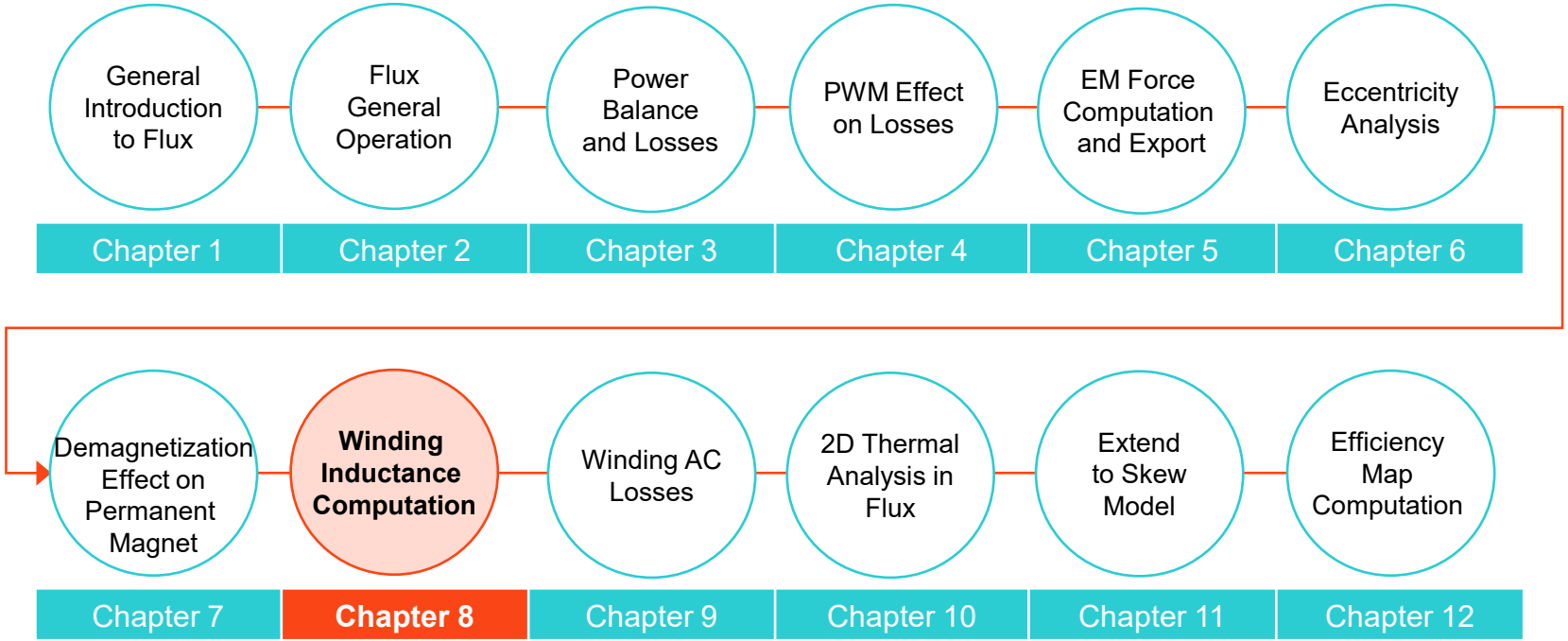
ALTAIR

ANALYSIS OF ELECTRIC MOTORS USING FLUX V2021.1

WINDING INDUCTANCE COMPUTATION

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Analysis of Electric Motors using Flux - Agenda

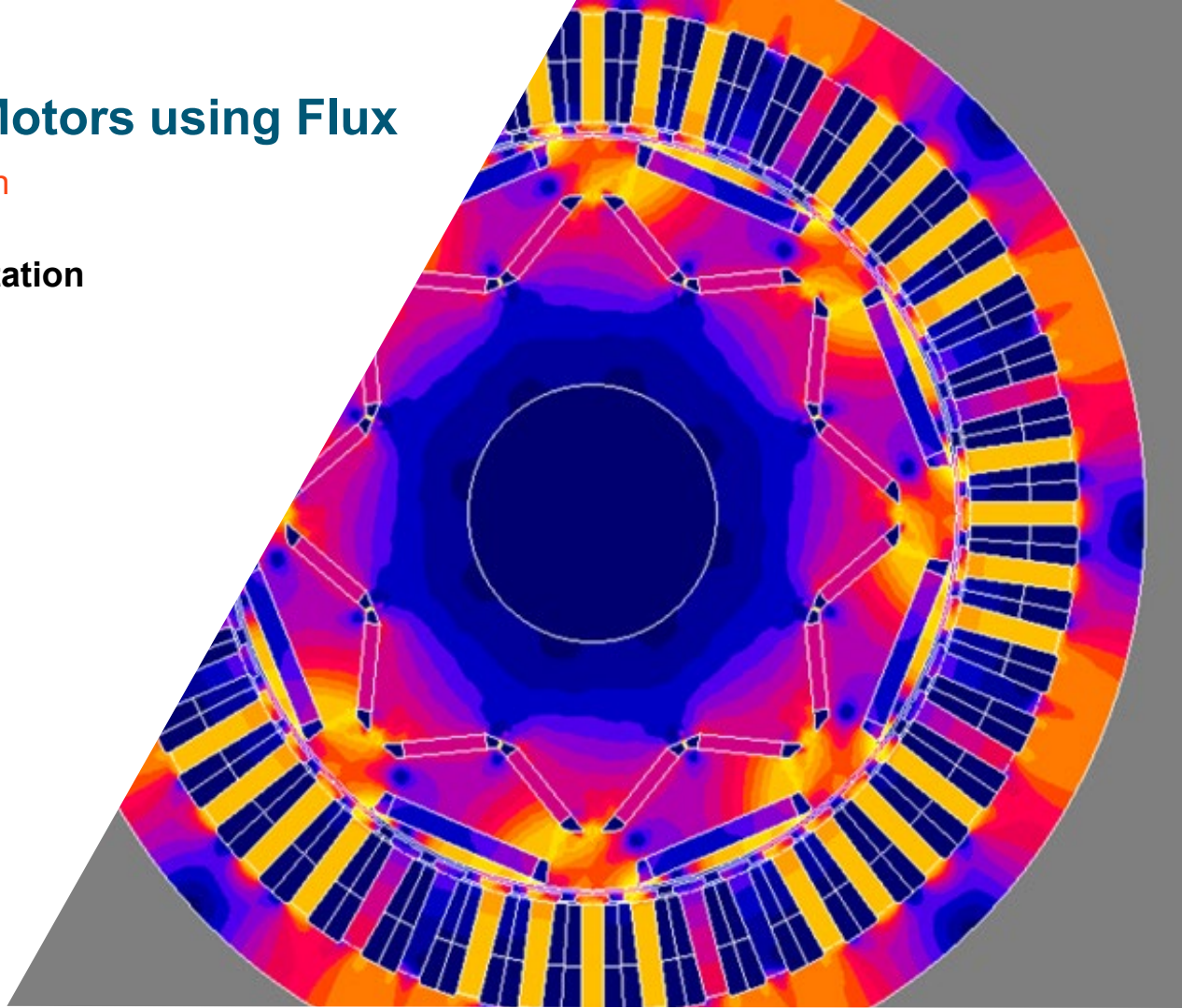


ALTAIR

Analysis of Electric Motors using Flux

Winding Inductance Computation

- **Winding Inductance Computation**
  - Overview
  - Example
- **Phase Inductance in Flux**
  - Exercise 8A - Phase Inductance
- **DQ Inductance in Flux**
  - Exercise 8B - DQ Inductance
- **Additional Information**
  - DQ Inductance Computation
  - Compute Inductance macro



© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

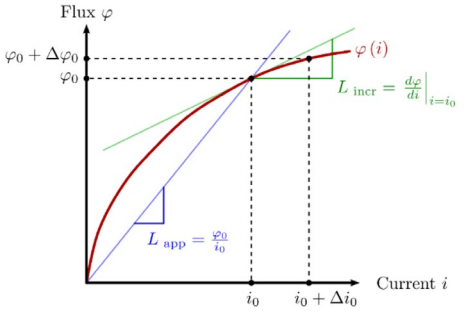
WINDING INDUCTANCE COMPUTATION

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Winding Inductance Computation

Overview

- **Inductance** (**L**) is characteristic of **coil** (inductor) measured in **Henry** (**H**).
  - It define how much energy the coil can store as magnetic field created by the current flow ($W_m = Li^2/2$)
  - It define the opposition to the change in current due to the induced **bemf** (electrical inertia) ($\mathrm{V} = L\, di/dt$)
  - It also define the relation between the magnetic flux linkage and the current flowing in the coil (can be defined as apparent or incremental inductance)



- The **computation of winding inductance** is important for modeling and control of electric machines:
  - Define the equivalent electric circuit of the motor.
  - Define a reduced order model of the machine on system simulation platform.
  - Adapt the control parameters for better dynamic performance.

