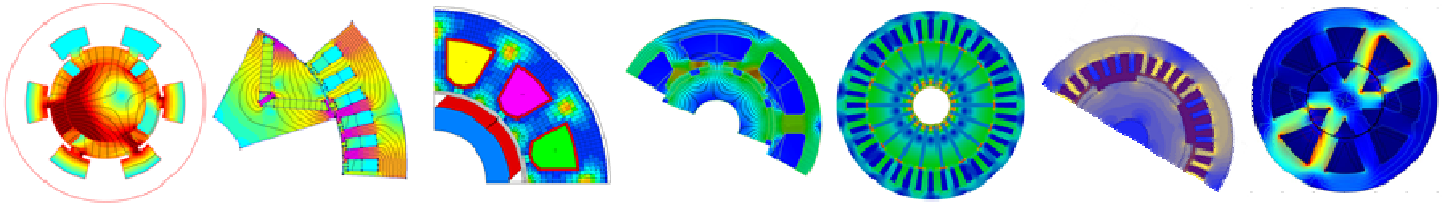
PYLEECAN

PYthon Library for Electrical Engineering Computational ANalysis

PYLEECAN: an open-source Python object-oriented software for the multiphysic design optimization of electrical machines



J. Le Besnerais, P. Bonneel

05/09/18

EOMYS | ENGINEERING CONSULTING RESEARCH

Who's who?

A flying squirrel

An underwater cow

A water bird

A <u>commercial</u> electromagnetic and vibro-acoustic simulation software of electrical machines

An <u>open-source</u> software for the simulation of electrical machines and drives

A consulting company specialized in noise & vibrations of electrical machines

EOMYS

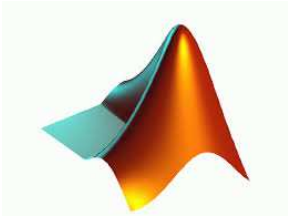
- *Young Innovative Company* created in 2013 in Lille, North of France
- Activities : **engineering consultancy** specialized in NVH of electrical systems
- 7 R&D Engineers (electrical engineering, vibro-acoustics, scientific computing)
- Sectors : transportation (railway, automotive, marine, aeronautics), energy (wind, hydro), home appliances, industry

A few of our references:

PORSCHE · LILIUM · PSA GROUPE · ebmpapst · GOLDWIND · Mando

EOMYS | ENGINEERING CONSULTING RESEARCH

Pyleecan's origin



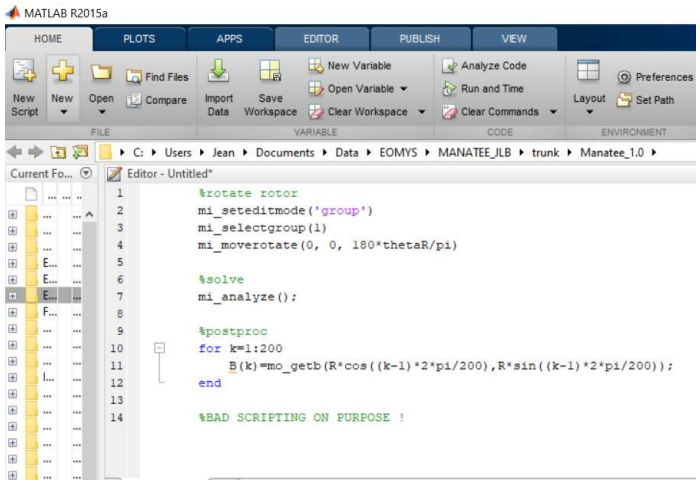
MANATEE was initially developed in **Matlab** proprietary language (but no toolbox):

