FUSION FOR ENERGY

Memorandum

idm@F4E Ref:	F4E_D_2F2ARF v.1.3	Date	2019-11-18

Subject: **SCtools: Elbow2Cyl guidelines**

From: **Álvaro Cubí**

Department: External Contractor. Technical Assistant working under a Fusion for Energy contract Supporting A&C Group in Nuclear Analysis Modeling and Calculation.

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

To: **Everybody interested**

IronPython routines can be used in SpaceClaim CAD software to automatize operations decreasing the time needed for the CAD manipulations and analysis. SCtools is a toolbox containing routines useful in the CAD radiation transport simulation process. Elbow2Cyl is a routine released in the SCtools context which automatically substitutes the elbow of a pipe or tube into a simplified geometry made of cylinders as required by the MCNP modelling rules. Elbow2Cyl has two main branches, each performing a different and independent simplification process. The process used will be decided according to a parameter provided by the user. The script has been extensively tested with SpaceClaim version 19.2 running on the API V17.

What Elbow2Cyl routine does

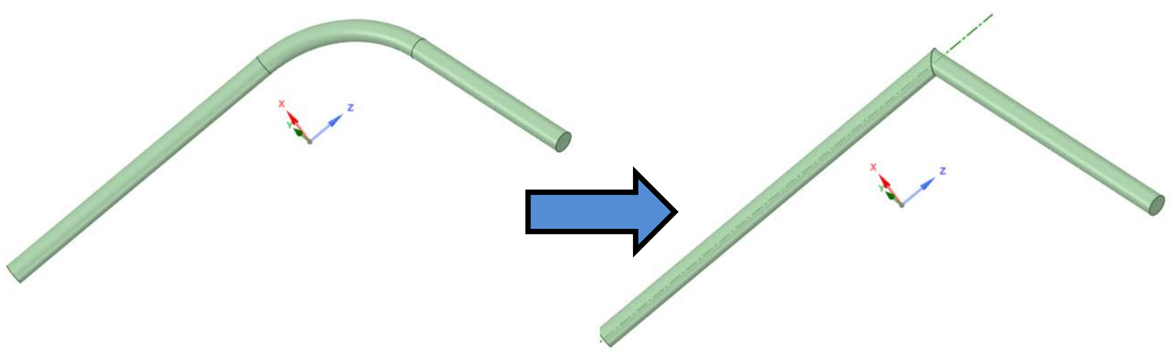
The radiation transport simulation software MCNP cannot represent a 'tilted[1]' torus by means of the CSG geometry neither splines. Unfortunately, it is very common to encounter these features in piping or complex shapes. Therefore, they have to be simplified prior to the geometry CAD conversion.

Along this line, Elbow2Cyl routine simplifies an elbow into cylinders which, hence, can be converted to MCNP. Two different simplification methods are available and selectable imposing a different "Parameter1" value:

- Method No.1 [Parameter1≤1]: the elbow is converted into two cylinders tangent to the surface extremities. The two resulting bodies are jointed to bodies in contact (if present).
- Method No.2 [Parameter1>1]: the elbow is replaced by a number of cylinders which have their extremities relying on the curve axis. The number of bodies employed is equal to the "Parameter1" value.