Electrical equations:

$$V_d = R_s * I_d + \frac{d\psi_d}{dt} - \omega * \psi_q$$
$$V_q = R_s * I_q + \frac{d\psi_q}{dt} + \omega * \psi_d$$

Magnetic equations:

$$\psi_q = I_q * L_q$$
$$\psi_d = I_d * L_d + \psi_{PM}$$

Magnetic Torque:

$$\Gamma = \frac{3}{2} p * (\psi_d * I_q - \psi_q * I_d)$$

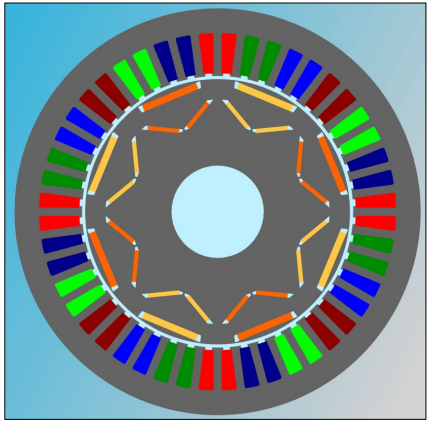
ALTAIR

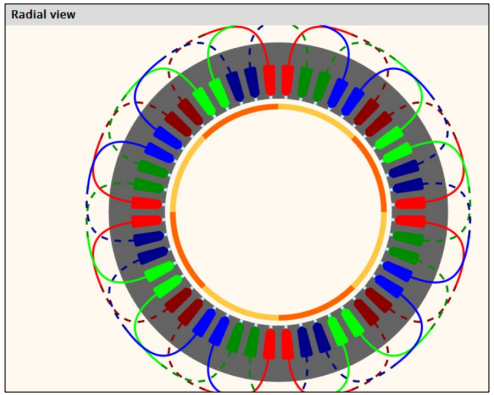
© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Winding Inductance Computation - Example

Problem Description

- **48 slots 8 poles Interior permanent magnet motor (IPM)**
  - Rated power of **88 kW @ 4000 rpm**, **375 DC** bus voltage and Max Speed of **15000 rpm**
  - Outer diameter **128 mm**, stack length is **150 mm** and air gap is **1 mm**
  - Lamination material: **M330_35A**
  - Magnet material: **NdFeB_1050_2800**
  - Three phase single layer winding with coil pitch of **5**
  - Number of coil per pole per phase is **1** and the number of turns per coil is **6** with two parallel paths





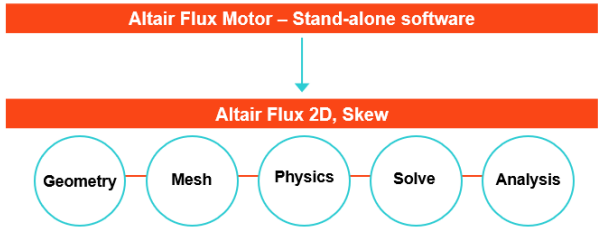
ALTAIR

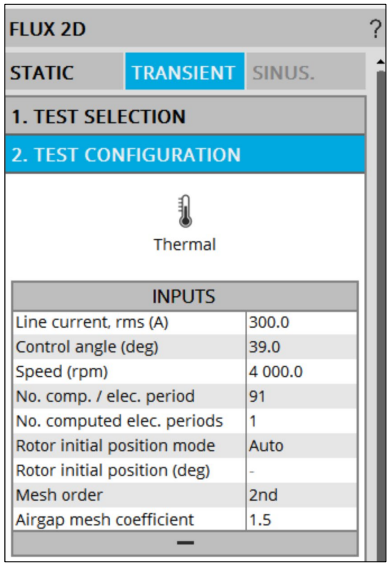
© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

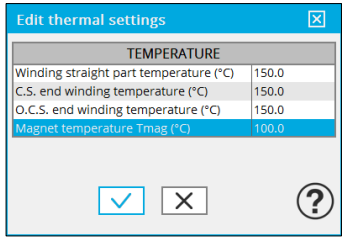
Winding Inductance Computation - Example

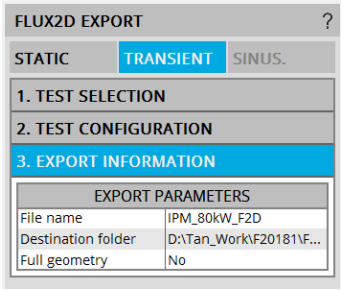
Model Build

- The prementioned motor is modeled and tested in **FluxMotor**.
- A **transient model** is exported from **FluxMotor** as a **Python script** under the following condition:
  - Line current of **300 Arms** and **39 deg** control angle
  - Constant speed of **4000 rpm**
  - Winding temp **150º C**, Magnet temp **100º C**
  - **91** steps and one electric cycle
  - Automatic determination of rotor initial position.
  - **Periodic model** (1/8th of the geometry)
- The generated script "**IPM_80KW_F2D**" can be run in **Flux** to create the base model (Ready to solve).









ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

PHASE INDUCTANCE IN FLUX

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Phase Inductance in Flux

Definitions

- When there is a change on the current flowing through a coil, there is induced emf generated which is called **Self Induction** and it defined by **L** (e $= -L\ di/dt$). If the emf is induced into an adjacent coil situated within the same magnetic field, it is called **Mutual Induction** and it defined by **M** (e $= -M\ di/dt$).

$$\begin{bmatrix}\varphi_a\\ \varphi_b\\ \varphi_c\end{bmatrix} = \begin{bmatrix}L_a & M_{ab} & M_{ac}\\ M_{ba} & L_b & M_{bc}\\ M_{ca} & M_{cb} & L_c\end{bmatrix}\begin{bmatrix}I_a\\ I_b\\ I_c\end{bmatrix} + \begin{bmatrix}\varphi_{0a}\\ \varphi_{0b}\\ \varphi_{0c}\end{bmatrix}.$$

  - **Flux** can compute both **self** and **mutual inductance** for different phases.
    - **Starting version 2020**, the user can use the integrated function **"Computation of inductance matrices"** in the post-processing, which can compute both **apparent** and **incremental Inductances**.
    - **Prior to version 2020**, the user can use the **macro** "**ComputeInductanceMatrix**", which allow to compute the apparent inductance.

Note: Please check **Additional information** section for more information on how this macro works.

ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

EXERCISE 8A - PHASE INDUCTANCE

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

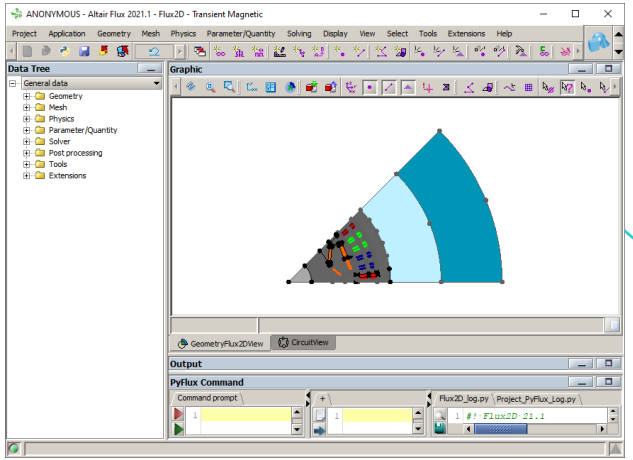
Exercise 8A - Phase Inductance

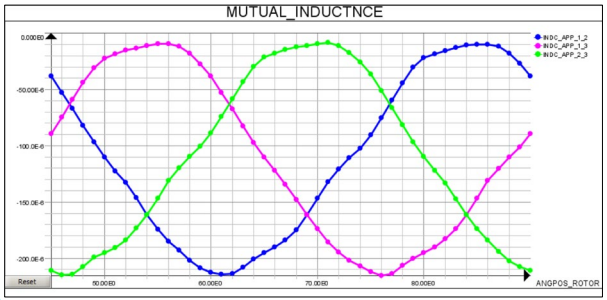
File Name and Location

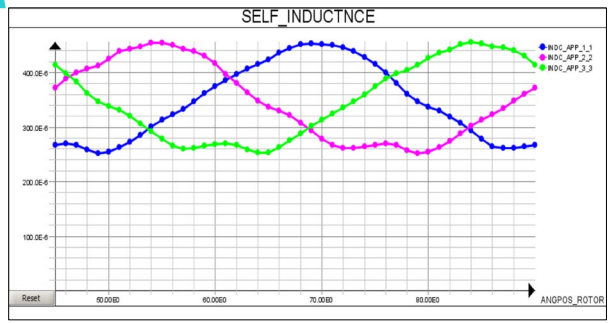
…\STUDENT-EXERCISE\ IPM_80KW_F2D.py

Exercise Goals

1. Building base model
2. Model Setup
3. Solving
4. Inductance Computation







ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Exercise 8A - Phase Inductance

Step 1: Building base model

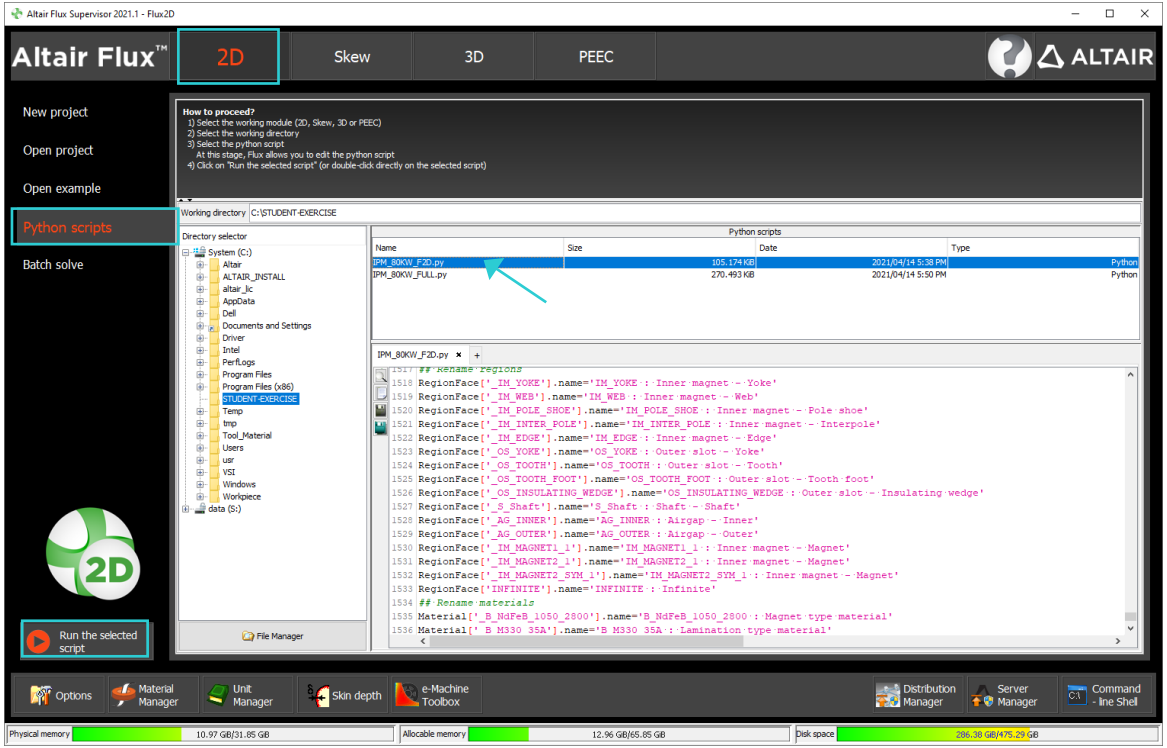
a) Open **Flux Supervisor**.

b) Select **Flux 2D** in topology.

c) Select **Python script** in actions.

d) Select the script from the folder:
`...\STUDENT-EXERCISE\IPM_80KW_F2D.py`

e) Run the selected script.



ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Exercise 8A - Phase Inductance

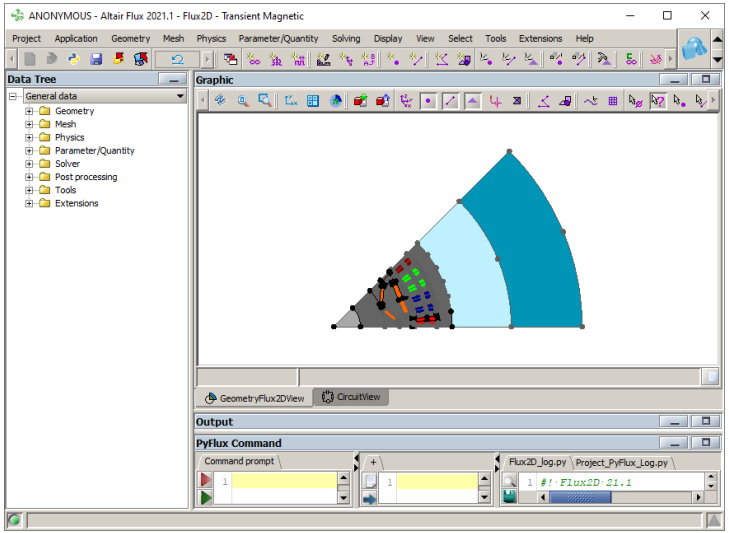
Step 2: Model Setup

a) Once the script is done, it will create **Flux 2D transient project**.

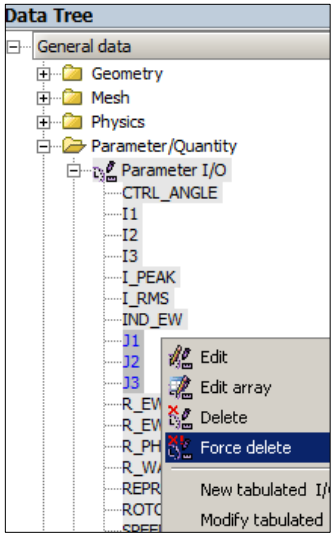
b) Circuit dependent **I/O Parameters** are **not** allowed when using computation of inductance function.

c) Force delete any **I/O Parameters** based on the electric circuit (**J1**,**J2**,**J3**,**V1**,**V2**,**V3**).

a



c



ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

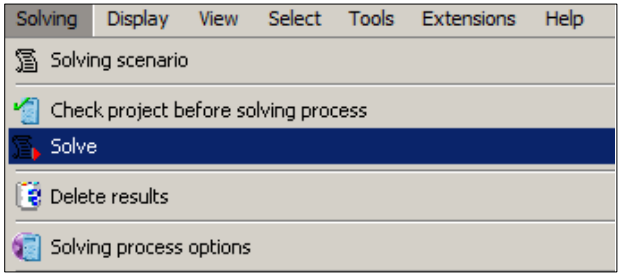
Exercise 8A - Phase Inductance

Step 3: Solving

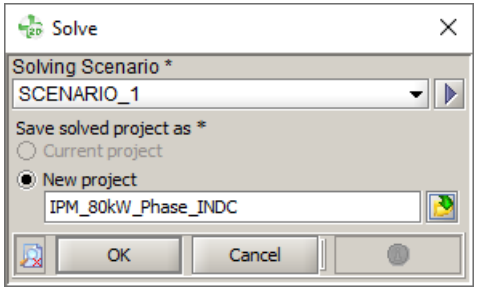
a) Solve the project “**SCENARIO_1**”.

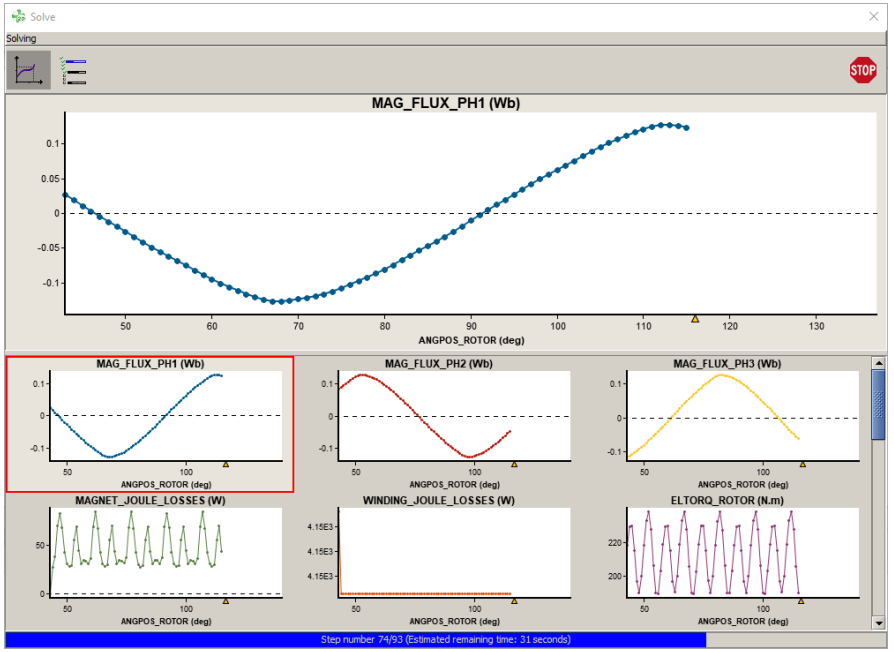
b) Provide a **project name** to the solved project.

a



b





© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Exercise 8A - Phase Inductance

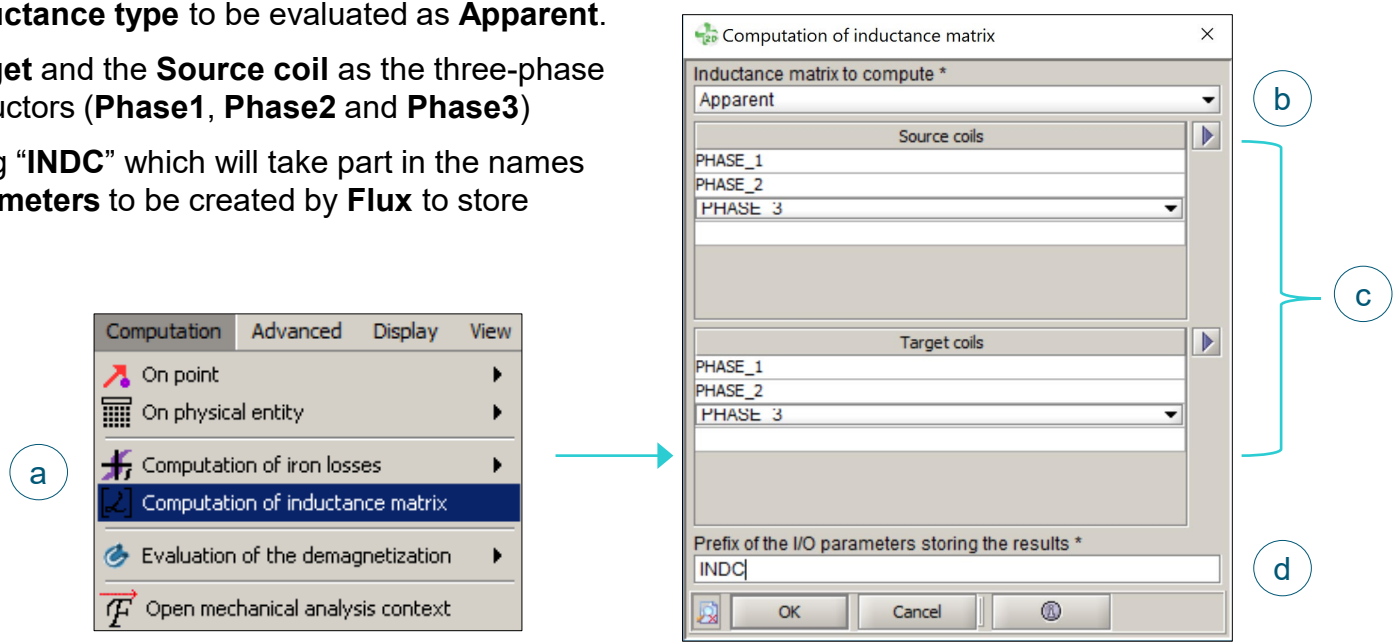
Step 4: Inductance Computation

a) In the post-processing context, under the **Computation** menu select **Computation of Inductance matrix**.

b) Select the **Inductance type** to be evaluated as **Apparent**.

c) Select the **Target** and the **Source coil** as the three-phase stranded conductors (**Phase1**, **Phase2** and **Phase3**)

d) Provide a string "**INDC**" which will take part in the names of the **I/O Parameters** to be created by **Flux** to store the results.



© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

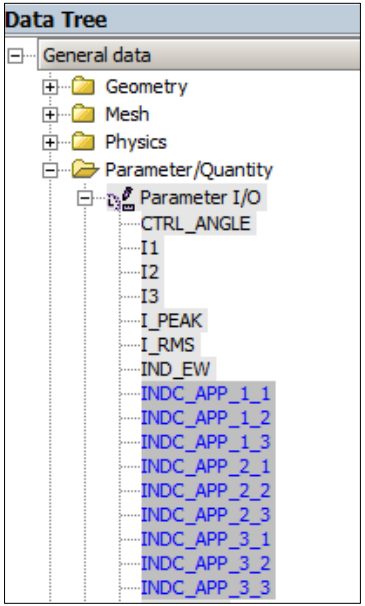
Exercise 8A - Phase Inductance

Step 4: Inductance Computation

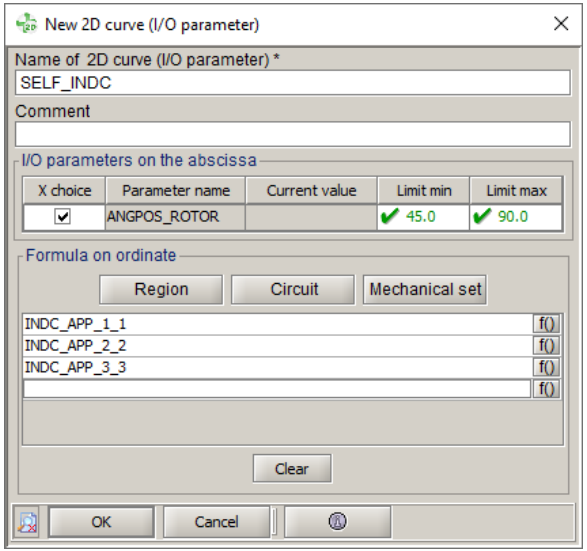
e) By the end of the computation, the results can be found in the list of **I/O Parameters**.

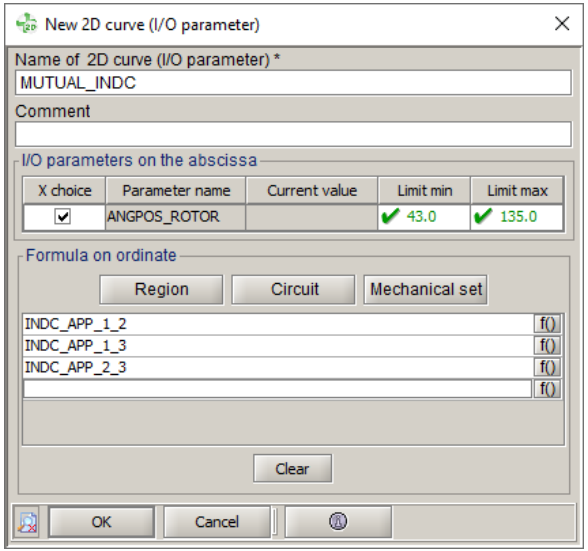
f) Those parameters can be used to plot the **auto(self)** or **mutual inductance** of the phases.

e



f





ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Exercise 8A - Phase Inductance

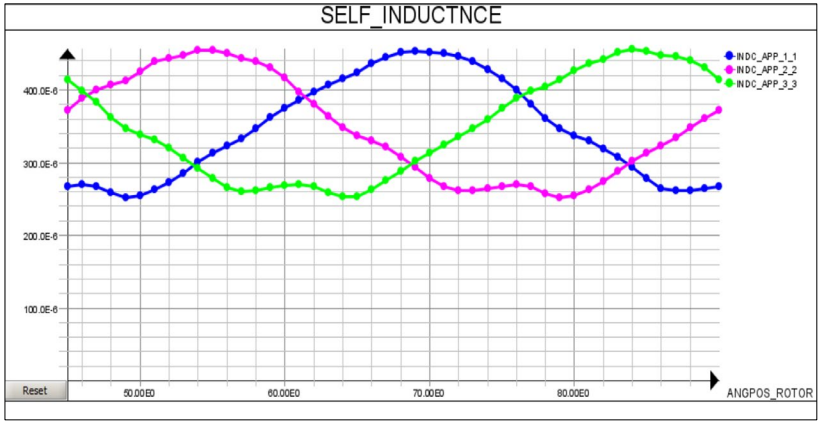
Step 4: Inductance Computation

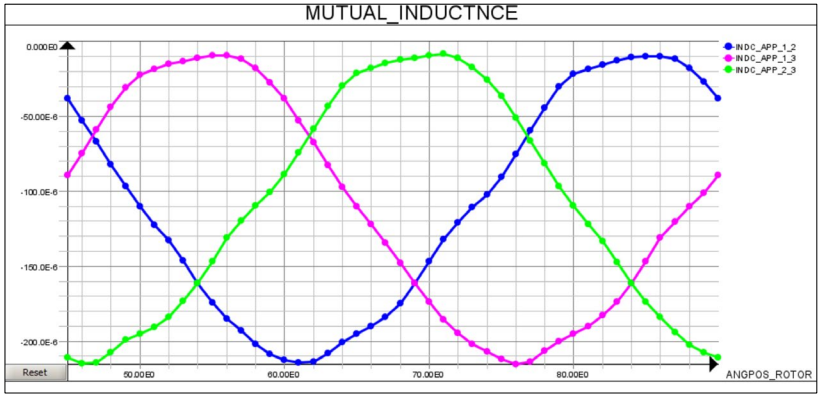
g) Results.

You can find the **solved project** at the following location:

…\STUDENT-EXERCISE-SOLVED\CH8\Inductance_Comp

folder





ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

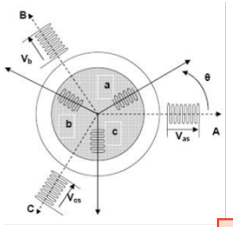
DQ INDUCTANCE IN FLUX

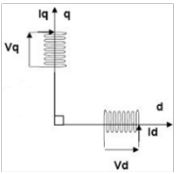
© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

DQ Inductance in Flux

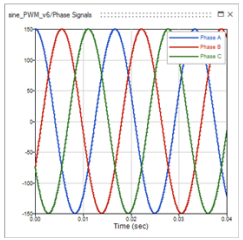
Principle

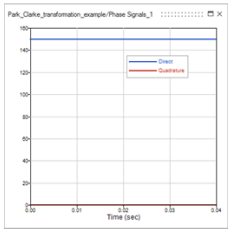
- **Convert 3 phase rotating reference frame** to **2 phase rotor-fixed reference frame** (direct-quadrature).





Park/Clarke Transformations





- It easier to model and control the motor on **DQ** reference frame (**DC quantities**) than in **ABC** reference frame (**Sinusoidal quantities**).

Electrical equations:

$$V_d = R_s * I_d + \frac{d\psi_d}{dt} - \omega * \psi_q$$

$$V_q = R_s * I_q + \frac{d\psi_q}{dt} + \omega * \psi_d$$

Magnetic equations:

$$\psi_q = I_q * L_q$$

$$\psi_d = I_d * L_d + \psi_{PM}$$

Magnetic Torque:

$$\Gamma = \frac{3}{2} p * (\psi_d * I_q - \psi_q * I_d)$$

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R_s \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 & -\omega L_q \\ \omega L_d & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \psi_m \end{bmatrix}$$

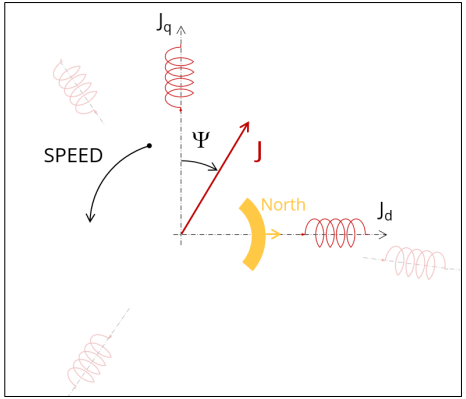
ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

DQ Inductance in Flux

Procedure

- Determine the **rotor angle** which **align** it with **d-axis** and **q-axis** positions.
- Define or convert the application to **magnetostatics application**.
- Define the **coils** and their **excitation**.
- Align the **rotor** position to **d-axis** and solve for different current values (only **Id**). Repeat the process for **q-axis** position (only **Iq** current)
- For each case compute the **d-axis** (**q-axis**) **Flux linkage** and differentiate against current for incremental **Ld** (**Lq**).
- For apparent **Lq divide** the **q-axis Flux linkage** over the **current**, while for **Ld** you need to **subtract** the **magnet Flux linkage** from the **d-axis Flux linkage** first.



Magnetic equations:

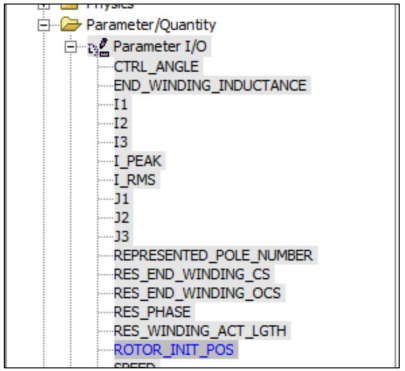
$$\psi_q = I_q * L_q$$

$$\psi_d = I_d * L_d + \psi_{PM}$$

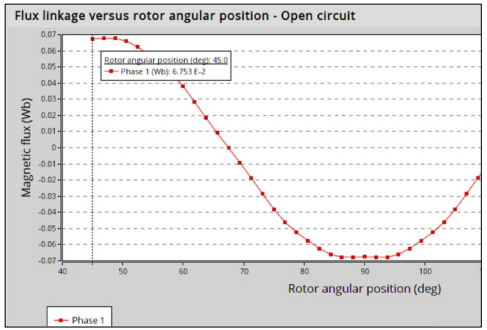
© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

DQ Inductance in Flux

Find the Initial Rotor Angle



- **There are different method to find the initial rotor angle to align with d-axis:**
  - If the model is created from **FluxMotor script**. This position will be directly defined in the model as **I/O parameter** "**ROTOR_INIT_POS**".
  - Use the macro "**Find_rotor_angle_2D**" available under **Flux 2D post-processing** folder. This macro will create a new parameter "**Pos_INI**" with the required value.
  - Perform an **open circuit simulation** in **Flux** and find the position from the **Flux linkage** or bemf.



