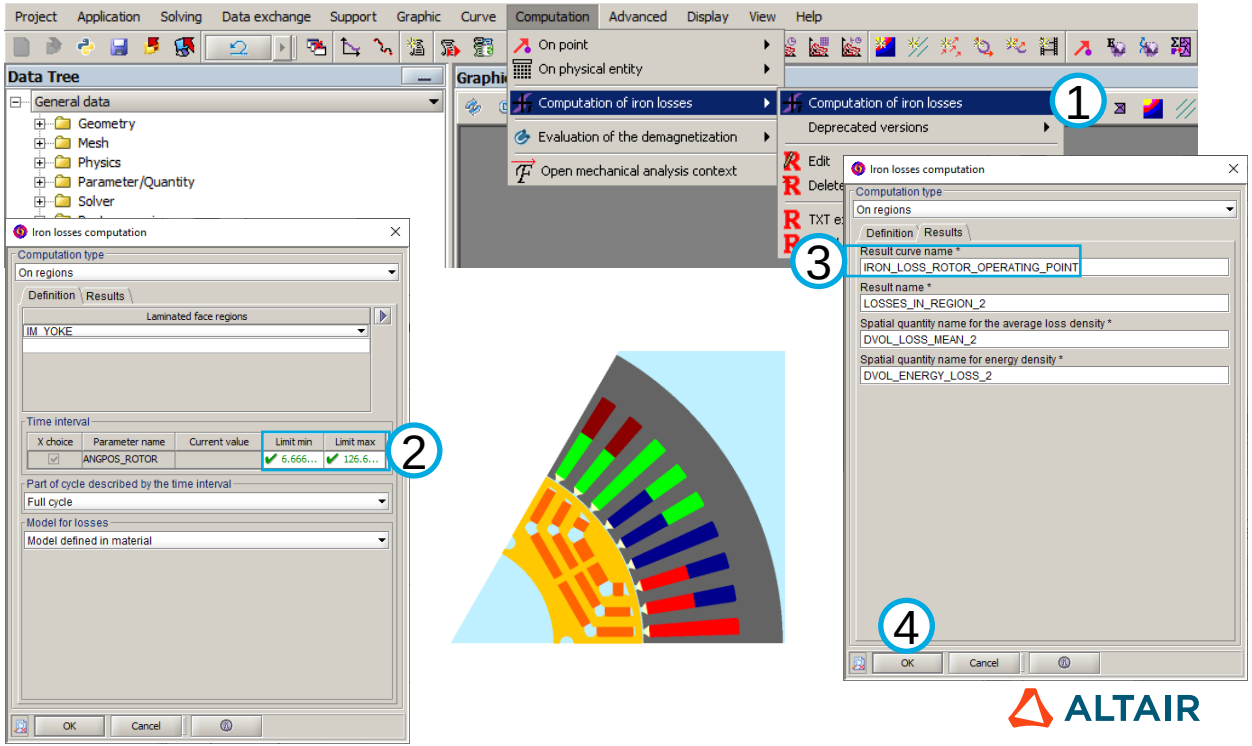


SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Magnetic analysis: rotor iron losses at the specific operating point

- Plot the rotor iron loss curve

Step	Action
1	Click on [Computation] – [Computation of iron losses] – [Computation of iron losses]
2	Define the computation configuration in tab [Definition]: Face region: - IM_YOKE Interval: [6.6666, 126.6666]
3	Define the computation configuration in tab [Results]: Result curve name: IRON_LOSS_ROTOR_OPERATING_POINT
4	Click on [OK]



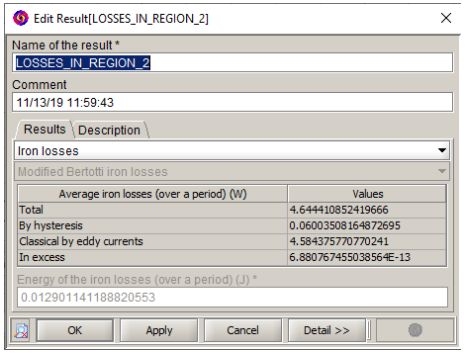
ALTAIR

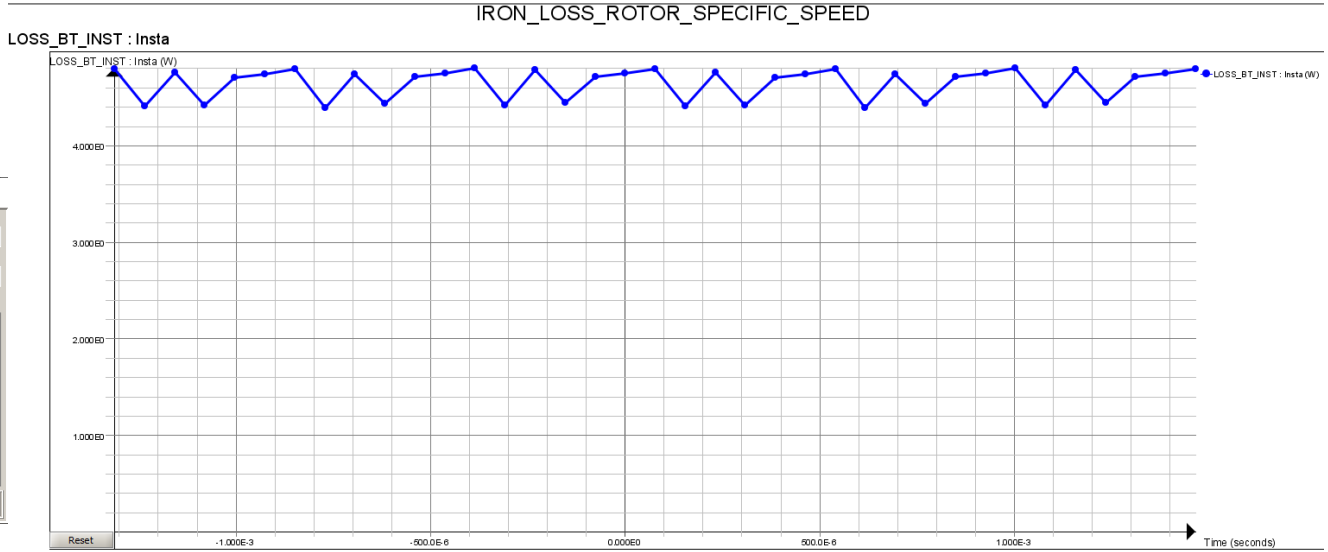
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Magnetic analysis: rotor iron losses at the specific operating point

