Memorandum

idm@F4E Ref:	F4E_D_2F2ARF v.1.2	Date	2019-07-08

Subject: **iWW-GVR: A tool to manipulate MCNP weight window (WW) and to generate Global Variance Reduction (GVR) parameters**

From: **M. Fabbri, Á. Cubí**

Department: A&C / Engineering Unit / ITER Department

Contact alvaro.cubi@ext.f4e.europa.eu, marco.fabbri@f4e.europa.eu

To: **Everybody interested**

In order to keep F4E nuclear analysis capabilities up-to-date with the latest, state-of-the-art method and post-processing tools, a new practical tool called iWW-GVR to manipulate MCNP weight window and to generate global variance reductions format has been developed.
The tool is a python 3.6 based script able to generate global variance reductions weight window (WW) using any mesh format in meshtally by D1SUNED, MCNP5 or MCNP6. Mesh format includes usual MCNP column or matrix format and also specific D1SUNED format including cell or isotope contribution binning and source mesh importance format. Both Cartesian and cylindrical meshes can be read. The tool incorporates also simple functions to operate with weight windows (e.g., analyse, add and remove WW set, write, plot).
The tool is used through a text based interactive menu, and it can be run under Windows system. It is worth reminding that this tool employs the Cython package to improve the efficiency of the operations taking advantage of C native libraries and functions.

What iWW-GVR does

Capabilities of iWW-GVR are divided in four categories:

- WW information
- WW analysis
- Operation on WW
- Generation of GVR WW

WW information:
Display a summary of the WW sets contained and the correspondent detailed information (No. sets, dimensions, No.bins, No. Voxel).

WW analysis:
Analysis of the WW sets providing to the user the minimum and maximum value, the maximum ratio between nearby voxels and the corresponded histogram , the per cent No. of bins with value bigger that zero, reports the negative values (if present), the voxel average dimensions and volume.

Operation on WW
Operation on WW allows performing various operations as[1]:

- Add: to add a WW set copying the first one, normalizing hence soften it according to the parameters inserted. In addition, the user has also the possibility to fill automatically "all the holes" present within the domain using the closest value different from zero[2].
- Remove: to remove the first or the second WW set[3].
- Mitigate: to decrease the ratio between nearby voxels which are bigger than the threshold inserted by the user. The single WW_{value} is computed as follow:

$$WW_{value} = \frac{\max|WW_{nearby}}{\max|\,WW_{ratio}*0.9}$$

  Where:

  - $\max|WW_{nearby}$ is the maximum value of the nearby cell
  - $\max|WW_{ratio}$ is the maximum ratio with the nearby cell

- Plot: to plot in an external JPEG file a user defined plane of a specific particle and energy.
- Write: to write the modified WW in the MCNP ASCII format or to VTK format which can further analyzed/manipulated in dedicated readers (e.g., Paraview)

Generation of GVR WW:
To generate GVR WW employing a meshtally file given by the user[4] [REF]. Taking advantage of the mesh2vtk libraries [REF], the meshtal is loaded, selected and hence processed as follow:

$$WW\_GVR_i = \frac{\Phi_i}{\max|\,\Phi_{\text{value}}} * \frac{2}{\beta + 1} * softening_{factor}$$

where:

- Φ_i is the mesh tally voxel value at the generic position i
- $\max|\,\Phi_{\text{value}}$ is the maximum value of the mesh tal selected
- β is the maximum splitting ratio
- $softening_{factor}$ is a factor between]0,1] to mitigate the splitting/russian roulette [REF]

The WW-GVR generated has a single energy bin and the dimension/discretization of the meshtally selected. If necessary, the WW can be further manipulated by the routine.

---

[1] WW headers are updated if necessary according to the user selections.
[2] This operation can be done if only one WW set is present.
[3] This operation can be done if only two WW sets are present.
[4] The meshtally selected shall have only one energy bin.

How to create a new package

How to install it

The iWW-GVR is distributed in a preassembled package which automatically installs the script itself and all the related modules (i.e., dependencies) if not already present. So, browse to the dist folder and type following command line:

```
pip install iww_gvr-<<version>> --user
```

Moreover, it is recommended to create a dedicated virtual environment or a conda env to employ all the tested versions.