- **The q-axis position can be found directly from the d-axis position.**
  - **d-axis** position is **45 deg**.
  - $\theta_q = \theta_d - \frac{180}{pole\ number}$ **= 22.5 deg**

ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

EXERCISE 8B - DQ INDUCTANCE

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

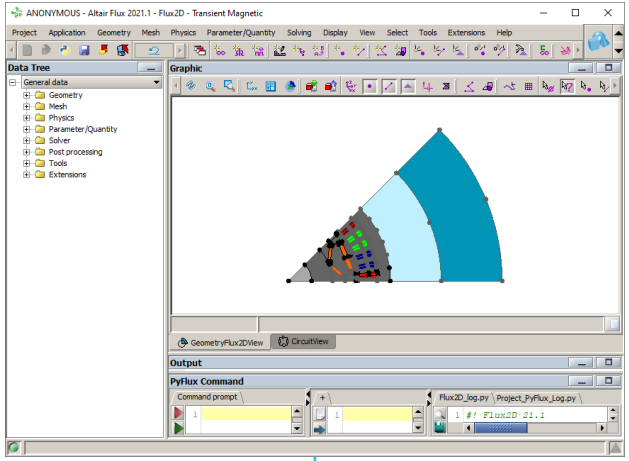
Exercise 8B - DQ Inductance

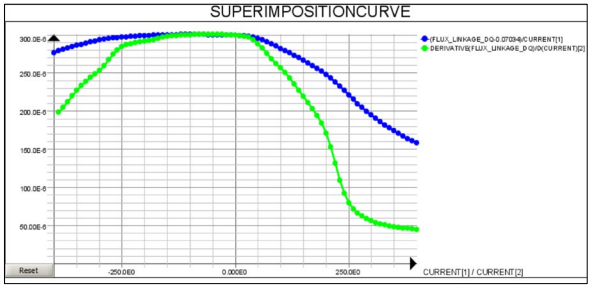
File Name and Location

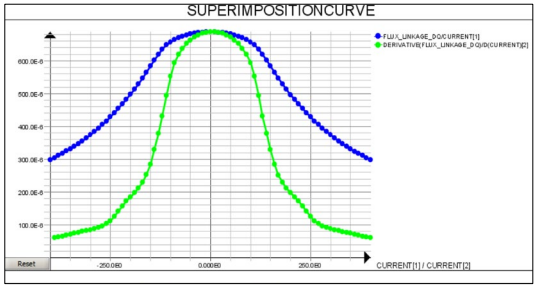
…\STUDENT-EXERCISE\ IPM_80KW_F2D.py

Exercise Goals

1. Magnetostatic Application
2. Coils and Excitation
3. Post-processing
4. Solving Scenarios
5. Ld Computation
6. Ld Results
7. Lq Results







© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Exercise 8B - DQ Inductance

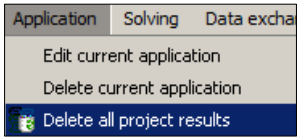
Step 1: Magnetostatic Application

a) Start from **transient project**, **Delete the project results** (if the project is solved).

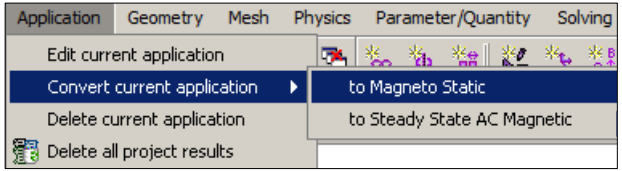
b) Convert current application to **Magneto Static**.

c) Define the **magnets region** as **Magnetic non conducting region**.

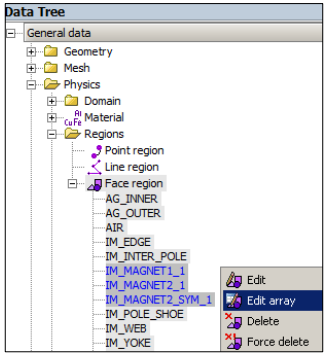
a

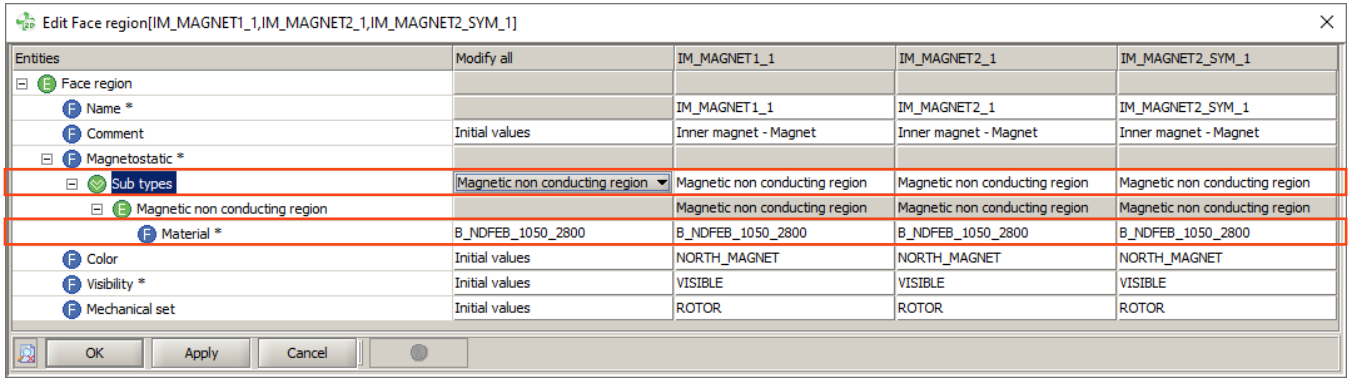


b



c





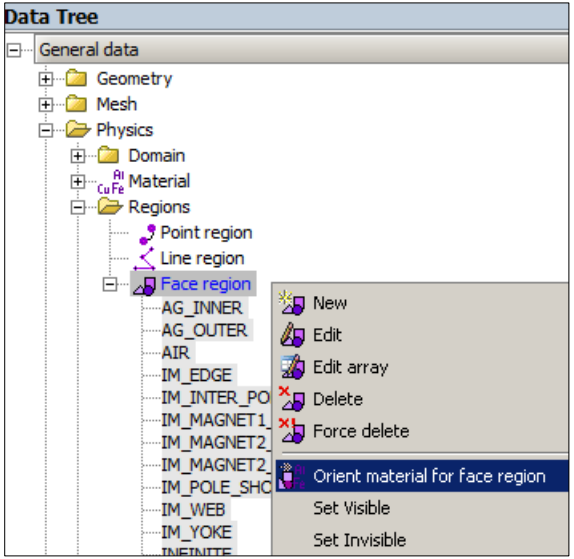
ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

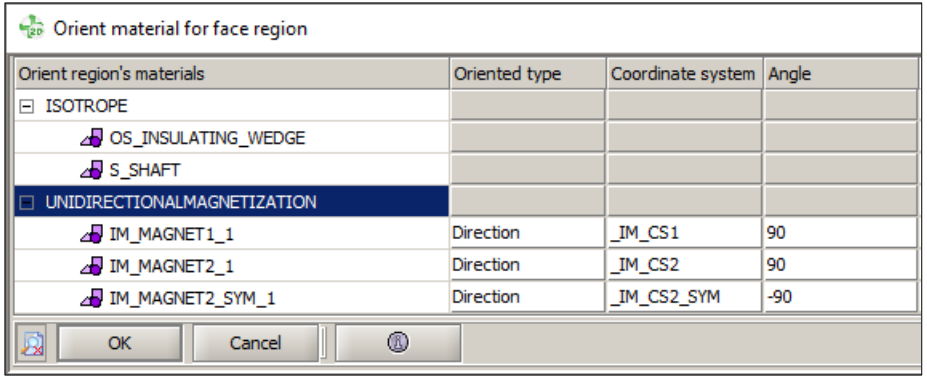
Exercise 8B - DQ Inductance

Step 1: Magnetostatic Application

d) Redefine the **right orientation** for the magnet.



d



© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

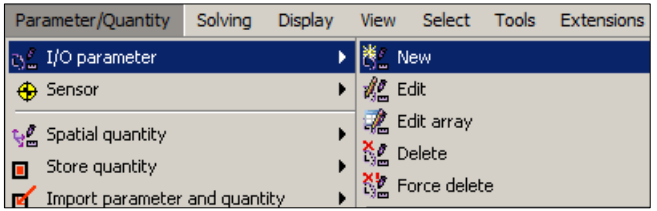
Exercise 8B - DQ Inductance

Step 2: Coils and Excitation

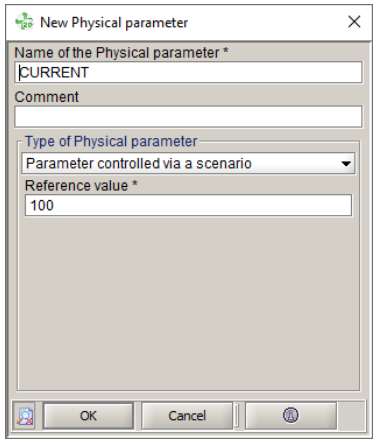
a) Define a **New Physical parameter** to define current variation.

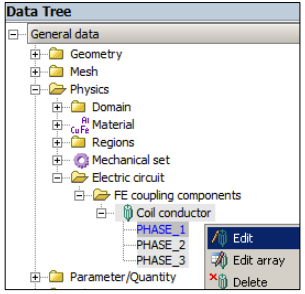
b) Modify **Coil conductors** (**PHASE_1**, **PHASE_2**, and **PHASE_3**) with the values shown below.