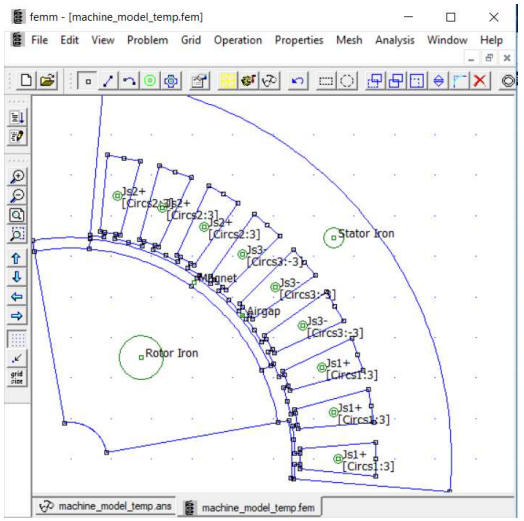
- Commercial licence
- Closed code for core functions
- Not suitable to create an advanced software GUI

A new MANATEE version has been started in Object Oriented Python

Pyleecan's origin

Who (or one of your PhD student) has already scripted a coupling between Matlab/Scilab/Octave and Femm?



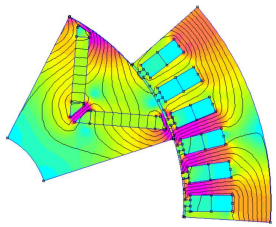


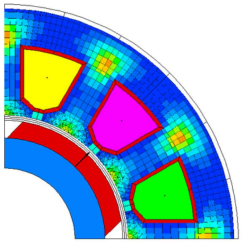
A new version of FEMM has been released 25th February 2018 including sliding mesh:

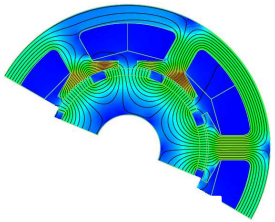
Did you all modify your coupling scripts?

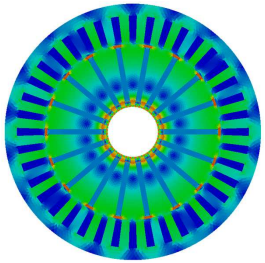
Pyleecan's origin

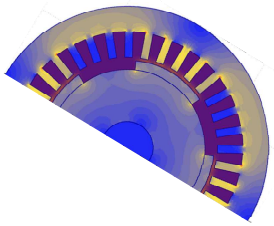
Femm is not the only free software which requires to **draw a geometry, define magnetic materials, and mesh:**

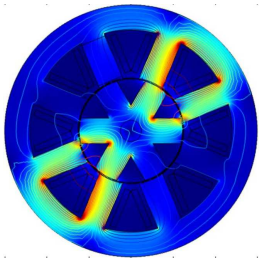

Femm


FEMAG


GetDP


Code_CARMEL


Agros2D


Elmer

Commercial FEA software also includes scripting features to drive their solver.

In the worse case, this scripting work is lost, not enough documented or not good enough so that next researcher has to make it again...

PYLEECAN

Pyleecan aims at **capitalizing all the scripting and modelling efforts of the electrical engineering R&D community** and improve software quality by:

- Code peer review
- Standardization
- Documentation
- Validation cases

Let's spend less time on scripting and more time on physics & creativity!