It is compound of two modules, the principal module "iWW_GVR.py" and a modules dependency "iWW_GVR_engine.pyx". The path of the location of the *meshtal_mod.py* module should be adjusted in the "*dep_path*" variable located in the first line of the mesh2vtk.py script. In addition, the script uses others package dependencies which if they are not installed in the system, they could be downloaded or installed as follow:

How to use it

Hereinafter the most reliable method to use is Elbow2Cyl routine is explained.

1. Open the CAD model with SpaceClaim.
2. Open the SpaceClaim Script Editor: **File>New>Script**.
3. Load the Elbow2Cyl on the Script Editor clicking on the folder icon (see Figure 4)

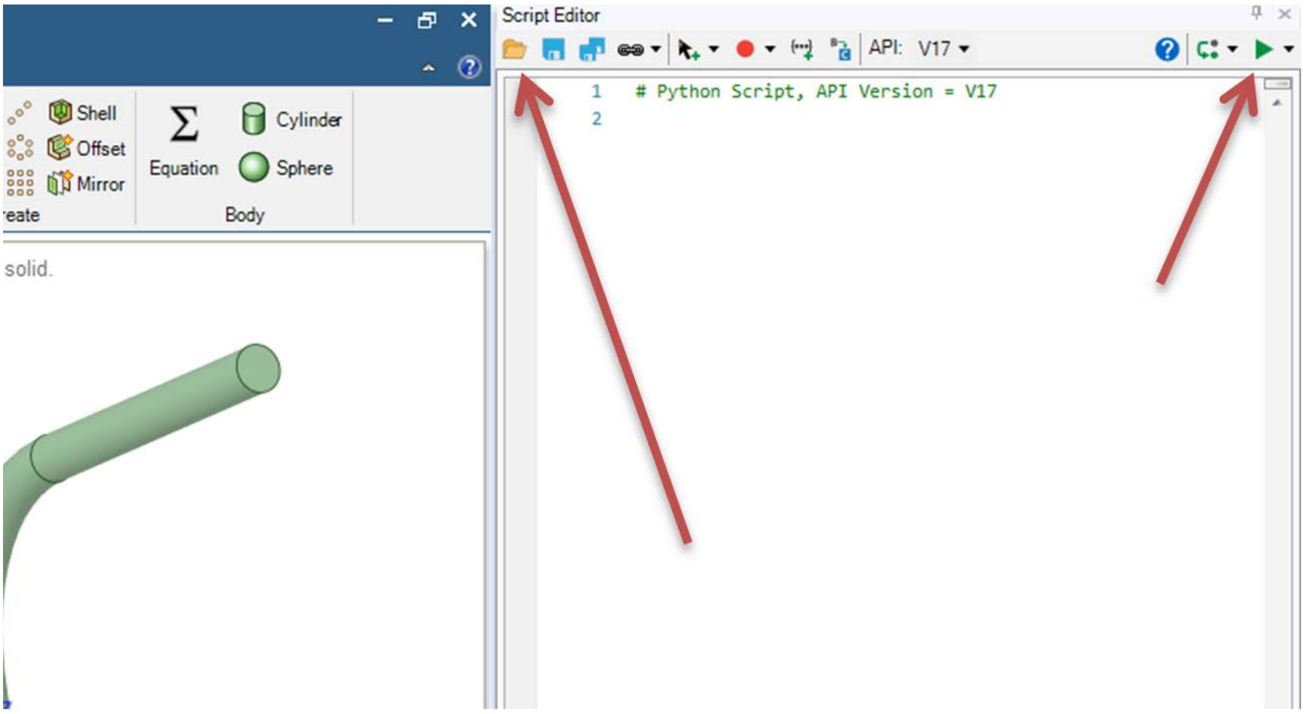

Figure 4 - Script Editor window

4. Create and set the value of Parameter1in the **Groups** window (see Figure 5).

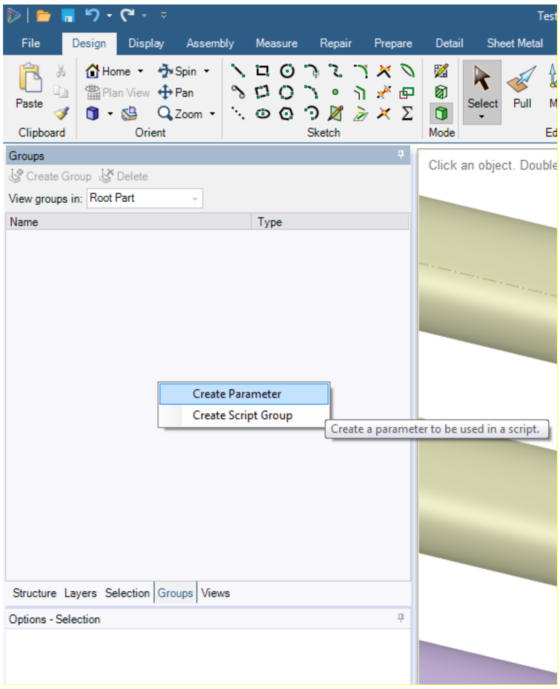

Figure 5 - Groups window

5. Select the surface of the elbow to be converted left clicking on it.