Joule losses | Stator iron losses | Rotor iron losses | Magnet losses

- Plot the rotor iron loss curve





ALTAIR

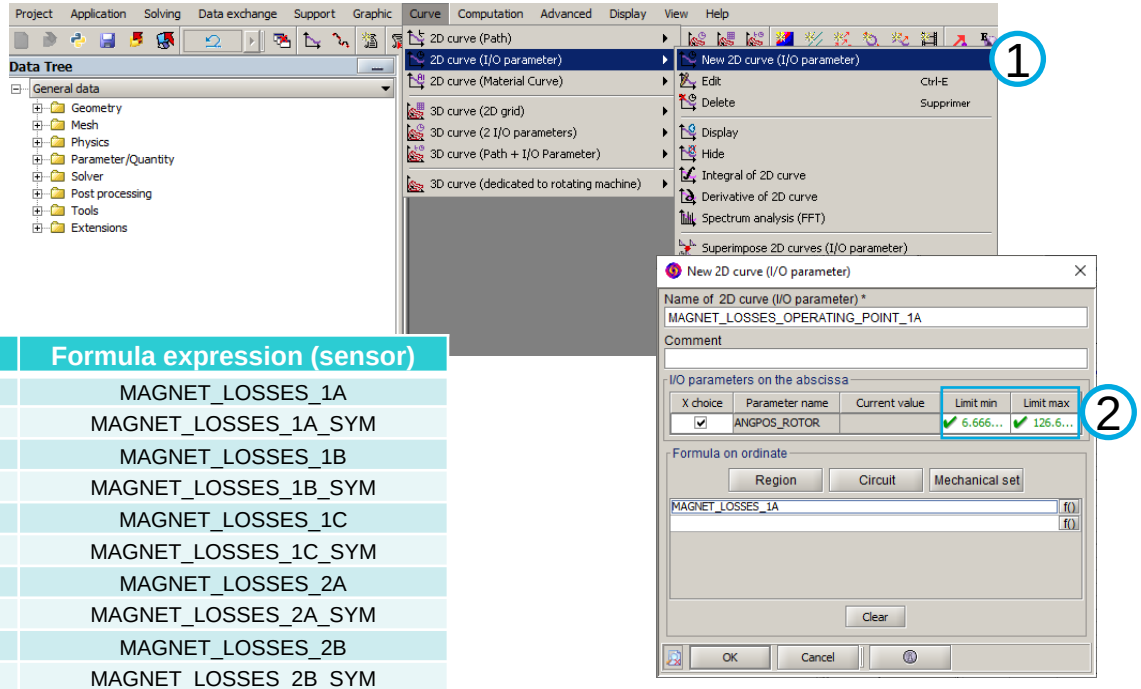
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Joule losses | Stator iron losses | Rotor iron losses | Magnet losses

Magnetic analysis: magnet losses at the specific operating point

- Plot magnet loss curves for all predefined magnet regions

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [New 2D curve (I/O parameter)]
2	Create new 2D curves with the following interval - Limit min: 6.666666 - Limit max: 126.666666



Curve name	Formula expression (sensor)
MAGNET_LOSSES_OPERATING_POINT_1A	MAGNET_LOSSES_1A
MAGNET_LOSSES_OPERATING_POINT_1A_SYM	MAGNET_LOSSES_1A_SYM
MAGNET_LOSSES_OPERATING_POINT_1B	MAGNET_LOSSES_1B
MAGNET_LOSSES_OPERATING_POINT_1B_SYM	MAGNET_LOSSES_1B_SYM
MAGNET_LOSSES_OPERATING_POINT_1C	MAGNET_LOSSES_1C
MAGNET_LOSSES_OPERATING_POINT_1C_SYM	MAGNET_LOSSES_1C_SYM
MAGNET_LOSSES_OPERATING_POINT_2A	MAGNET_LOSSES_2A
MAGNET_LOSSES_OPERATING_POINT_2A_SYM	MAGNET_LOSSES_2A_SYM
MAGNET_LOSSES_OPERATING_POINT_2B	MAGNET_LOSSES_2B
MAGNET_LOSSES_OPERATING_POINT_2B_SYM	MAGNET_LOSSES_2B_SYM
MAGNET_LOSSES_OPERATING_POINT_2C	MAGNET_LOSSES_2C
MAGNET_LOSSES_OPERATING_POINT_2C_SYM	MAGNET_LOSSES_2C_SYM

ALTAIR

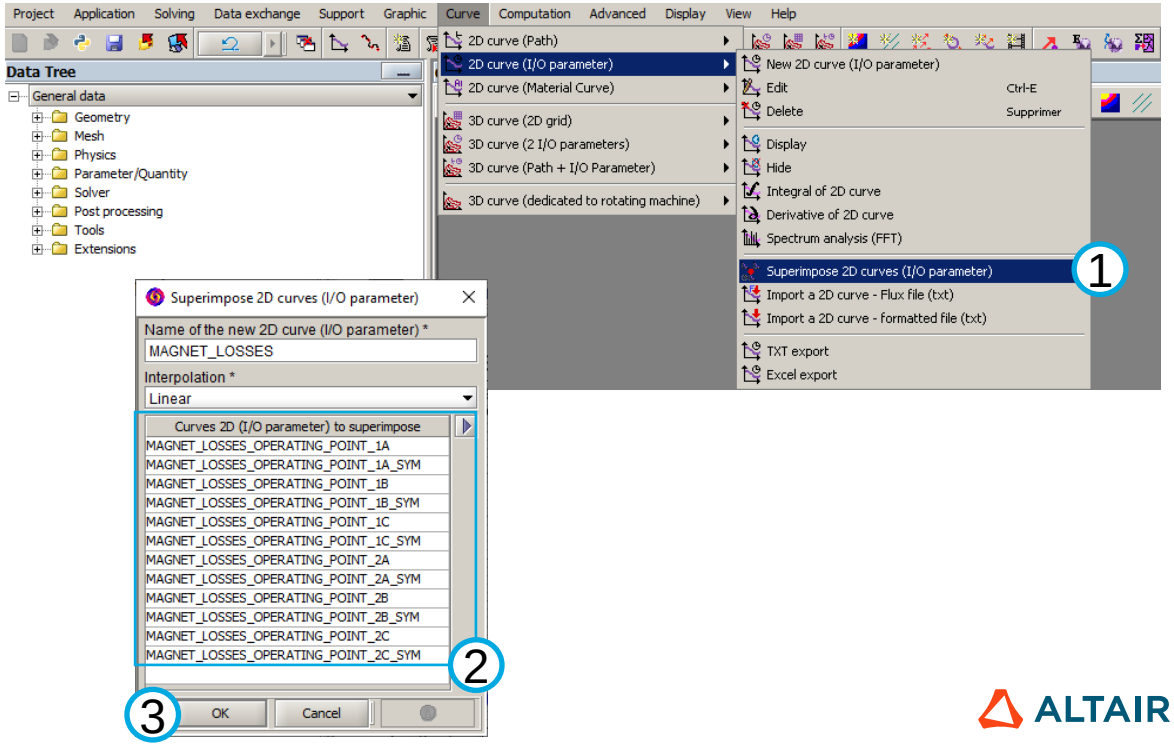
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