Pyleecan's objectives and key features

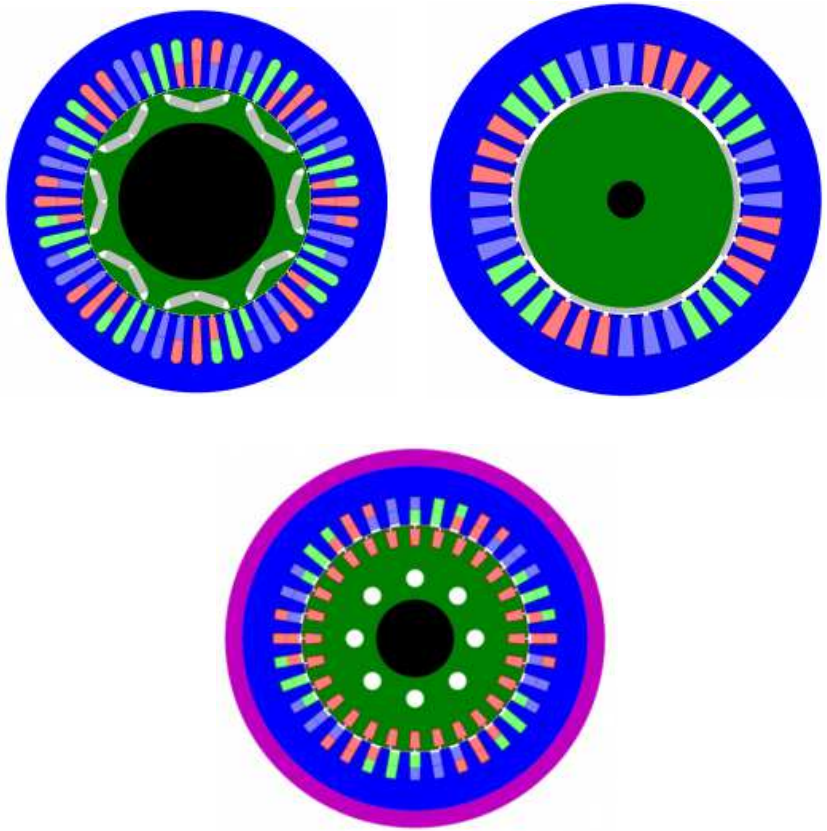
A **research-friendly, unified, flexible simulation framework for the design and optimization** of electrical machines:

- Object-oriented modelling of electrical machines
- Python based software
- Handling of multiphysics and different granularity levels
- Optimization, parallelization and data visualization tools
- Graphical User Interface
- Online documentation

Pyleecan's scope

Pyleecan aims at simulating:

- Surface, Surface Inset or Interior Permanent Magnet Synchronous Machines (**SPMSM, SIPMSM, IPMSM**) with inner or outer rotor
- Squirrel Cage Induction Machines (**SCIM**) and Doubly Fed Induction Machines (**DFIM**)
- **Every other machine type** when prioritized by the community



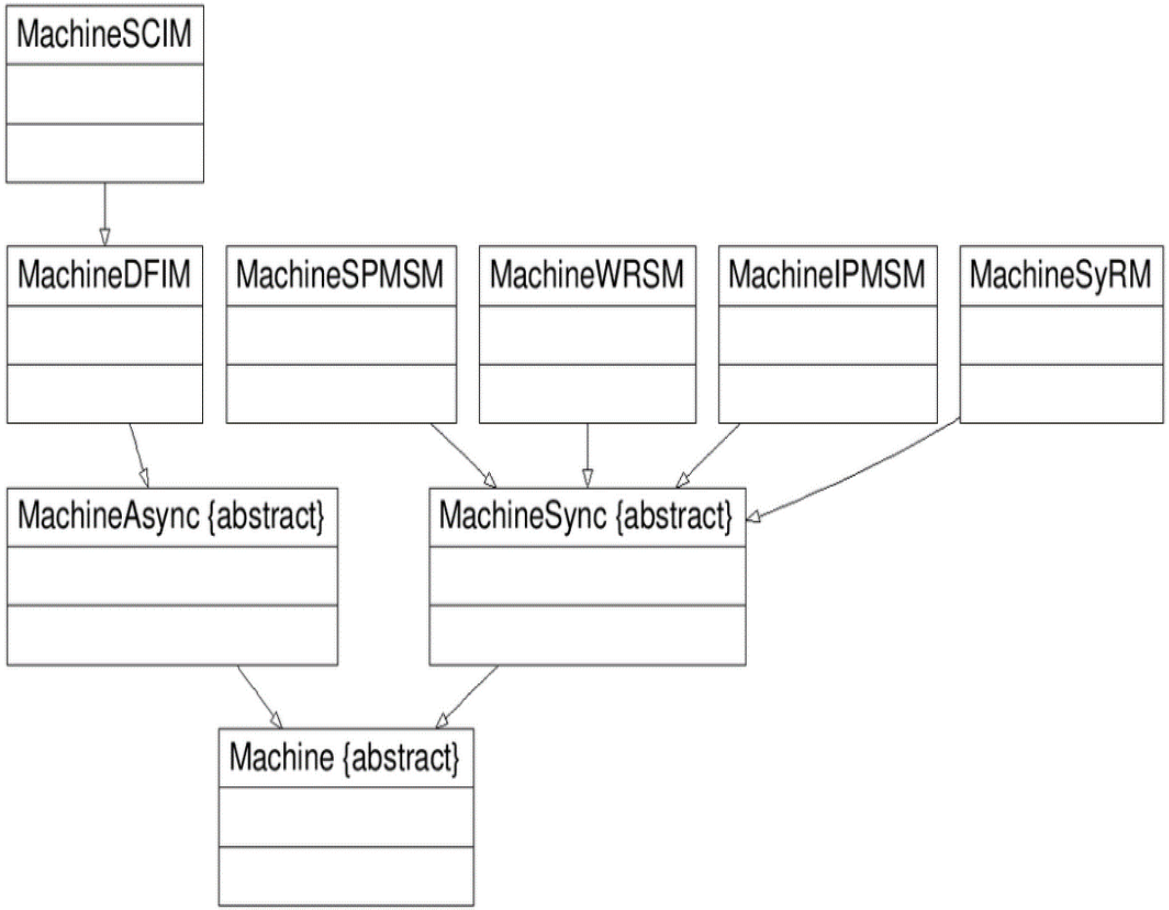
Don't be afraid of Objects!