6. Execute the routine by clicking on the **Run** button (see Figure 6).

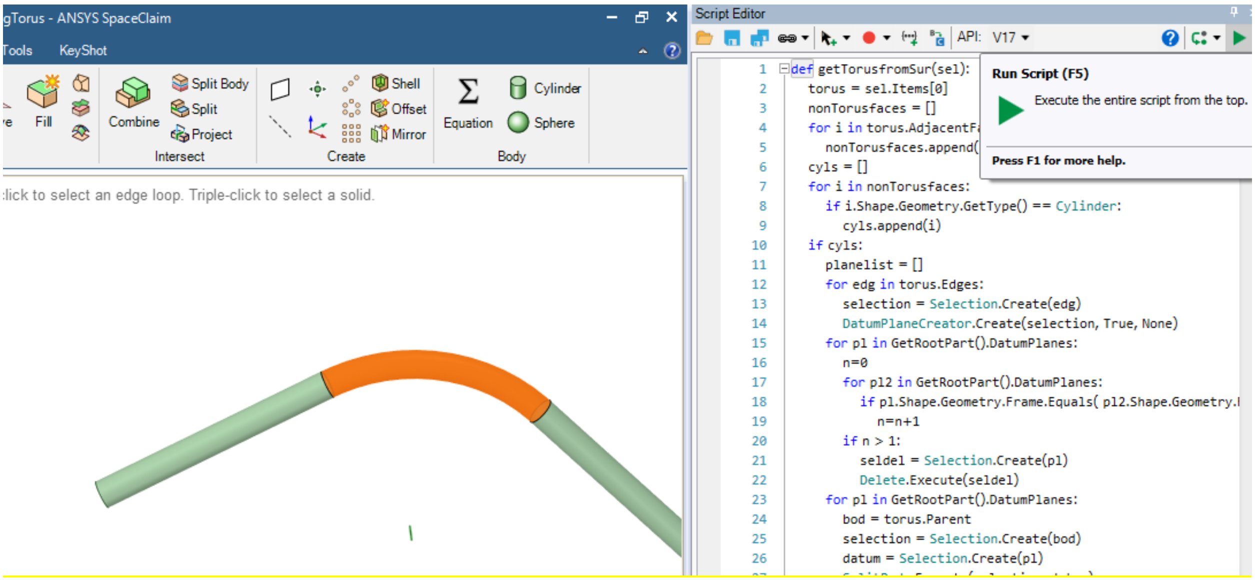

Figure 6 - Executing the script

Recommendations and warnings

- Voxel volume averaged over all the domain.
- ZoneID definition limitation
- Mitigate limitation: no iteration may be bigger than the imposed value
- GVR cannot employ meshtal with more than one energy bins.
- Plot are not allowed for cylindrical mesh
- It is recommended to save before using the script.
- It is recommended having the elbow connected at both extremities to connecting pipes. In this way, the routine will automatically and smoothly adjust the transition between the straight pipes and the elbow.
- This routine can deal with isolated elbows but not for those connected to a pipe only at one of extremity.
- It is recommended to have the elbow and the pipes in the same body. If not, they can be easily merged as follow **Design>Intersect>Combine**.

