



EDAM: Element Definition AutoMator

Documentation

Yaël MOUSSOUNI

University of Strasbourg, Department of Physics and Engineering

Sunday 25th June, 2023

Contents

1	Introduction	2
2	Installation	2
3	Configuration	2
3.1	rotor position and grapher	2
3.2	Uncertainties	3
4	Run	3

1 Introduction

EDAM is an extension to FROMAGE, that can be used to generate multiple configuration files at once and execute FROMAGE on each one of them. EDAM itself is a group of three programs (`rotor_position`, `edam_uncert` and `grapher`) that can be executed independently depending on your goal. EDAM is written using Python (tested using version 3.10), and requires the libraries `os`, `subprocess`, `numpy` and `matplotlib`, `scipy` and `pandas`.

2 Installation

The python files (`edam_rotor_position.py`, `edam_uncert.py`, `edam_grapher.py`) can be placed in the same directory as FROMAGE.

3 Configuration

3.1 rotor position and grapher

`edam_rotor_position.py` generates multiple configuration files describing the displacement of the rotor along one coordinate. `edam_grapher.py` is only there to visualize the result and compare multiple models, along with the analytical model.

In the file `edam_rotor_position.py` :

```
1 type_model = "simple" # choosing the model to use (available :  
    simple, ncal_04)  
2 var_X = "Z" # choosing the displacement coordinate denoted X (  
    available : R, PHI, Z)  
3 prefix = "EDAM_{}_{}_{}".format(type_model, var_X) # Prefix of the  
    cfg_files  
4 directory = "cfg_files/EDAM/" # directory for cfg files  
5 out_dir = "./" # directory of the output  
6 output_csv = True # output a CSV with the results ?  
7 sep = ";" # CSV columns separator
```

```
8 endl = "\n" # CSV lines separator
9 r, phi, z = 1, 0, 0 # Default position (defines statics coordinates
10 X_min = 0 # minimum value of X
11 X_max = 4 # maximum value of X
12 N = 100 # number of values of X
13
14 ## FROMAGE
15 STEP = "10 36" # [angle] [number of steps]
16 ARM_LENGTH = "3000"
17 SIGNAL = "2"
```

In the file `edam_grapher.py` :

```
1 var_X = "PHI" # Same as rotor_position
2 prefix = "EDAM_simple_{}_".format(var_X) # Prefix of the cfg_files
   (same format as rotor_position)
3 super_prefix = "EDAM_ncal_04_{}_".format(var_X) # Prefix of the
   superposed file (same format as rotor_position)
4 directory = "cfg_files/EDAM/" # working directory for cfg files (
   same as rotor_position)
5 out_dir = "./" (same as rotor_position)
6 sep = ";" # CSV columns (same as rotor_position)
7 endl = "\n" # CSV lines (same as rotor_position)
8 superposition = True # Replot a second model
9 analytical = True # Replot the analytical model
10 N = 1000 # nb of points for analytical model
11 r, phi, z = 1, 0, 0 # default parameters (same as rotor_position)
12 # m, DEG, m
13 m = 2.288 # kg - mass of the rotor quadrant
14 M = 42.3648 # kg - mass of the mirror
15 R_prime = 66.217e-3 # m - rotor masse center distance from the axis
16 to_png = True # export to png ?
17 to_screen = True # show the result ?
```

3.2 Uncertainties

This file is not well written and can be improved. However, some configuration are possible at the beginning of the file. The result will only be printed in a csv file. To print it in the terminal, remove the `#` at the beginning of the last lines.

4 Run

Once the file has been configured, you can run it using one of the following command :

```
1 python edam_[...].py # Linux and Windows
2 python3 edam_[...].py # Linux, MacOS and Windows
3 python3.10 edam_[...].py # Specify which version of python to use
```

Note: `edam_grapher.py` must be executed after `edam_rotor_position.py`.