Object Oriented Programming enables to organize and optimize the code

Everything is designed as general concept (i.e. the object)

OOP enables to design interfaces: "**black boxes**" part of the code that can be interchanged

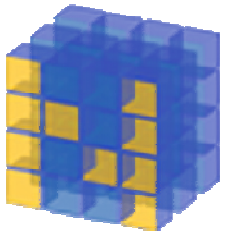
Pyleecan will include **tutorial** to understand how to work in OOP

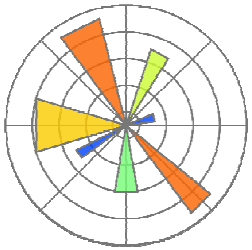


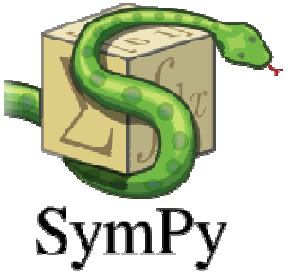
Don't be afraid of Python(s)!

Pyleecan is developed in Python because:

- Free and open
- Designed to be easy to read and to use
- Large and active scientific community



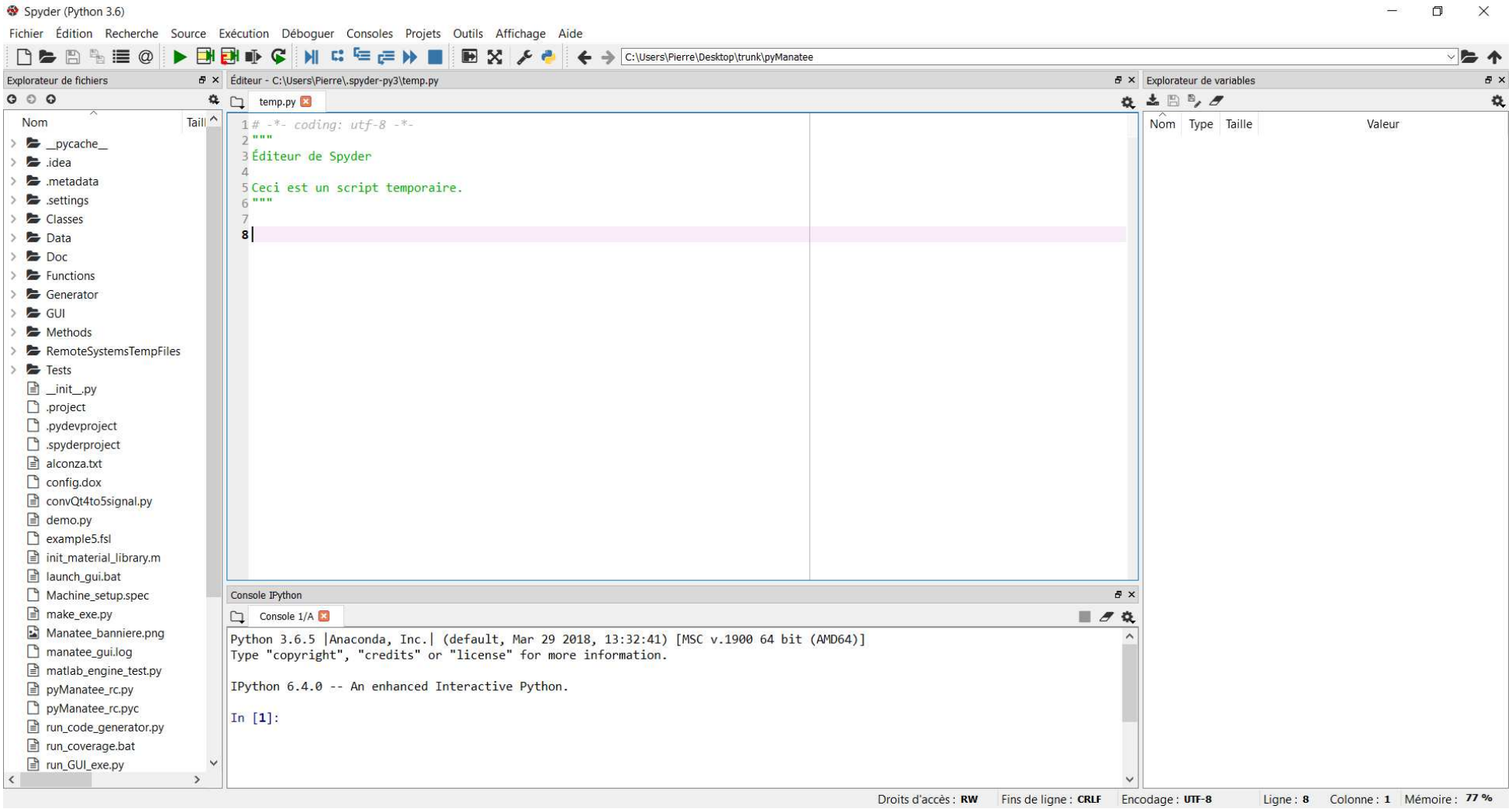