Joule losses | Stator iron losses | Rotor iron losses | **Magnet losses**

Magnetic analysis: magnet losses at the specific operating point

- Plot magnet loss curves for all predefined magnet regions

Step	Action
1	Click on [Curve] – [2D curve (I/O parameter)] – [Superimpose 2D curves ((I/O parameter)]
2	Select all the magnet loss curves
3	Click on [OK]



ALTAIR

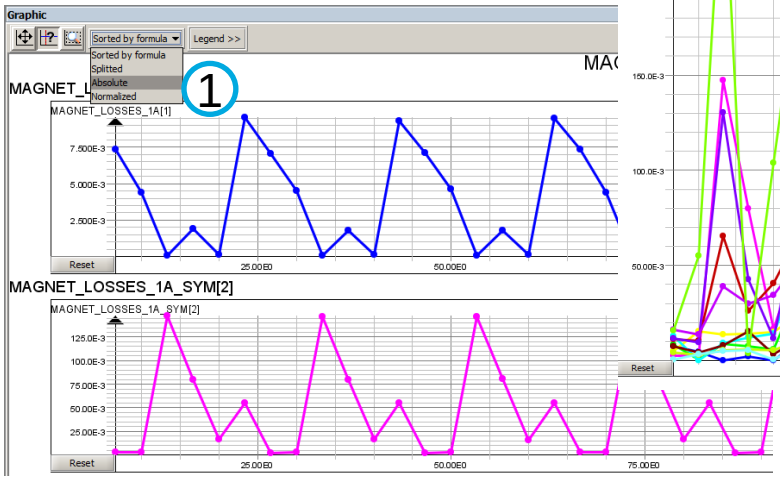
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

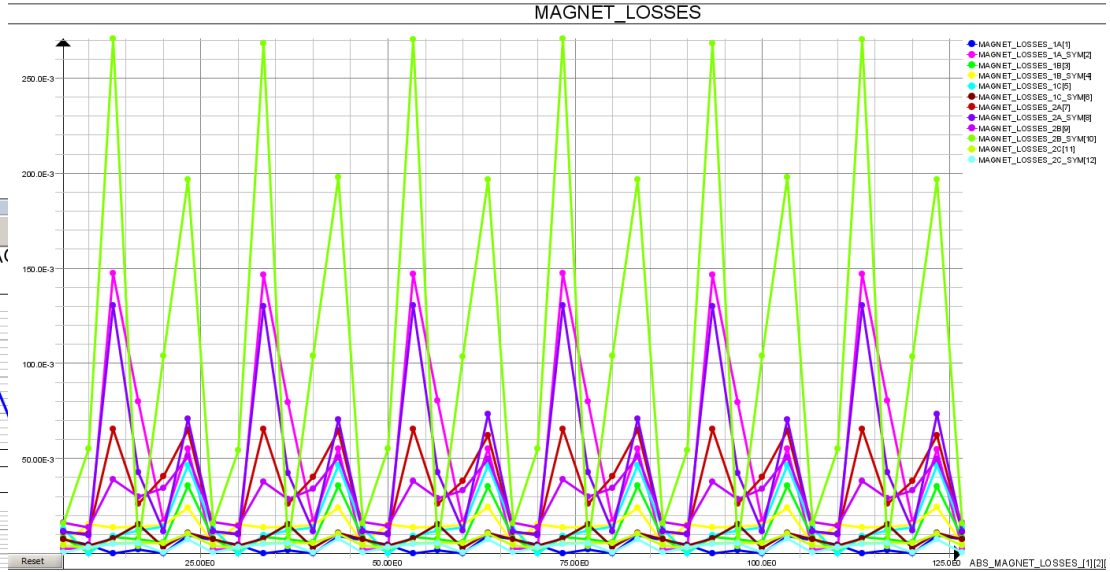
Joule losses | Stator iron losses | Rotor iron losses | **Magnet losses**

Magnetic analysis: magnet losses at the specific operating point

- Plot magnet loss curves for all predefined magnet regions

Step	Action
1	Select the plot configuration as “Absolute”