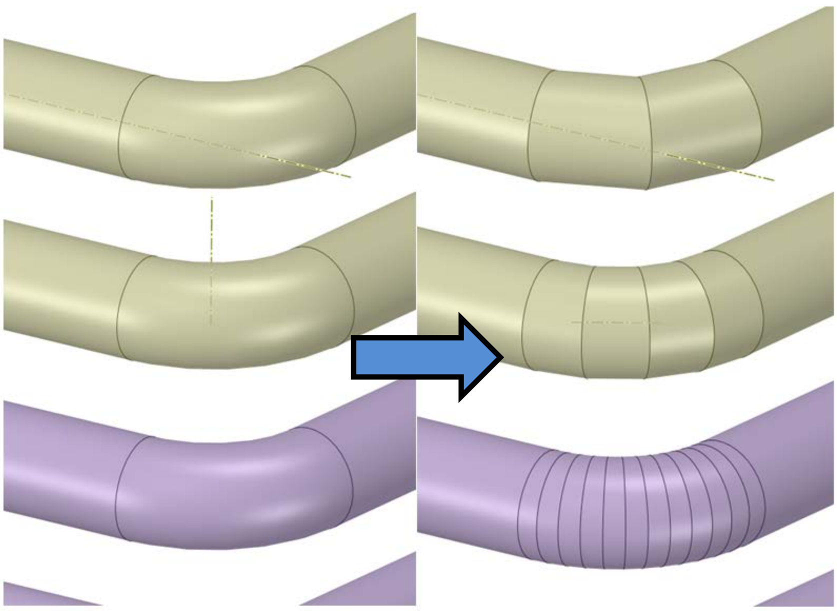
Whereas, the Method No.1 limits the complexity of the MCNP of the model, Method No.2 employs more surfaces which allow a better representation of the original shape hence a smaller mass discrepancy.

Both methods can be applied to solid and hollow pipes which are described by a torus or a spline surface.



Before **After**

Figure 1 - Method 1 example

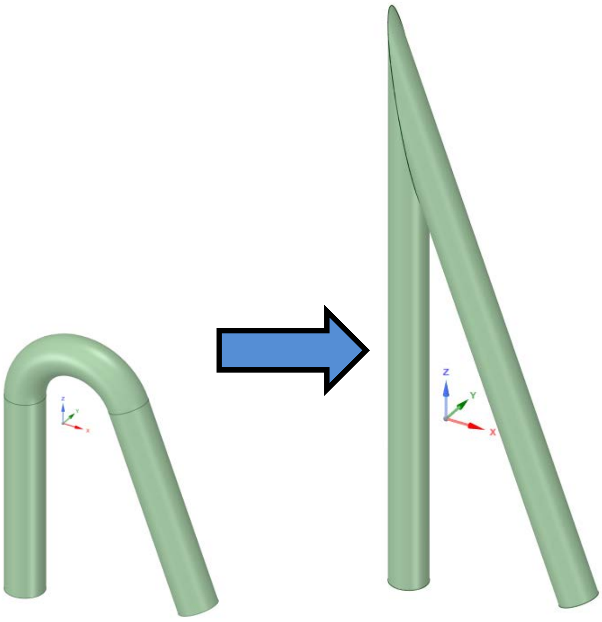


Before **After**

Figure 2 - Method 2 example

[1] A torus which revolution axis is not one of the three main coordinate axis (X, Y, Z).

Moreover, Elbow2Cyl routine can be used for angle up to 180 degrees. Nevertheless, for piping which angle exceeds 90 degrees method No.2 is recommended as the usage of method No.1 would imply an exponential increase of pipe length hence a substantial mass deviation. Figure 3 shows this increase in pipe length after applying the method No.1 to a pipe with a 160 degrees elbow.



Before **After**

Figure 3 – Method No.1 for 160 degrees angle

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)