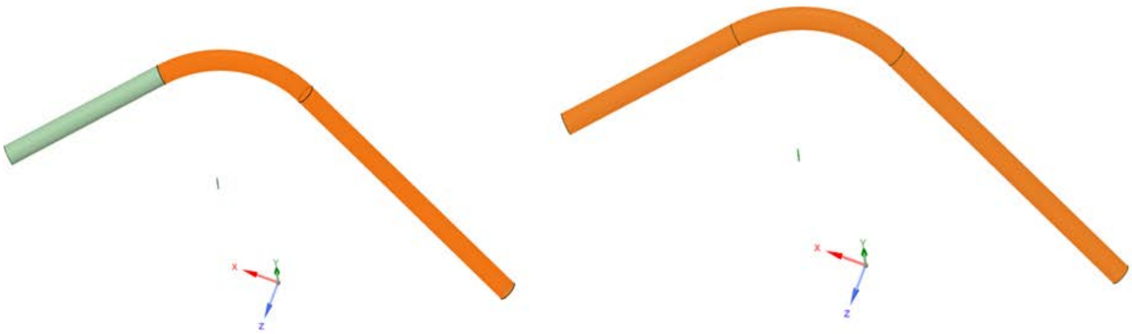

Figure 7 - Before and after merging

- The routine might split the body at some undesired locations due to the large extension of the element as showed in Figure 8. A combine operation can easily mitigate this.

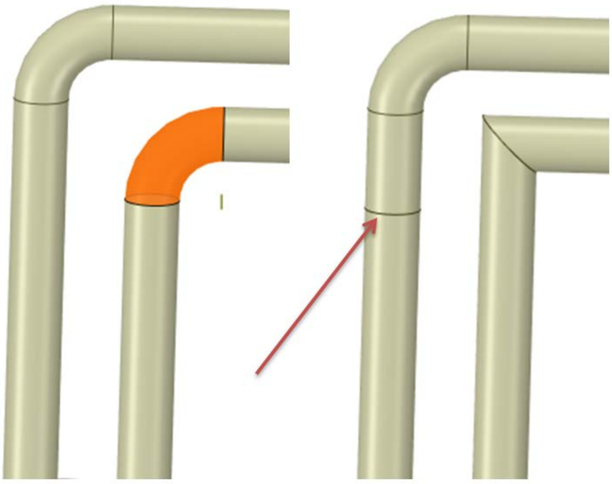

Figure 8 - Pipe being split by the script

- The script could fail to execute printing the error message "Unable to cut the body": this might be solved merging the elbow with the two connecting pipes.
- Sometimes when executing the routine the SpaceClaim graphics crashes due to the too frequent display refreshing. In this case, please save and open again the model.

Further developments

- Insert the flipping methodology as a further option of the WW manipulation
- Capacity to parse and manipulate cylindrical meshes

iWW-GVR verification process

By Alvaro

How to build a distribution package for your code

https://packaging.python.org/tutorials/packaging-projects/

https://python-packaging.readthedocs.io/en/latest/command-line-scripts.html

How to build a Cython extension for your code

Please follow these instruction to build a Cython extension for your code, [REF]:

1. Save your python code as wwgvr.pyx
2. Create a setup.py which should look like

```
from distutils.core import setup
from Cython.Build import cythonize

setup(_modules=cythonize("wwgvr.pyx",annotate=True),)
```

3. Build your Cython file use the commandline options, which will create a file with the suffix .pyd:

```
$ python setupCYTHON.py build_ext --inplace
```

4. Now to use this file: start the python interpreter and simply import it as if it was a regular python module:

```
import wwgvr
wwgvr.main()
```

5. Rename your files only if strictly necessary and enjoy!

Reference:

1. Cython tutorial, https://cython.readthedocs.io/en/latest/src/tutorial/cython_tutorial.html#the-basics-of-cython
2. https://cython.org/