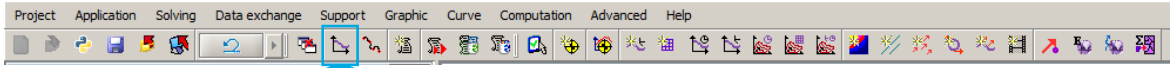


ALTAIR

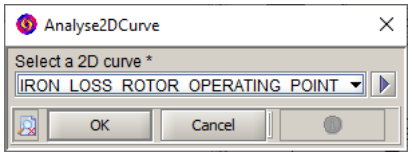
SOLVING AND POSTPROCESSING: SPECIFIC OPERATING POINT

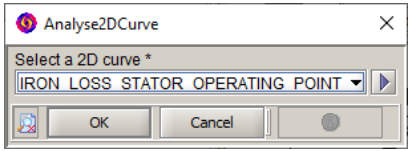
Magnetic analysis

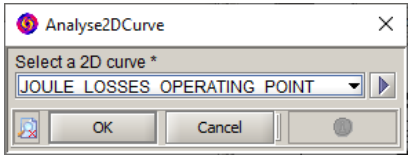
- Analyze the curves with macros

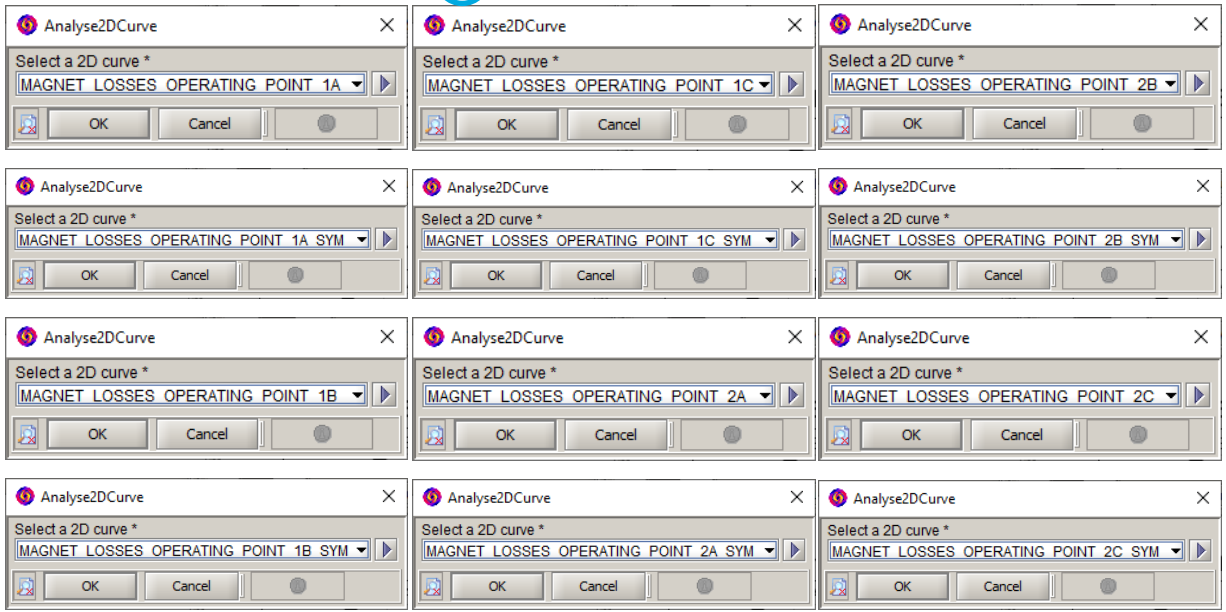


Step	Action
1	Click on the icon to run the macro “Analyse2DCurve”, and analyze respectively ALL the created curves (3+12=15)









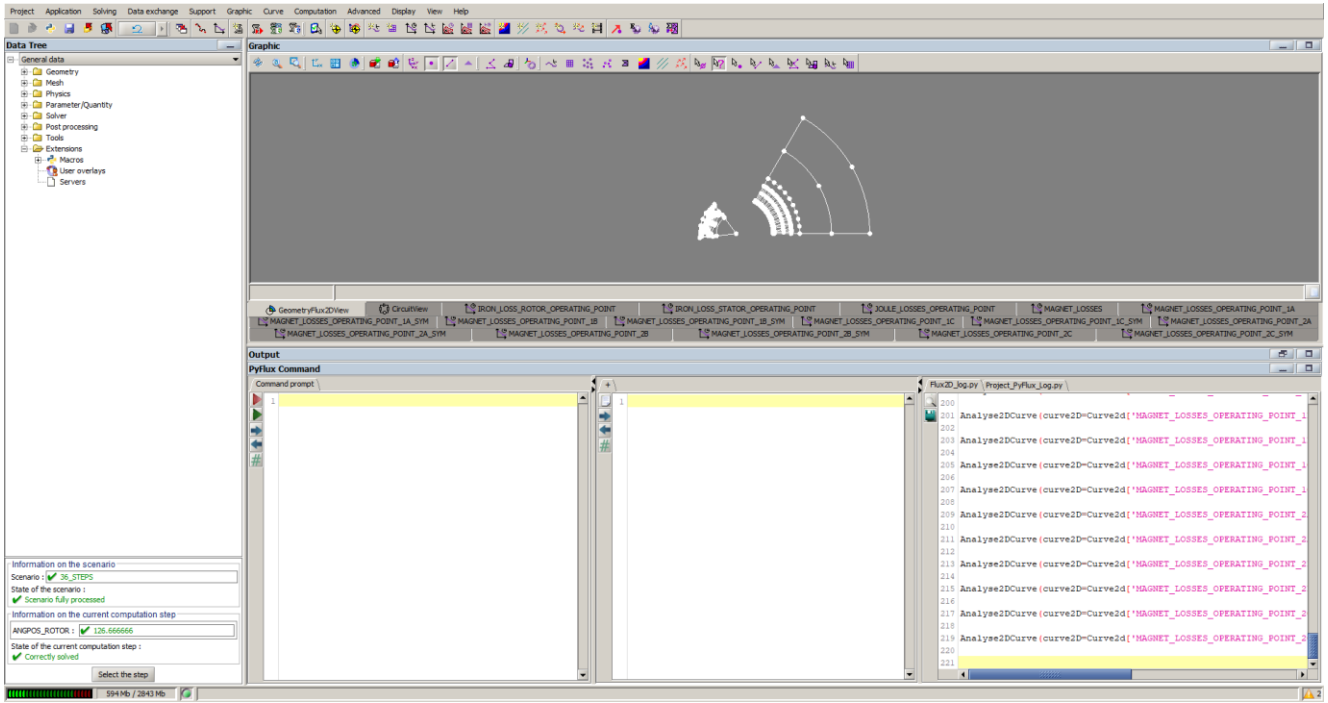
Attention: Do NOT select the superimpose curves “MAGNET_LOSSES”

ALTAIR

GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Preparing post-processing Python script for HyperStudy connector