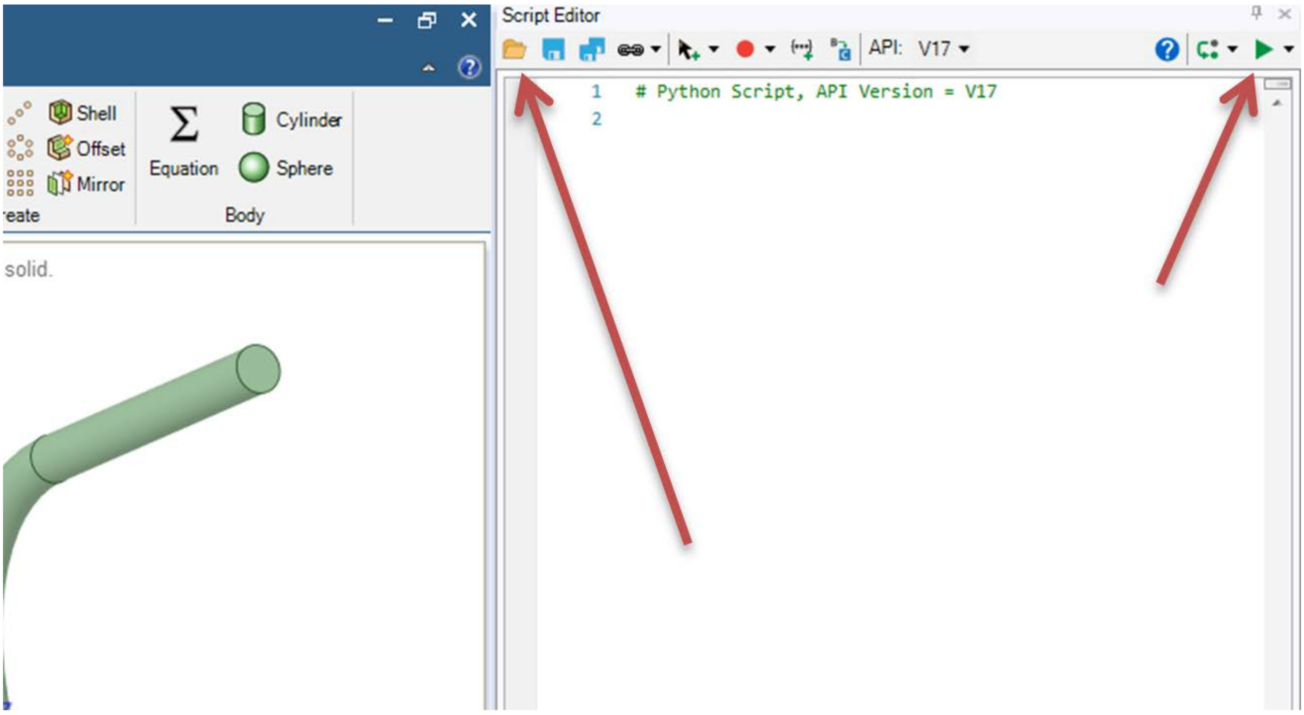


Figure 4 - Script Editor window

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

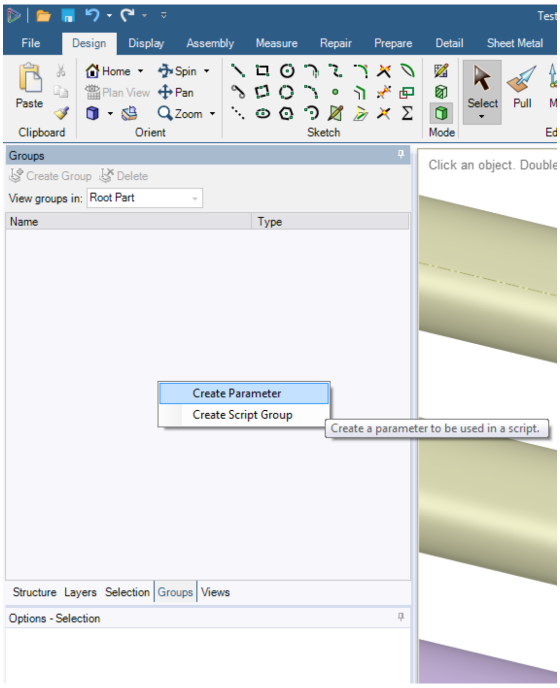


Figure 5 - Groups window

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).