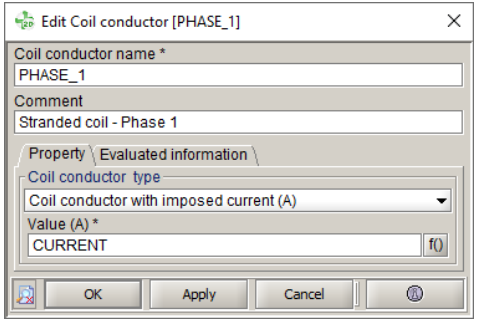
a



b







Coil_Name	Value d_case
PHASE_1	CURRENT
PHASE_2	- CURRENT/2
PHASE_3	- CURRENT/2

Hint: Once the rotor position is aligned with D axis position, using **Park transformation**, the values assigned to the conductors will correspond to **Ld = Current** and **Lq = 0**. The same is true when the rotor position is aligned with **q-axis**.

ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

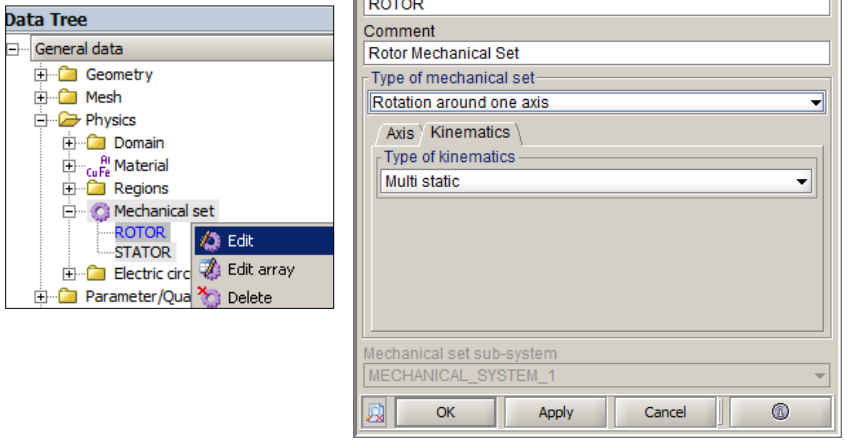
Exercise 8B - DQ Inductance

Step 3: Post-processing

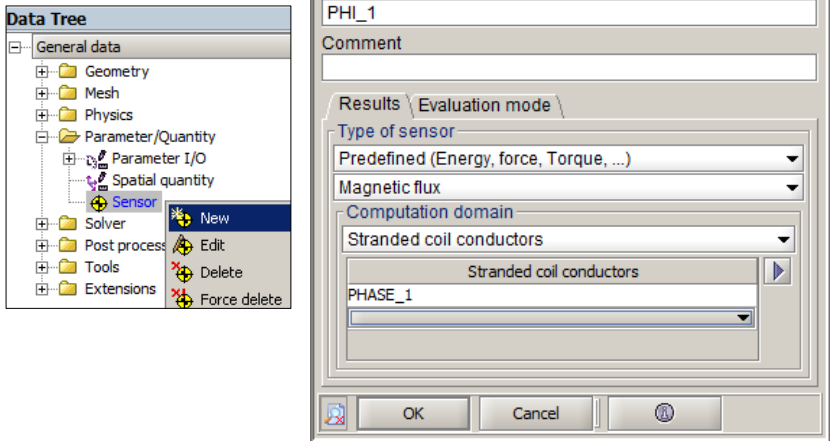
a) Open and validate (click on **OK**) the new setting for "**ROTOR**" **Mechanical set** (**Multi static**).

b) Create **Sensors** to compute the **Flux linkage** on each phase (**PHI_1**, **PHI_2** , and **PHI_3**)

a



b



ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

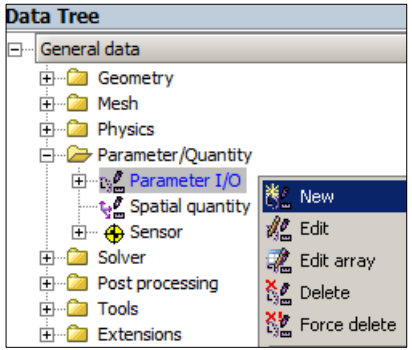
Exercise 8B - DQ Inductance

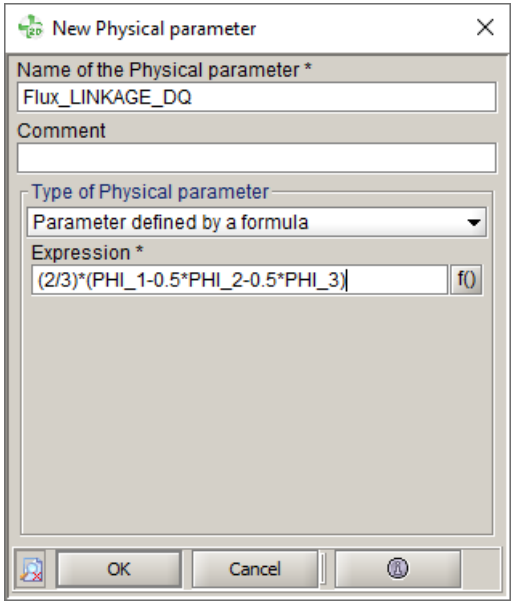
Step 3: Post-processing

c) Define an **I/O Parameter** with the equation below with **Theta** equal to zero. This equations is for **d-axis**, but it hold true for **q-axis** since the initial position will be changed as shown in the next slides.

$$\Phi_d = \frac{2}{3} \times \left(\Phi 1 \times Cos(\theta) + \Phi 2 \times Cos\left(\theta - \frac{2\pi}{3}\right) + \Phi 3 \times Cos\left(\theta - \frac{4\pi}{3}\right) \right)$$

c





ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

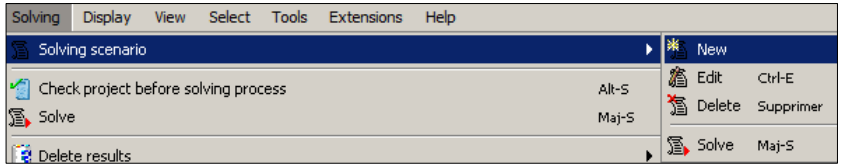
Exercise 8B - DQ Inductance

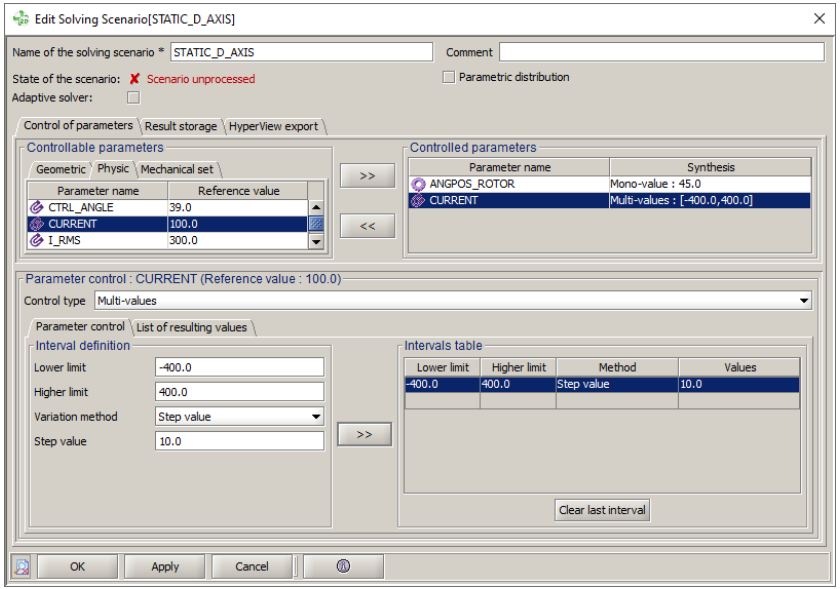
Step 4: Solving Scenarios

a) Define a **New Solving Scenario** “**STATIC_D_AXIS**”:

- Rotor position fixed at **45 deg**.
- Current varies from **- 400** to **400** by step of **10 A**

a





© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

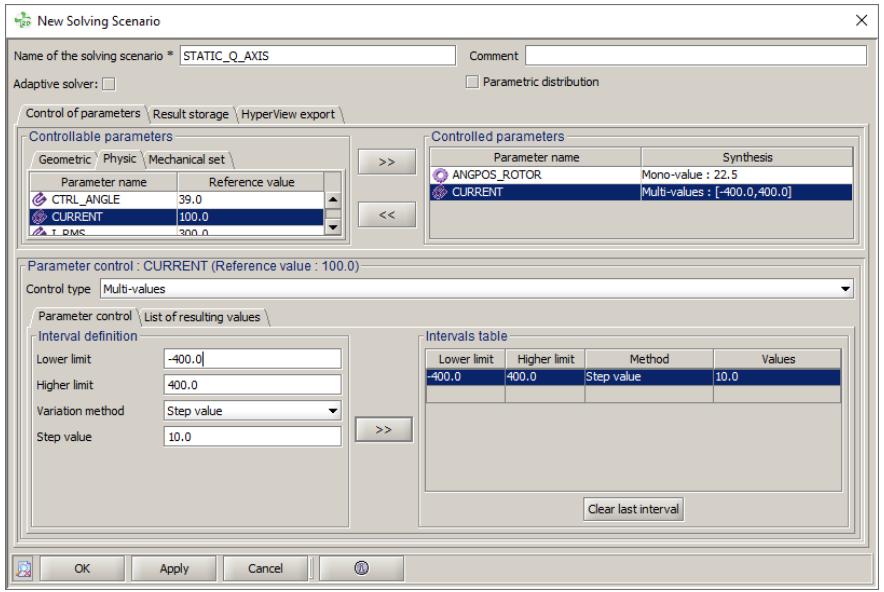
Exercise 8B - DQ Inductance

Step 4: Solving Scenarios

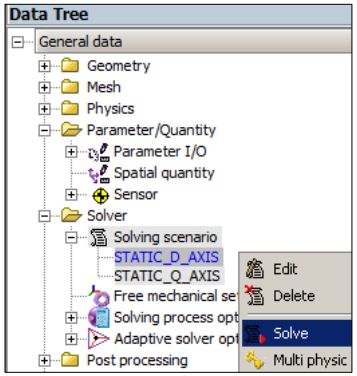
b) Define a similar scenario “**STATIC_Q_AXIS**” with the rotor position fixed at **22.5 deg**.