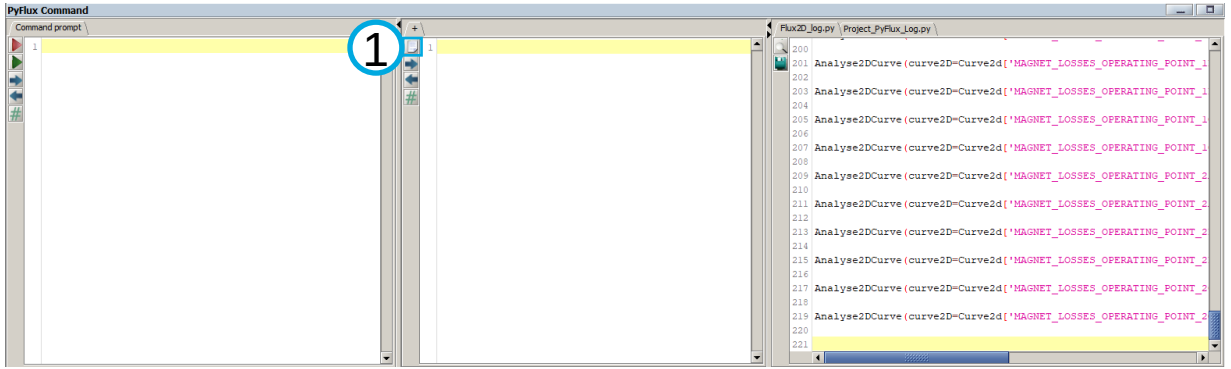
ALTAIR

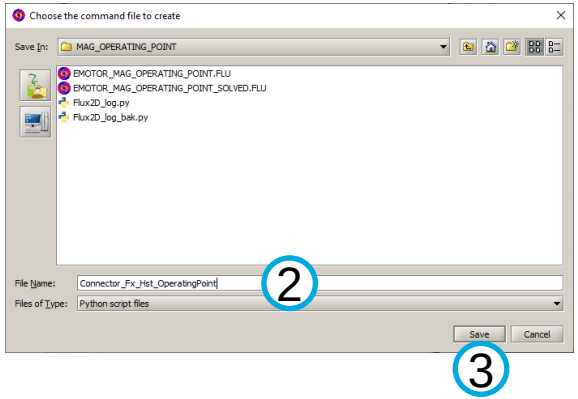
GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Preparing post-processing Python script for HyperStudy connector

- Create a new Python script

Step	Action
1	Click on the icon to create a new Python script file
2	Define the script name and the saving folder
3	Click on [Save]





Script name	Connector_Fx_Hst_OperatingPoint
Saving folder	~/MDO_EMOTOR/Flux/FluxProjects/ MAG_OPERATING_POINT

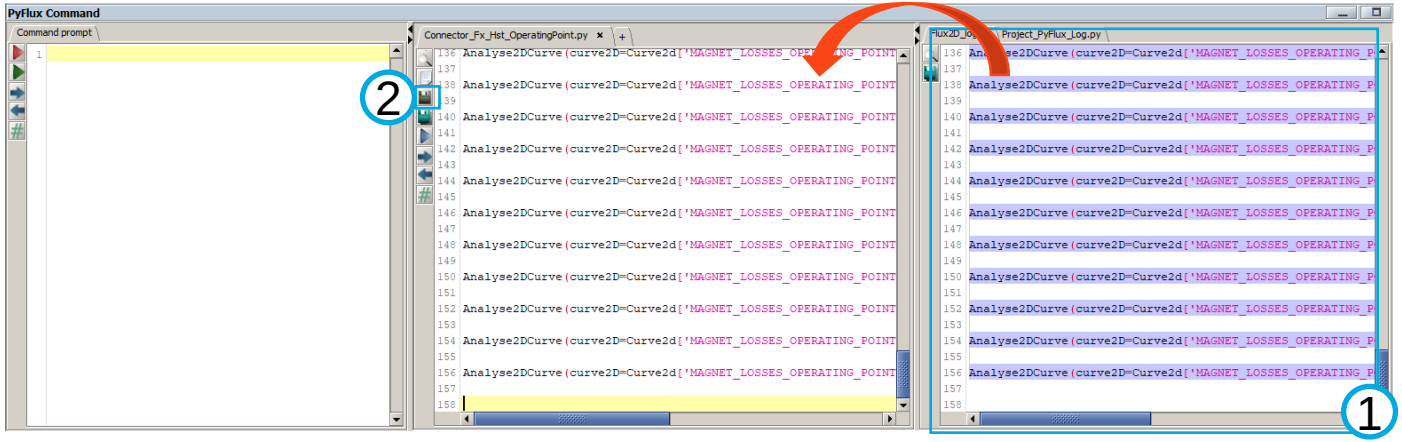
ALTAIR

GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Preparing post-processing Python script for HyperStudy connector

- Create backup of the log script

Step	Action
1	Copy all commandes from the Flux2D_log.py file into the created Python script
2	Click on the icon to save the script
3	Click on [Project] – [Exit]



ALTAIR

GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Preparing post-processing Python script for HyperStudy connector