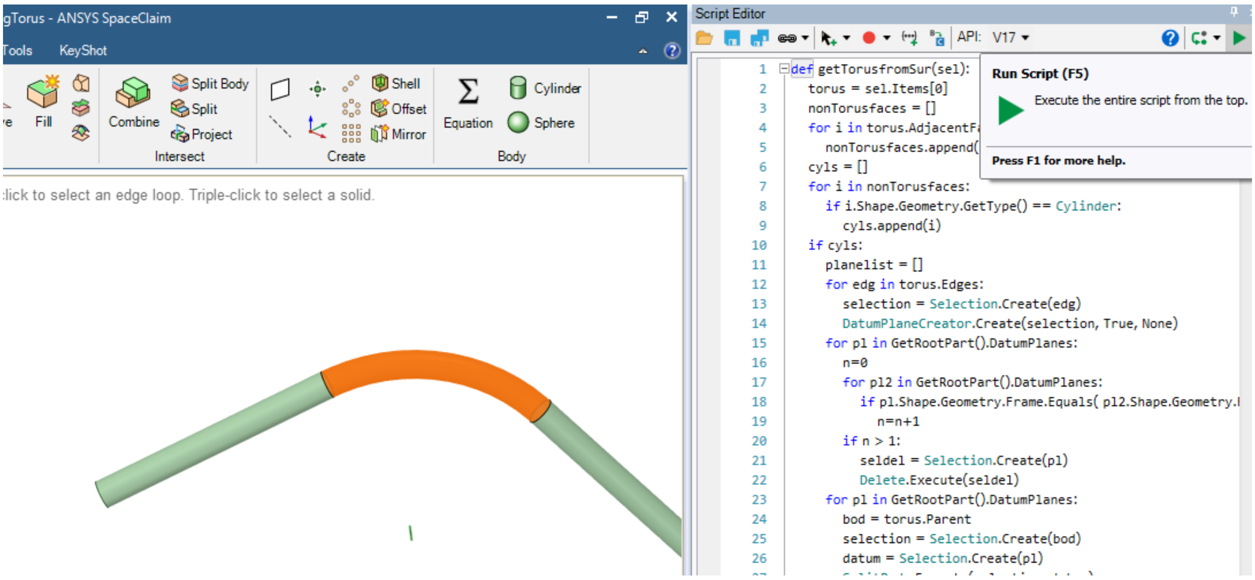


Figure 6 - Executing the script

Recommendations and warnings

- 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**.

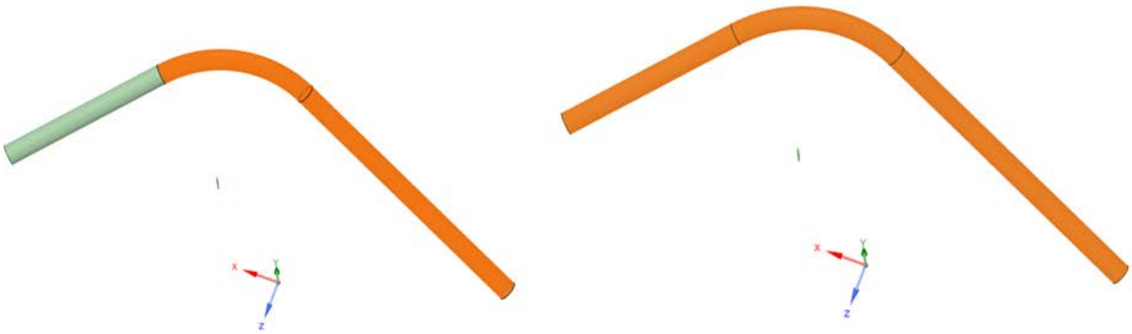


Figure 7 - Before and after merging

- 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.