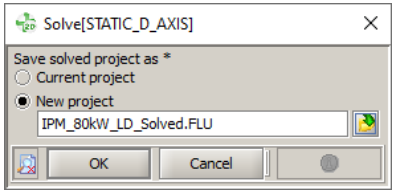
c) Solve the scenarios, you can save **each solved scenario** in a **New project**.

b



c





ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Exercise 8B - DQ Inductance

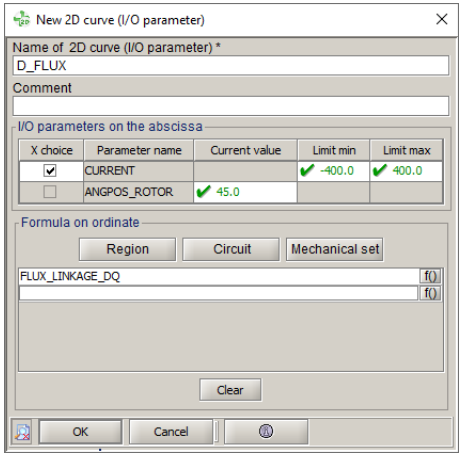
Step 5: Ld Computation

a) Plot the parameter "**FLUX_LINKAGE_DQ**" versus **Currents**.

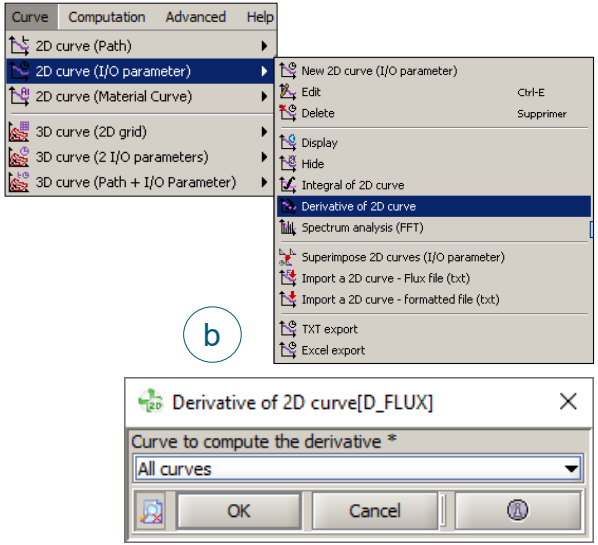
b) Differentiate the plots to find the incremental inductance.

c) Deduct the **magnet Flux linkage** (**0.07034 Wb**) from the **Flux linkage** results and divide the results by the current to find the apparent inductance.

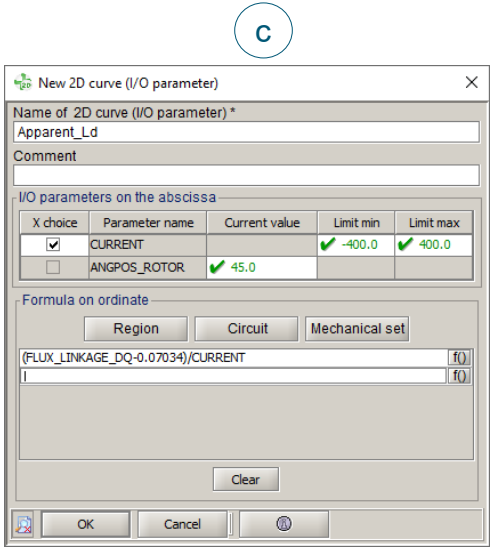
a



b



c



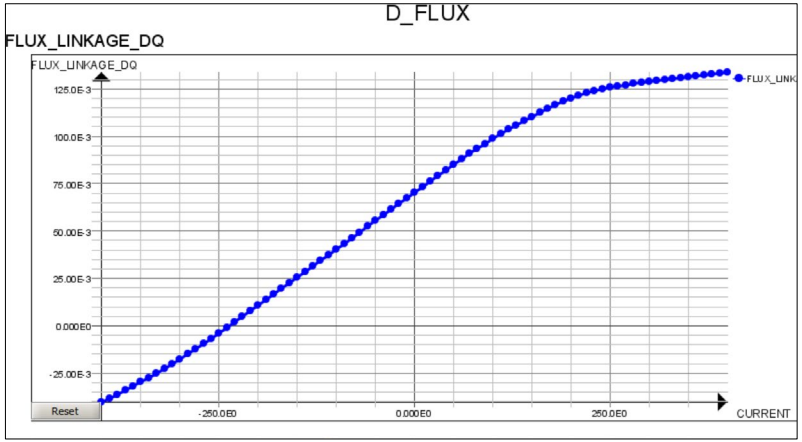
ALTAIR

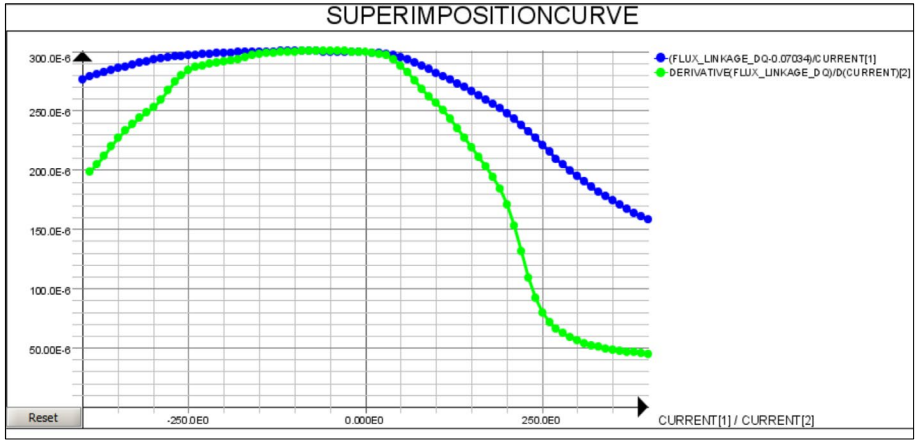
© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Exercise 8B - DQ Inductance

Step 6: Ld Results

a) Superimpose the plots for better comparison.





Hint: **Curve > 2D curve (I/O parameter) > Superimpose 2D curves (I/O parameter)**

ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

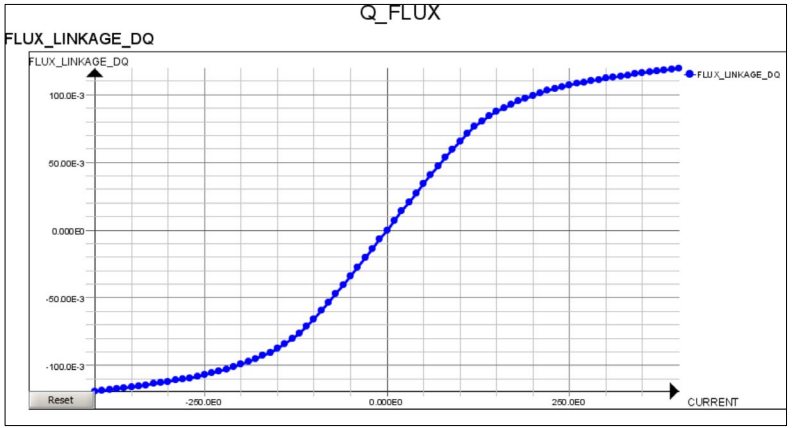
Exercise 8B - DQ Inductance

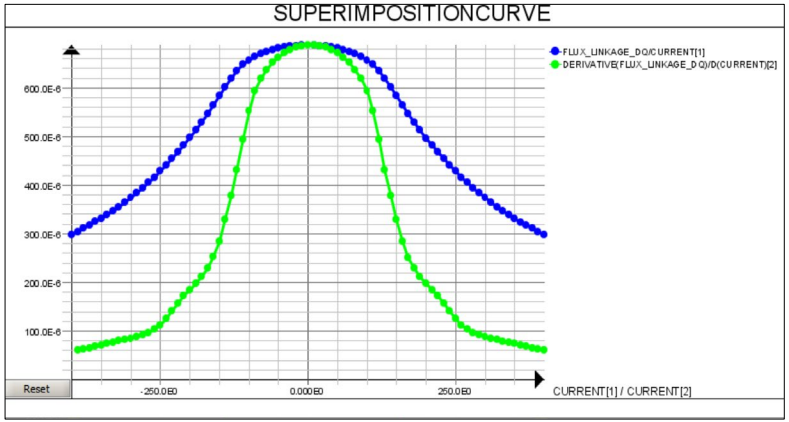
Step 7: Lq Results

a) Similarly, the computation can be done for **Lq**, the difference is that you don't need to subtract the **magnet Flux linkage** for the apparent **Lq** computation.