ANACONDA®

We don't use C++ (for instance) because we want to keep Pyleecan as simple as possible so that **any scientist can participate**

EOMYS | ENGINEERING CONSULTING RESEARCH



The Open Source approach

Open Source's benefits are:

- Code Peer review
- The user can know exactly how everything is computed
- The user can customize the software

GitHub

Pyleecan will use an Apache licence:

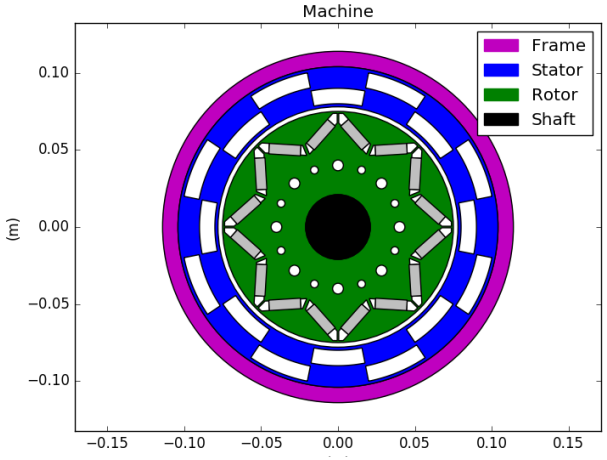
- Anyone can read, use, change the code
- **You don't have to share your confidential code** if you work with Pyleecan
- Pyleecan can be included **even in closed source commercial software**