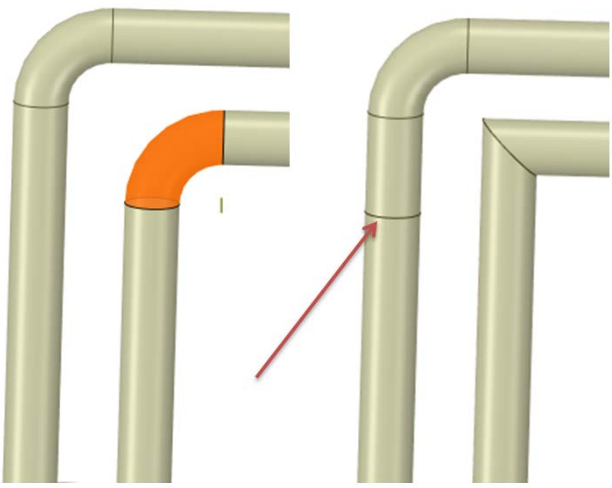


Figure 8 - Pipe being split by the script

- 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

- To insert artificial time-delays in different routine positions to avoid the unpleasant, but seldom, SpaceClaim graphics crashes.
- Take advantage of the SpaceClaim tool publication feature to avoid the unpleasant, but seldom, SpaceClaim graphics crashes. This option might be explored once this tool is not anymore a beta version.
- To split the multiple cylinders created by means of the method No.2.

Mass variation

Component mass variation depends only on the simplification method employed and on angle between the elbow ends as demonstrated in the analytic formulas derived next.

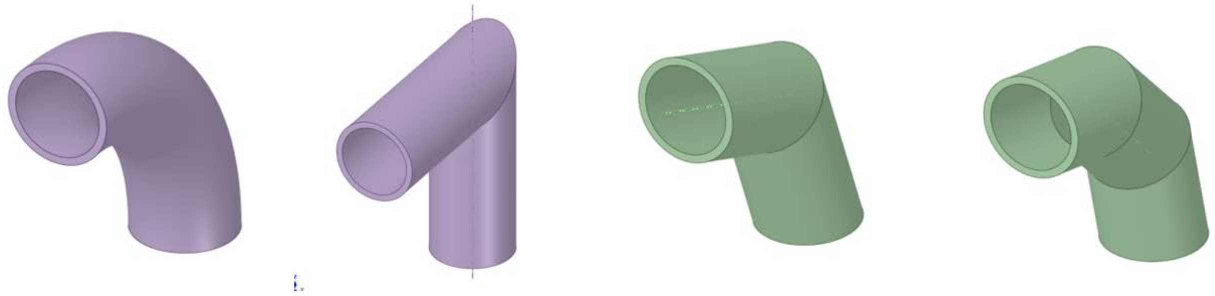
Indeed, the mass, or volume of the solid, is proportional to the length of the revolution profile for the elbow and the length of the resulting cylinders. Being R the bend radius of the elbow and α, the angle between both ends of the torus, the original length of the torus is $L_{torus} = \pi R \frac{\alpha}{180}$ while the simplified one is:

- Method 1: $L_1 = 2R\tan\left(\frac{\alpha}{2}\right) \rightarrow \frac{360}{\alpha\pi}\tan\left(\frac{\alpha}{2}\right) L_{torus}$
- Method 2: $L_2 = nR\sqrt{2\left(1-\cos\left(\frac{\alpha}{n}\right)\right)} \rightarrow n\frac{180}{\alpha\pi}\sqrt{2\left(1-\cos\left(\frac{\alpha}{n}\right)\right)}\, L_{torus}$ where n is the number of cylinders of the resulting geometry (Parameter1).

This implies the following mass discrepancies:

- Method 1: $\Delta_{mass,\%} = 1 - \frac{360}{\alpha\pi}\tan\left(\frac{\alpha}{2}\right)$
- Method 2: $\Delta_{mass,\%} = 1 - n\frac{180}{\alpha\pi}\sqrt{2\left(1-\cos\left(\frac{\alpha}{n}\right)\right)}$

Figure 9, Figure 10 and Figure 11 illustrate an example and the mass discrepancies behaviour in function of α and n.