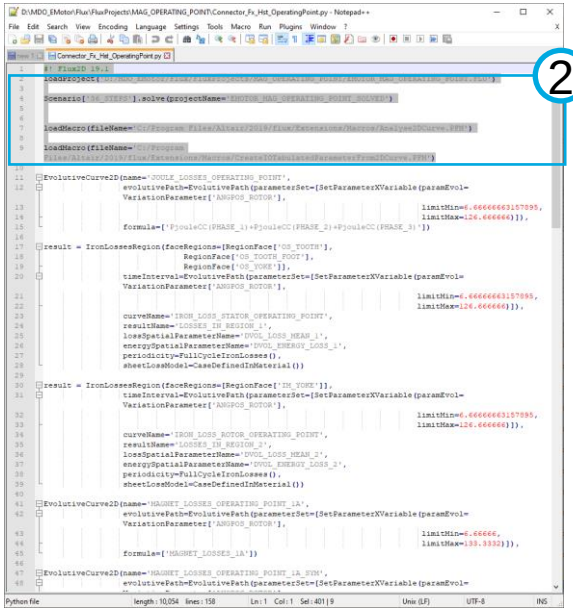
- Replace Flux macro import commands

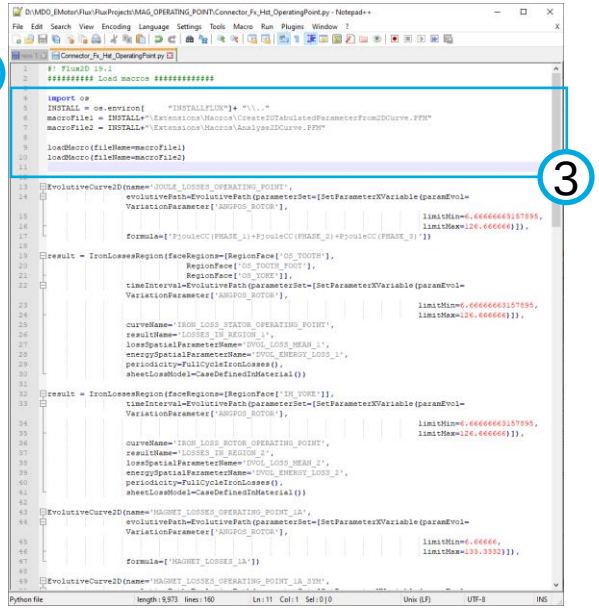
Step	Action
1	Open the Python script by Notepad editor
2	Delete the scripts before post-processing
3	Insert the following scripts before post-processing

########## Load macros ##############

```
import os
INSTALL = os.environ[     "INSTALLFLUX"]+ "\\.."
macroFile1 =
INSTALL+"\Extensions\Macros\CreateIOTabulatedParameterFrom2D
Curve.PFM"
macroFile2 = INSTALL+"\Extensions\Macros\Analyse2DCurve.PFM"

loadMacro(fileName=macroFile1)
loadMacro(fileName=macroFile2)
```





Attention: 1) the project will be solved automatically by HyperStudy;
2) the Flux macro location is depended on the Flux installation path. Therefore, these scripts should be replaced by a generic way to adapted to all the user.

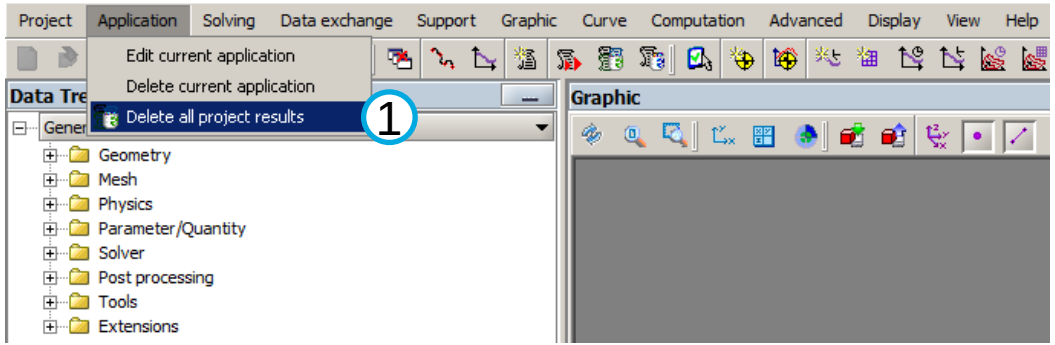
ALTAIR

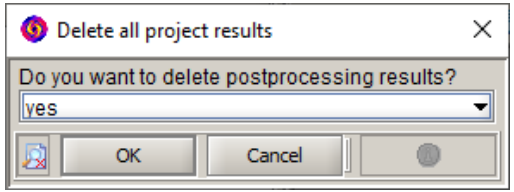
GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Generating HyperStudy connector

- Delete all project results

Step	Action
1	Click on [Application] – [Delete all project results]
2	Select [Yes], and click on [OK]





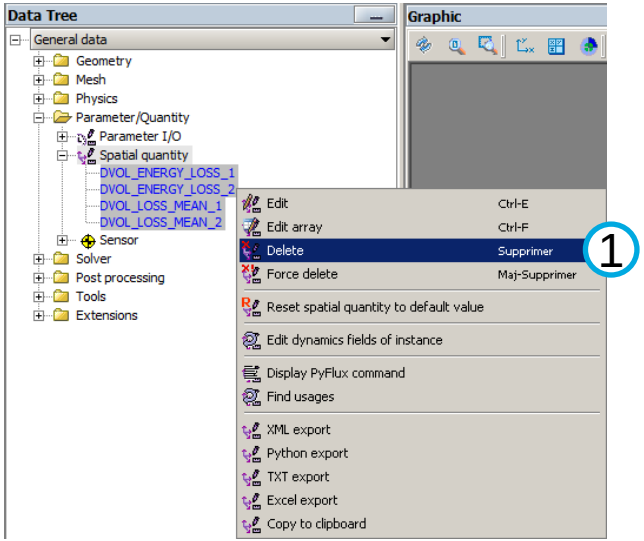
ALTAIR

GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Generating HyperStudy connector

- Delete spatial quantities not needed in connector

Step	Action
1	Select the 4 spatial quantities, right click and click on [Delete]