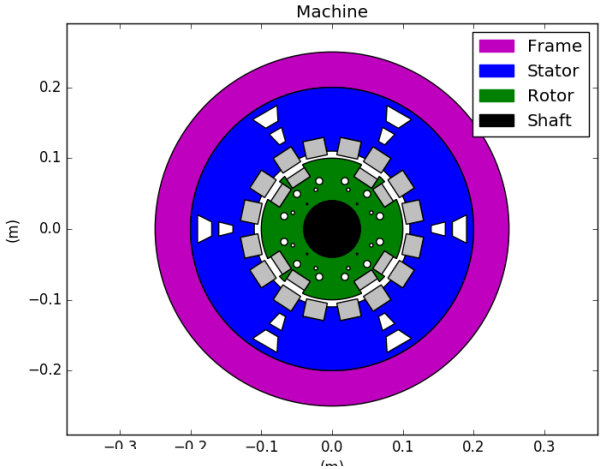
How EOMYS will sustain Pyleecan?

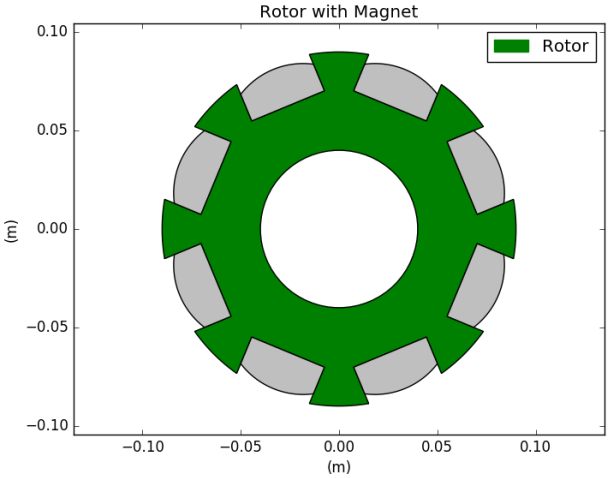
- An open source project, **not an EOMYS product**
- Initial deposit **created from a part of MANATEE commercial code**
- Current maintainer list: 6 persons (includes universities & industries)

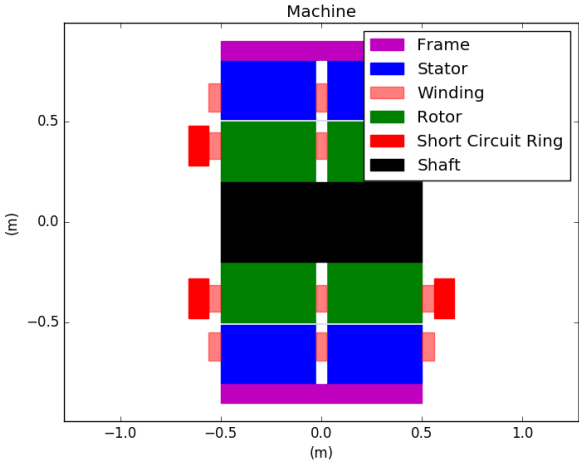
- **1.5 full time developer** on Pyleecan provided by EOMYS
- A new MANATEE version will be based on Pyleecan – PyMANATEE
- EOMYS would sell technical support on PyMANATEE and Pyleecan

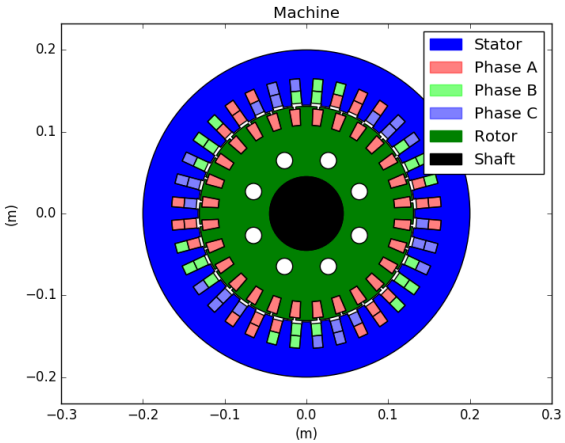
How it may look like (geometry)