You can find the **solved projects** at the following location:
…\STUDENT-EXERCISE-SOLVED\CH8\Inductance_Comp

folder





© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

ADDITIONAL INFORMATION

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

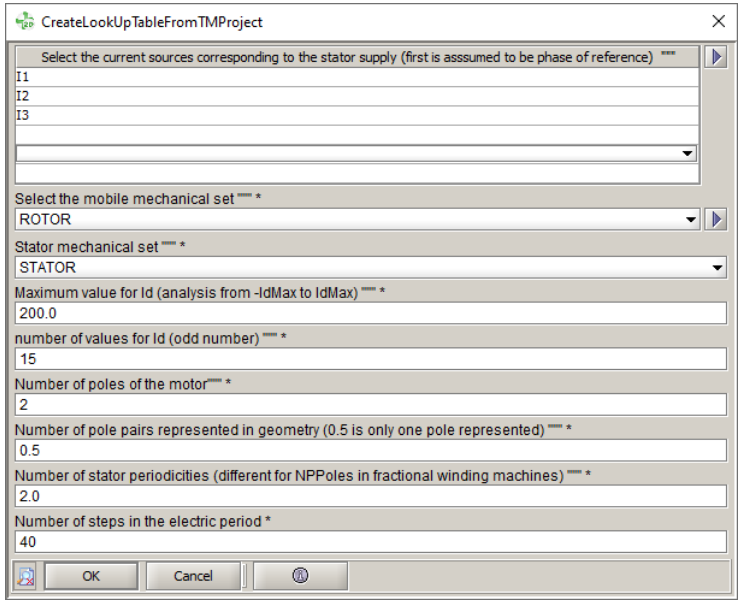
Additional Information

DQ Inductance Computation

- Another suggested workflow is to use the macro “ **CreateCreateLookUpTableFromTMProject”**
  - It will help by creating magnetostatics project and two I/O parameters to pilot **Id** and **Iq** current.

    Note: Refer to **Brushless IPM Motor** (embedded magnet) technical note Study case 5.

  - This macro could be useful as well to compute **Ld** & **Lq** as maps versus **Id** & **Iq**.



	A	B	C	D	E	F	G	H	I	J	K
1	ID_PILOT ; IQ_PILOT ; ANGPOS_ROTOR ; MAG_FLUX_PH1 ; MAG_FLUX_PH2 ; MAG_FLUX_PH3 ; ELTORQ_ROTOR ;										
2	-0.3000000000000000E+03 ;	-0.3000000000000000E+03 ;	0.0000000000000000E+00 ;	-0.1059142581252494E+00 ;							
3	-0.2500000000000000E+03 ;	-0.3000000000000000E+03 ;	0.0000000000000000E+00 ;	-0.9999205514595558E-01 ;							

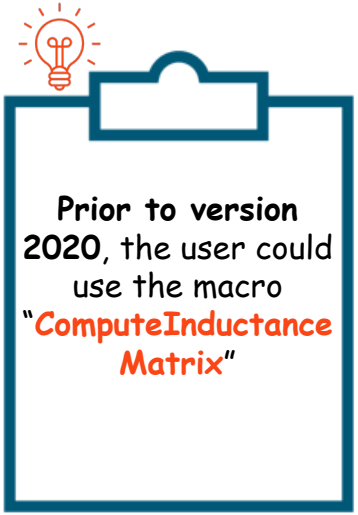
ALTAIR

© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Additional Information

Compute Inductance macro

- **Working principle**:
  - Freezing of the permeability of the magnetic circuit (rotor and stator)
  - Current modification in the sources.
    Example: For a **3 phases** circuit, **1 A** is applied in the first source, **0 A** in the others
  - Creation of sensors to compute the fluxes of the coils
  - Computation of the fluxes in the coils with the permeability freezed
  - Computation of the self and mutual inductances (**L_1_1**; **L_1_2**; **L_1_3**)
  - Modification of the circuit to impose **1 A** in the second source and **0 A** in the others
  - Computation of the fluxes again
  - The workflow is repeated based on number of coils selected.

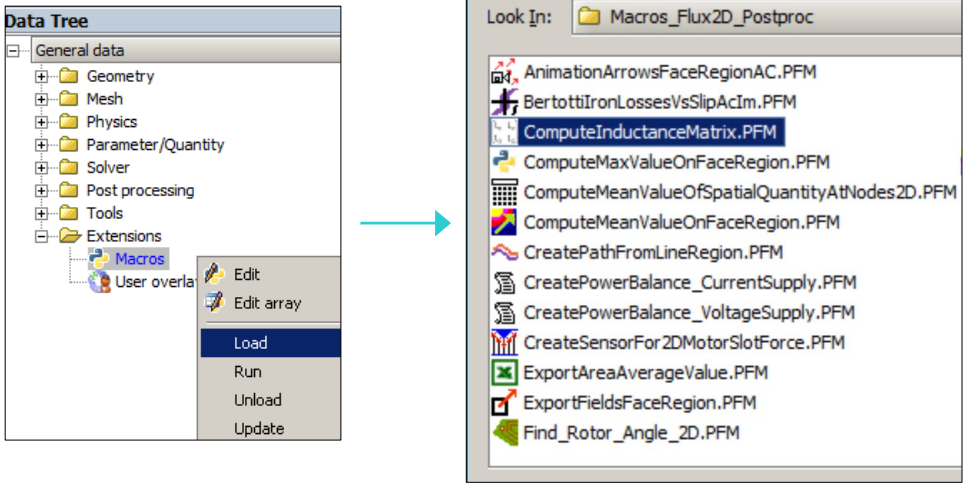


© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Additional Information

Compute Inductance macro

- In the post-processing context, load the macro “**ComputeInductanceMatrix**” located at the folder “**Macros_Flux2D_Postproc**”.



Note: Optionally, **run** the script “**IPM_80KW_F2D.py**” located at `STUDENT_EXCERCISE` folder and **solve** the **2D Transient project** before loading the macro.

ALTAIR

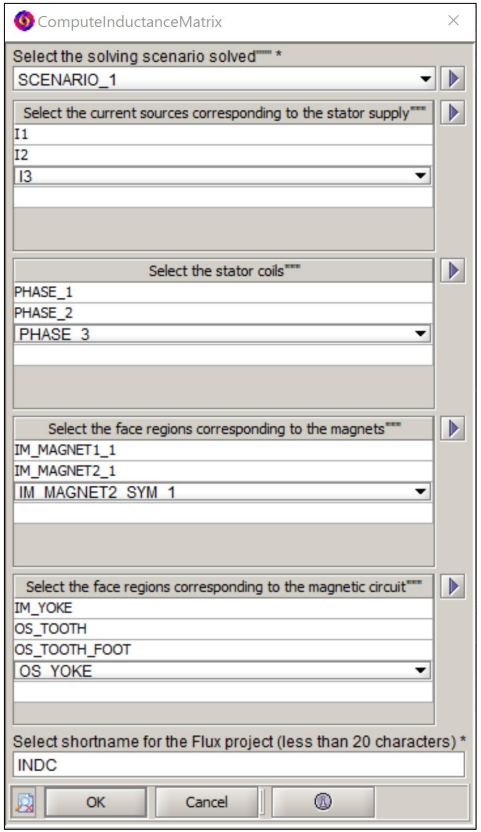
© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Additional Information

Compute Inductance macro

- **Run** the macro and set the following **parameters**:
  - A **solving scenario** with results
  - The **current sources** corresponding to the stator supply
  - The **stator coils**
  - The **face regions** corresponding to the **magnets**
  - The **face regions** corresponding to the **magnetic circuit**
  - A **name** for the resulting project

Note: It's **mandatory** to have the **same number of current sources and coils**.
The current sources and the coils should have the **same order**.

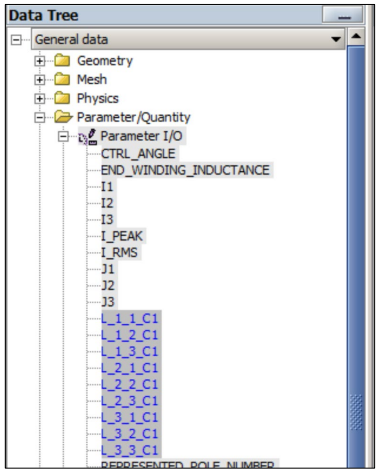


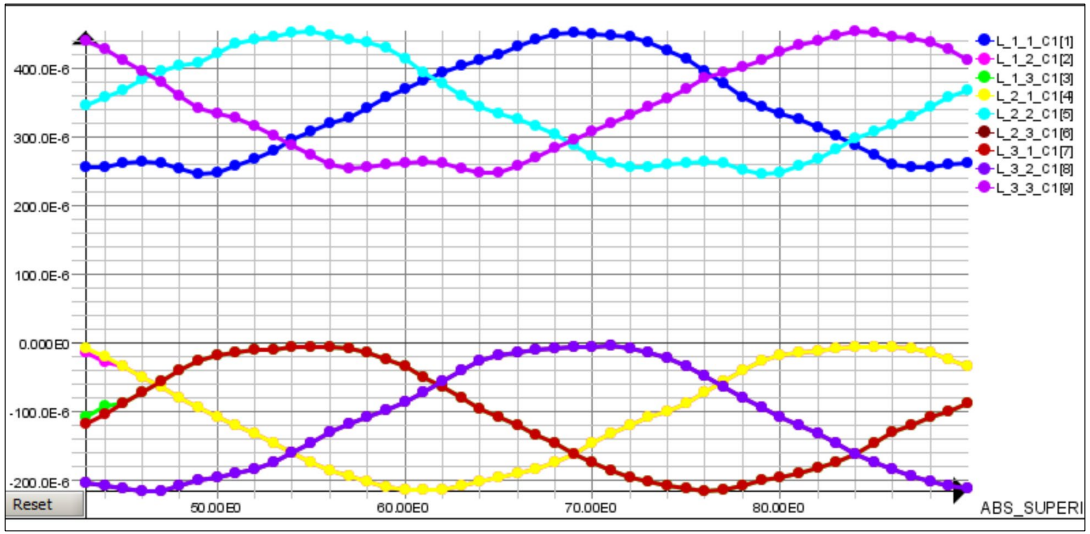
© Altair Engineering, Inc. Proprietary and Confidential. All rights reserved.

Additional Information

Compute Inductance macro

- As **Output**, the macro will create:
  - **Curves** corresponding to each inductance
  - **I/O tabulated parameters** corresponding to each inductance





Questions & Answers

THANK YOU

altair.com

#ONLYFORWARD

ALTAIR

DISCOVER CONTINUOUSLY.
ADVANCE INFINITELY.

Visit altair.com to learn more.

#ONLYFORWARD

ALTAIR