ALTAIR

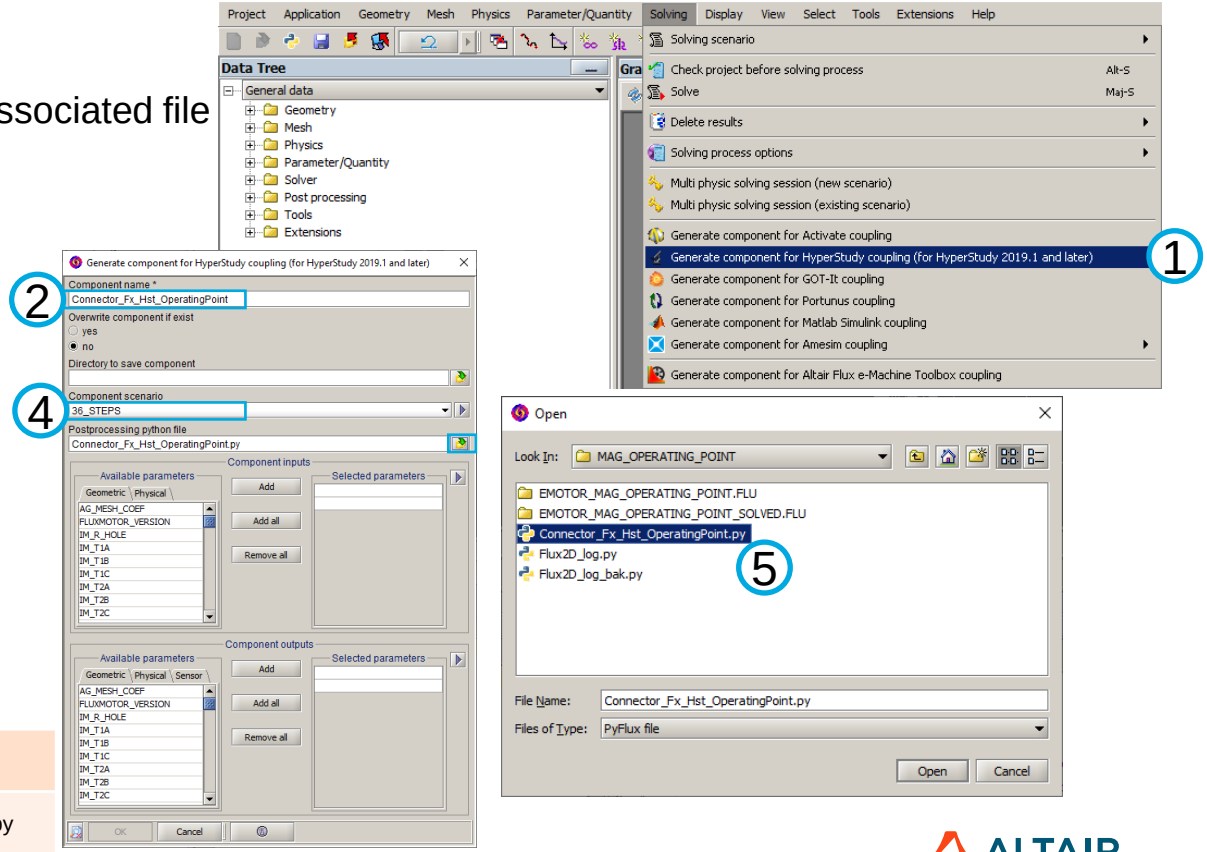
GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Generating HyperStudy connector

- Define connector name and associated file

Step	Action
1	Click on [Solving] – [Generate component for HyperStudy coupling (for HyperStudy 2019.1 and later)]
2	Define the component name as “Connector_Fx_Hst_OperatingPoint”
3	Directory by default
4	Define the component scenario “36_STEPS”
5	Define the Postprocessing python file as “Connector_Fx_Hst_OperatingPoint .py”

Connector name	Connector_Fx_Hst_OperatingPoint
Postprocessing Python file	Connector_Fx_Hst_OperatingPoint.py



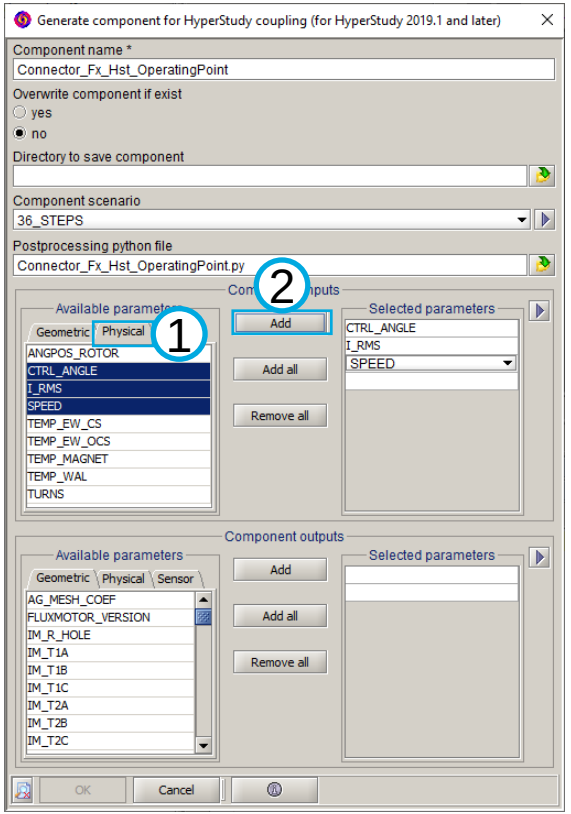
ALTAIR

GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Generating HyperStudy connector

- Define component inputs (physical)

Step	Action
1	Click on [Physical] tab, select the following three parameters: - CTRL_ANGLE - I_RMS - SPEED
2	Click on [Add]



ALTAIR

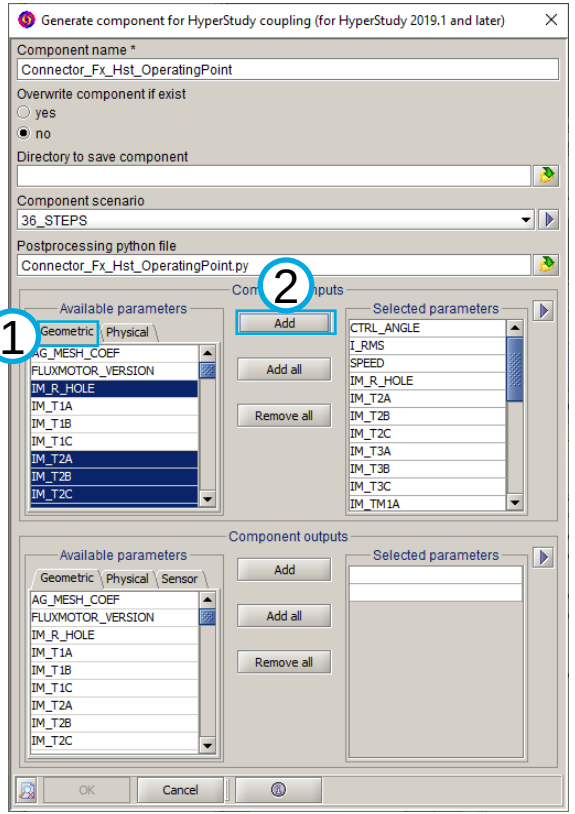
GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Generating HyperStudy connector

- Define component inputs (geometric)

Step	Action
1	Click on [Geometric] tab, select the following 22 parameters in the table
2	Click on [Add]

Input geometric variables	
IM_R_HOLE	IM_TM2B
IM_T2A	IM_TM2C
IM_T2B	IM_WA
IM_T2C	IM_WB
IM_T3A	IM_WC
IM_T3B	IM_WM1A
IM_T3C	IM_WM1B
IM_TM1A	IM_WM1C
IM_TM1B	IM_WM2A
IM_TM1C	IM_WM2B
IM_TM2A	IM_WM2C