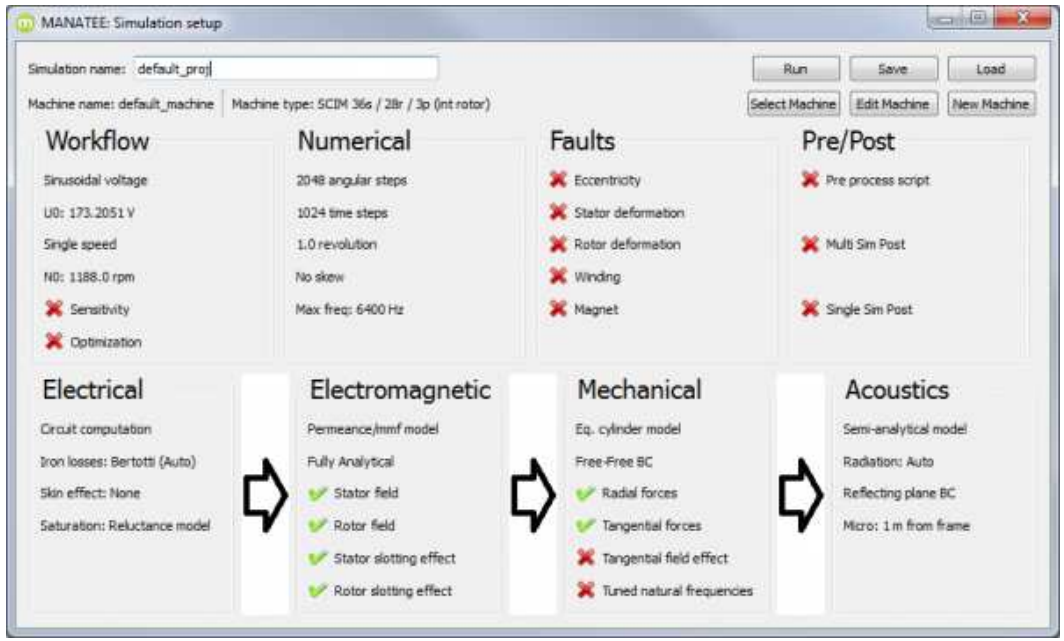


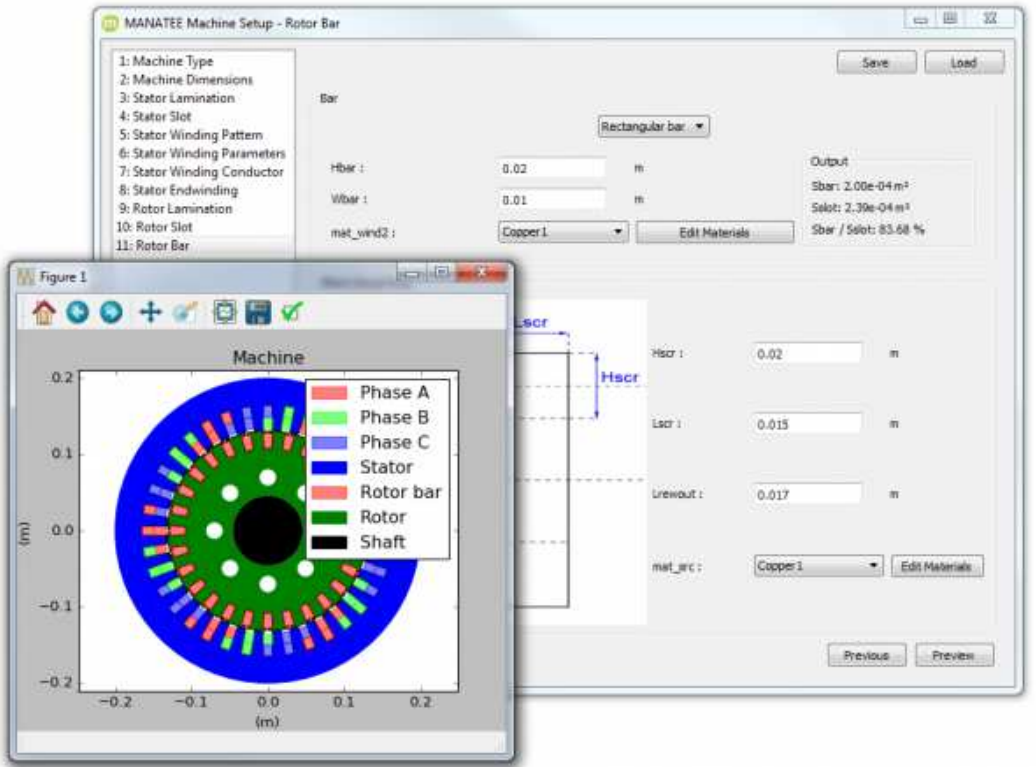






How it may look like (GUI)

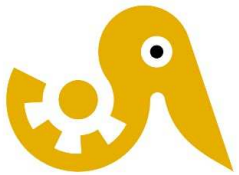




How it may look like (scripting)

```
from Classes.Machine.MachineSCIM import MachineSCIM

from Classes.Machine.LamSlotWind import LamSlotWind
from Classes.Machine.LamSquirrelCage import LamSquirrelCage
from Classes.Machine.Frame import Frame
from Classes.Machine.Shaft import Shaft

from Classes.Slot.SlotW10 import SlotW10
from Classes.Slot.SlotW21 import SlotW21

from Classes.Machine.Winding import Winding
from Classes.Machine.CondType11 import CondType11
from Classes.Machine.CondType21 import CondType21

from Classes.Machine.VentilationCirc import VentilationCirc

# Stator design
stator = LamSlotWind(Rint=0.1325, Rext=0.2, Nrvd=0, L1=0.35, Kf1=0.95,
                     is_internal=False, is_stator=True)

stator.slot = SlotW10(Zs=36, H0=1e-3, H1=1.5e-3, H2=30e-3, W0=12e-3,
                      W1=14e-3, W2=12e-3)

stator.winding = Winding(qs=3, type_winding=3, Lewout=15e-3, p=3,
                         coil_pitch=5, Ntcoil=7, Npcpp=2)

stator.winding.conductor = CondType11(Nwppc_tan=1, Nwppc_rad=1,
                                      Wwire=10e-3, Hwire=2e-3,
                                      Wins_wire=0, type_winding_shape=0)