Original (α =90º)	Parameter1=1	Parameter1=2	Parameter1=3
Mass Diff.	+27%	-3%	-1%

Figure 9 - Mass variation test case

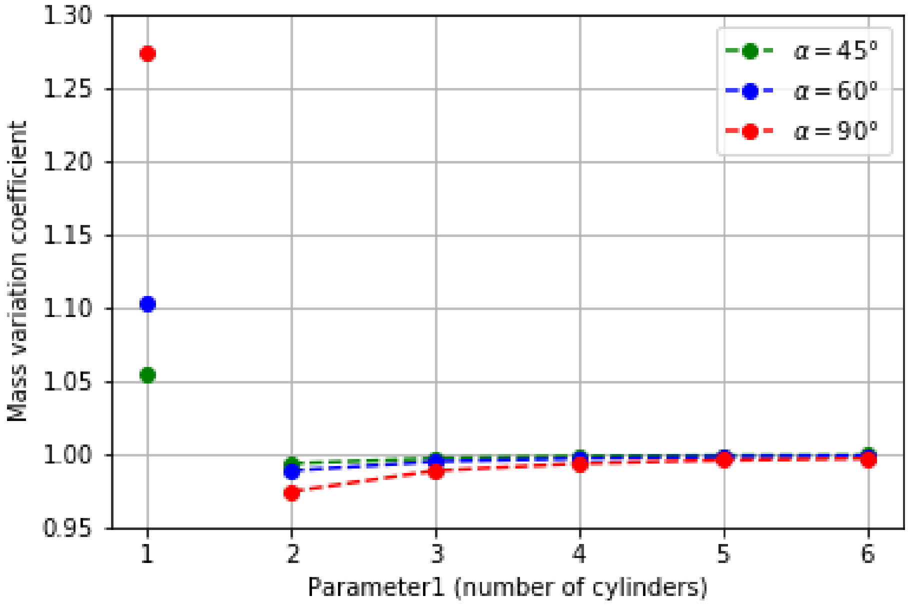


Figure 10 - Mass discrepancy behaviour for both methods

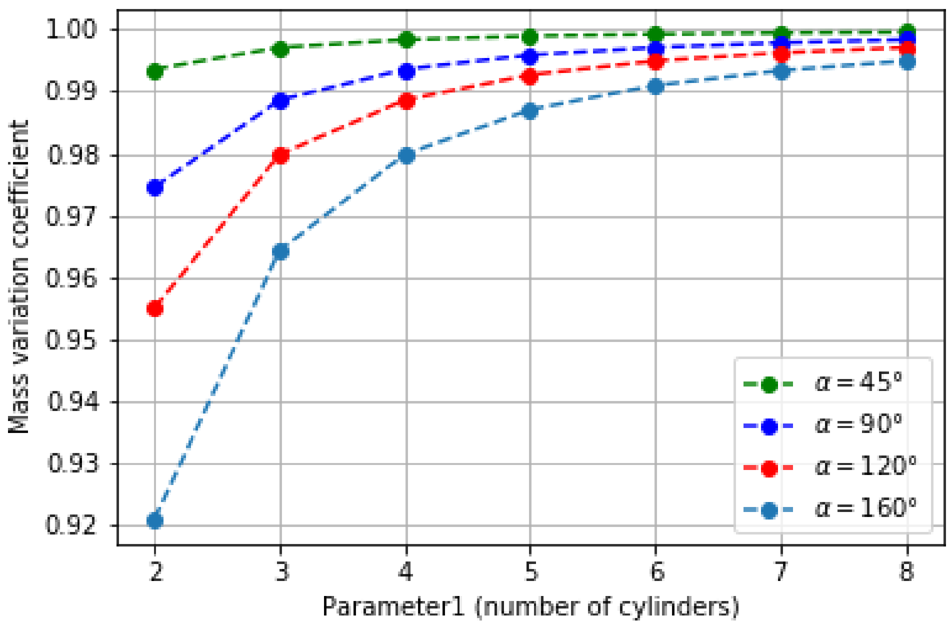


Figure 11 - Mass discrepancy behaviour for method No.2

Complexity variation

The complexity of the simplified geometry depends on the value of the Parameter1 imposed, Table No.1. Whereas, method No.1 creates a geometry description less complex than the original one, method No.2 needs more surfaces but it implies a minor mass discrepancies.

Original	Parameter1=1	Parameter1=2	Parameter1=3	Parameter1=n
5	3	7	9	3+2*n

Table 1 – No. of surfaces needed to describe the geometry