ALTAIR

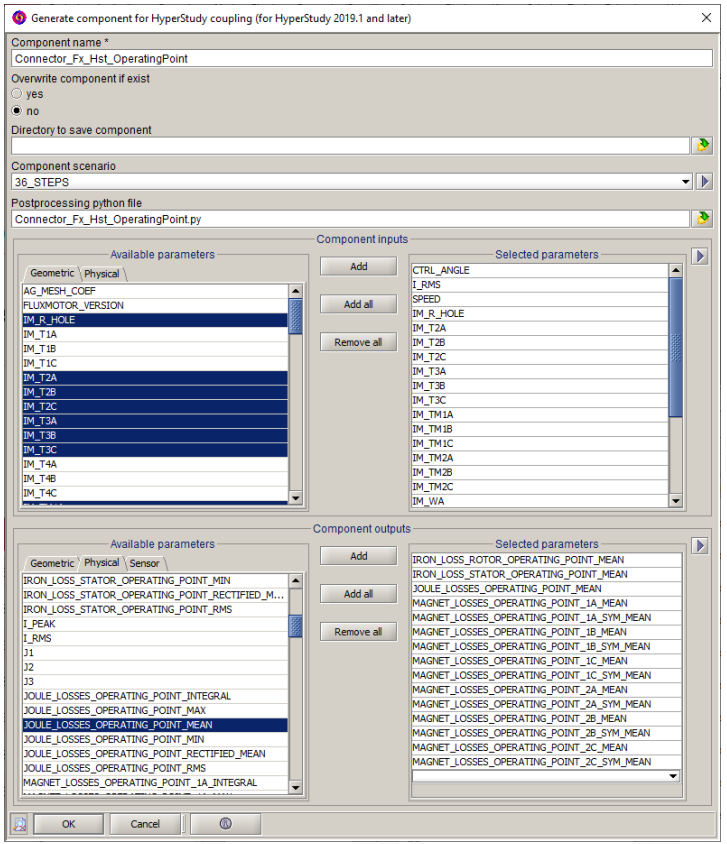
GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Generating HyperStudy connector

- Define component outputs

Step	Action
1	Select “Component outputs” 15 output variables: - 02 Iron losses - 01 Joule losses - 12 Magnet losses

Output variables	
IRON_LOSSES_ROTOR_SPECIFIC_SPEED_MEAN	MAGNET_LOSSES_OPERATING_POINT_1C_SYM
IRON_LOSSES_STATOR_SPECIFIC_SPEED_MEAN	MAGNET_LOSSES_OPERATING_POINT_2A
JOULE_LOSSES_SPECIFIC_SPEED_MEAN	MAGNET_LOSSES_OPERATING_POINT_2A_SYM
MAGNET_LOSSES_OPERATING_POINT_1A	MAGNET_LOSSES_OPERATING_POINT_2B
MAGNET_LOSSES_OPERATING_POINT_1A_SYM	MAGNET_LOSSES_OPERATING_POINT_2B_SYM
MAGNET_LOSSES_OPERATING_POINT_1B	MAGNET_LOSSES_OPERATING_POINT_2C
MAGNET_LOSSES_OPERATING_POINT_1B_SYM	MAGNET_LOSSES_OPERATING_POINT_2C_SYM
MAGNET_LOSSES_OPERATING_POINT_1C	



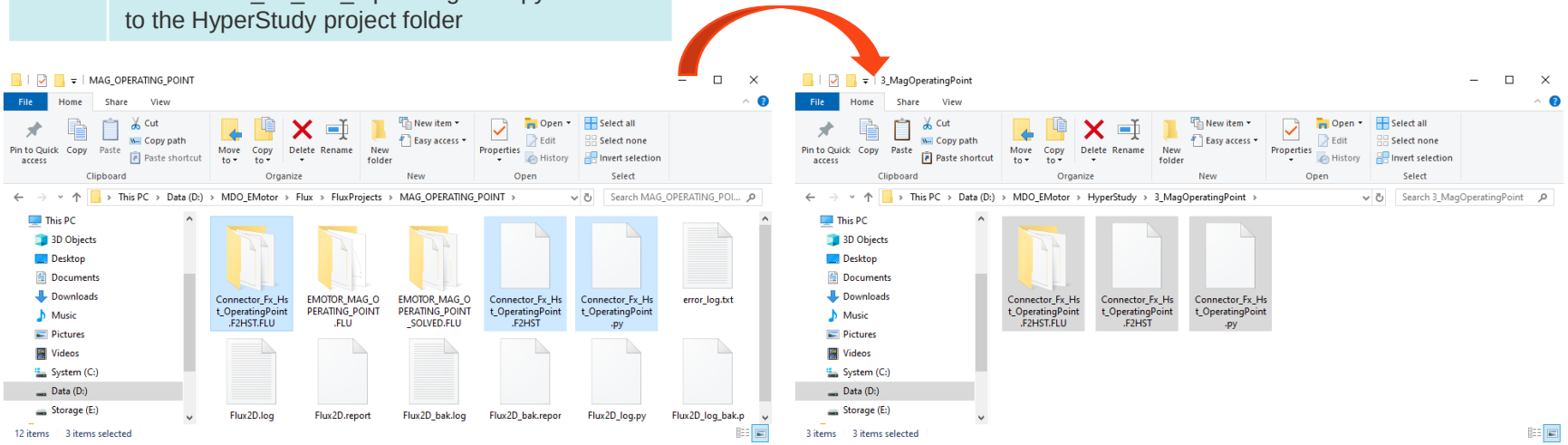
GENERATING HYPERSTUDY CONNECTOR: SPECIFIC OPERATING POINT

Generating HyperStudy connector

- Exchange files for the Flux / HyperStudy connector

Step	Action
1	Copy and paste the connector files: - Connector_Fx_Hst_OperatingPoint.F2HST - Connector_Fx_Hst_OperatingPoint.F2HST.FLU - Connector_Fx_Hst_OperatingPoint.py to the HyperStudy project folder

Path for saving the Flux magnetic analysis connector 2	~\MDO_EMotor\HyperStudy\2_OperatingPoint



THANK YOU

altair.com

#ONLYFORWARD

ALTAIR