Build Your Own Dedicated Tool with Excel®- Driving Flux!

Fabrice Marion - CEDRAT Group.

Driving **Flux** from **Excel** *(Excel®-Driving)* has been greatly simplified with the new version of Flux 10.3. Before illustrating its potential, reminders are needed.

Some basic notions

Command language

Firstly, we must know and be able to implement the commands described in the Python language. Such language is easily accessible via Flux. Perhaps without even paying attention, each command in Flux (for instance creating entities like points, regions and so on) is automatically recorded in a python file (*.py).
After that, the user can automatically replay all the commands previously implemented by executing the python file. Note that the name and the order of the sequence are customizable. This method is used to automate the various phases to build a model in Flux environment (geometry.py + meshing.py + physic.py,+…). This allows one to formalize and preserve modeling methods in order to be able to reuse them later.

Application and Programming Interface (A.P.I.)

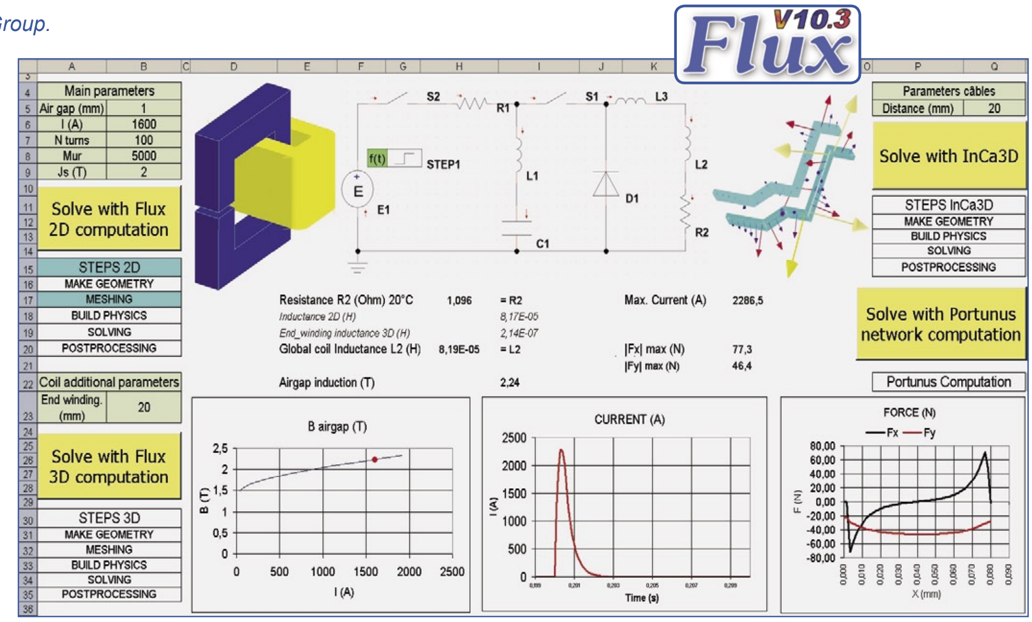
An A.P.I. is a common space in which data, functions, commands, procedures, … are pooled to allow different software to communicate together.
To do this, such software must be provided with entry points to allow for their control. For example, we can consider Visual Basic (V.B.A.) as an A.P.I. of Excel.

Excel®-Driving principles

Thanks to V.B.A. language and python script, driving Flux from Excel has become easier. An Excel file is provided to the user. This file contains a list of all the generic variables and A.P.I./Flux functions declarations (V.B.A. language).
It is necessary to customize the phases of initialization and connection to Flux (in particular, indicate the location of the Flux executable and the memory allocation needed), and then, to define the different phases of the model construction using the command files (python scripts).

A concrete example

To illustrate the potential of "Excel®-Driving", we have described the steps needed to simulate the behaviour of a magnetic induction generator with pulsed-current. In our example, a magnetic field is generated in the magnetic circuit (C-core) by a coil powered with an electric pulsed-current. To produce pulsed-current, an electronic circuit charges and then discharges a capacitor by sequentially driving switches. Finally, taking into account the peak of current generated, we need to know the electrodynamic forces applied between the conductors supplying the device. CEDRAT Software solutions allow one to simulate complete electromagnetic behaviour of the device.
Indeed, inductance – 2D effects – can be computed in the Flux 2D environment,
the impact of magnetic leakage flux – 3D effects – can be computed in the Flux 3D environment.
The voltage discharge of the capacitor through the field coil can be simulated by Portunus[1] and finally, InCa3D[2] is ideal for calculating electrodynamics forces between the conductors supplying the device.



Input parameters needed (green area in the main sheet) for the design of the considered electromagnetic device can be selected
Yellow button to run software in order to perform specific computations
Visualisation of progression of the different sequences of simulation

Figure 1 : All applications needed to design an electromagnetic device can be gathered on the same "Excel®" sheet, to offer your design office a tool with an easy use and strictly adapted to your needs

Main advantages !

Thanks to "Excel®-Driving", all these applications can be gathered on the same Excel sheet (see figure 1). They can be complemented with analytical calculations (computation of the resistance of the coil for example).
The resulting general interface is presented like an exchange area for the four simulation tools (Flux 2D and 3D environments, Portunus and InCa3D).
Exchange for input and output parameters (iterative calculations are possible between different tools if necessary to take into account the saturation in the magnetic circuit due to high values of pulsed-current for example). When integrated dedicated software are fashionable, our example illustrates an alternative method for modeling electromagnetic devices.
This method consists in controlling the modeling and simulating software dedicated to different kind of applications via a general interface such as Excel® or Matlab®. Thus, it becomes possible to easily associate tools dedicated to physical domains (magnetic, electrical, thermal, mechanical,…) that are different but complementary and necessary for the complete study of devices.
This approach gives each company the opportunity to implement "in-home design tools", easy to use and customizable to their needs. Indeed, for the user of the final environment, it is not necessary to have a thorough knowledge of the tools of numerical calculation (geometry, meshing, solver,…).
Only the dedicated application parameters needed are available to define the device.
On the other hand, the personalized driving of standard tools is much more interesting than the use of integrated dedicated software in terms of ergonomy, cost and independence.
Last but not the least, this method of work allows one to specify, to formalize and finally to sustain procedures ("in-home know-how") for analysing and studying of devices. Already, companies such as Schneider-Electric are studying the potential of the solution "Excel®-Driving" in their design processes for controlling software such as Flux and InCa3D.

In order to allow you to test this new approach, we wrote a tutorial dedicated to "Excel®-Driving" (available on request). Examples illustrate this method for driving software and associated computations. All these elements can be downloaded from CEDRAT's web site.

[1] Portunus: Multiphysics systems simulator
[2] InCa3D: Advanced interconnections simulation