```

```
# Rotor design
rotor = LamSquirrelCage(Rext=0.131, Rint=45e-3, L1=0.35, Kf1=0.95,
                        is_internal=True, is_stator=False,
                        Hscr=20e-3, Lscr=15e-3, Nrvd=0)

rotor.slot = SlotW21(Zs=28, H0=3e-3, W0=3e-3, H1=0, H2=20e-3, W1=13e-3,
                     W2=10e-3)

rotor.winding = Winding(Ntcoil=1, qs=28, Lewout=17e-3, Npcpp=1,
                        coil_pitch=0, type_winding=10)

rotor.winding.conductor = CondType21(Hbar=0.02, Wbar=0.01, Wins=0)

rotor.axial_vent = [VentilationCirc(Navd=8, Davd=20e-3, Havd=70e-3,
                                    Alpha_avd=0)]

# Shaft and Frame Design
shaft = Shaft(Drsh=90e-3)
frame = Frame(Rint=0.2, Rext=0.2)

# Assemble the machine
test_obj = MachineSCIM(stator=stator, rotor=rotor, shaft=shaft, frame=frame)

# Play with it
test_obj.plot()
test_obj.rotor.plot()
test_obj.stator.comp_mass()
...
```



PYLEECAN

PYthon Library for Electrical
Engineering Computational ANalysis

- Pyleecan aims at coordinating existing and future open-source / free developments in electrical engineering (e.g. Syr-e, Xfemm, femagtools)
- Regular meetings are planned to prioritize developments

next meeting: 05/09/18 (today) 17:30 Orfeas Hall
newsletter: http://eepurl.com/dov3PH
website: www.pyleecan.org