Users manual of GmSAFIR, the graphical pre-processor of SAFIR

1) Table of content

2) Origin

GmSAFIR is a graphic pre-processor used to build input files for SAFIR®.

It has been developed in cooperation between Uliege and Efectis France.

It is based on the open source mesh generator GMSH (https://gmsh.info/).

A lot of information about GMSH can be found here for a general overview (https://gmsh.info/doc/texinfo/gmsh.html#Overview-of-Gmsh) and tutorials can be found here: https://gmsh.info/doc/texinfo/gmsh.html#Gmsh-tutorial .

GmSAFIR is a python executable, adding more functionalities to GMSH, to create your models, to attribute SAFIR related thermo-mechanical properties and constraints to the model, and to write this information in an input file that can be used by SAFIR to run your model.

3) Concepts

3.1 Windows on the screen

The screen is divided in two main windows.

The tool window, with different commands seen on the left in Figure 1. Each command can be expanded or condensed by clicking on the "+" or on the "-"sign that is visible at the left of the command. If too many commands have been expanded and some are not visible, the vertical scroll bar on the right of the window can be used.

The drawing window, in light blue on the right of Figure 1.

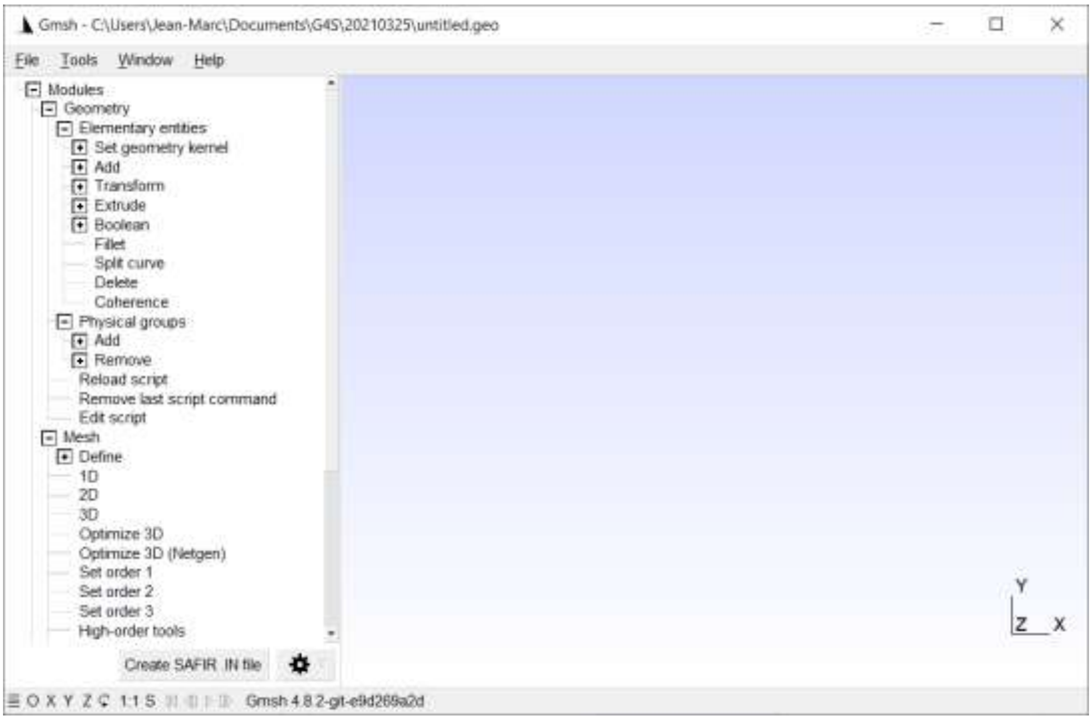

Figure 1: the tool window and the drawing window

There is also a tool bar above the tool window. The scroll down menus "File – Tools – Window – Help" are visible in the tool bar on Figure 1.

3.2 Elementary entities

Elementary entities are the objects at the base of the description of the geometry of the structure to be analysed by SAFIR. Figure 2 shows where the menu of elementary entities is found in the tool window.

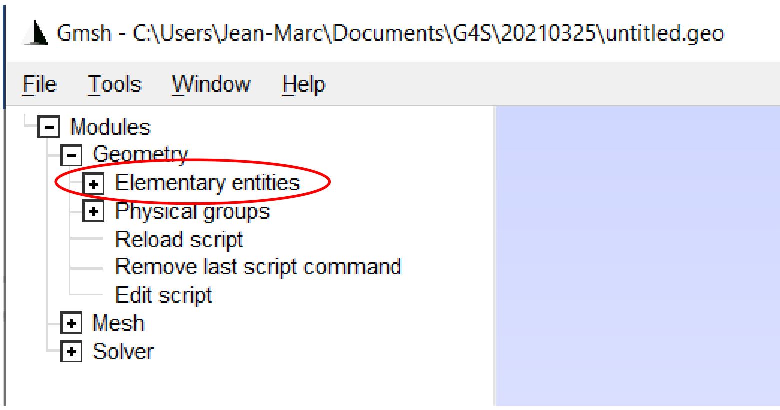

Figure 2: location of "elementary entities" in the tool bar

There are 4 types of elementary entities, with a clear hierarchy between them.

1) Points.
   A point is defined by its coordinates, 2 in 2D models, 3 in 3D models.
   In SAFIR models, points will be used:

- As bases of other elementary entities of higher order (curves);
- As floating nodes of the model if needed (4th node of 3D beam finite elements)
- As support of SPRING finite elements.

2) Curves.
   A curve is defined by its type, by its properties and by the points on which it is built. In our models, we will essentially use:
   - Lines, which are based on the two nodes at the end of the line.
   - Circles, which are based on one node and have the radius as property.

   In SAFIR models, curves will be used:

   - As bases for other elementary entities of higher order (surfaces);
   - As beam or truss finite elements,

3) Surfaces.
   A surface is defined by the curves on which it is based. In our models, we will essentially use:
   - Rectangles, based on 4 lines.
   - Disks, based on a circle
   - More complex surfaces made of differences between simple surfaces. For example, in a 2D thermal analysis, a rectangular concrete section that comprises 2 reinforcing bars is made of:
     - Two disks, one for each reinforcing bar;
     - One complex surface, made of the difference between the rectangle that encompasses the section and the 2 disks.

   In SAFIR models, surfaces will be used:

   - As bases for other elementary entities of higher order (volumes);
   - As slabs, shear walls, steel plates modelled with shell finite elements;
   - In 2D thermal analyses

4) Volumes
   A volume is defined by the surfaces on which it is based. In our models, we will essentially use:
   a. Parallelipedic rectangles, based on 6 surfaces.
   b. Cylinders, based on two disks and the “tube” surface that links the two disks.
   c. More complex volumes made of differences between simple volumes. For example, in a 3D thermal analysis, a timber joint that comprises 2 steel dowels is made of:
      - Two cylinders, one for each dowel;
      - One complex surface, made of the difference between the parallelipedic rectangle that encompasses the joint and the 2 cylinders.

3.3 Physical groups

Physical groups are groups of elementary entities of the same type.

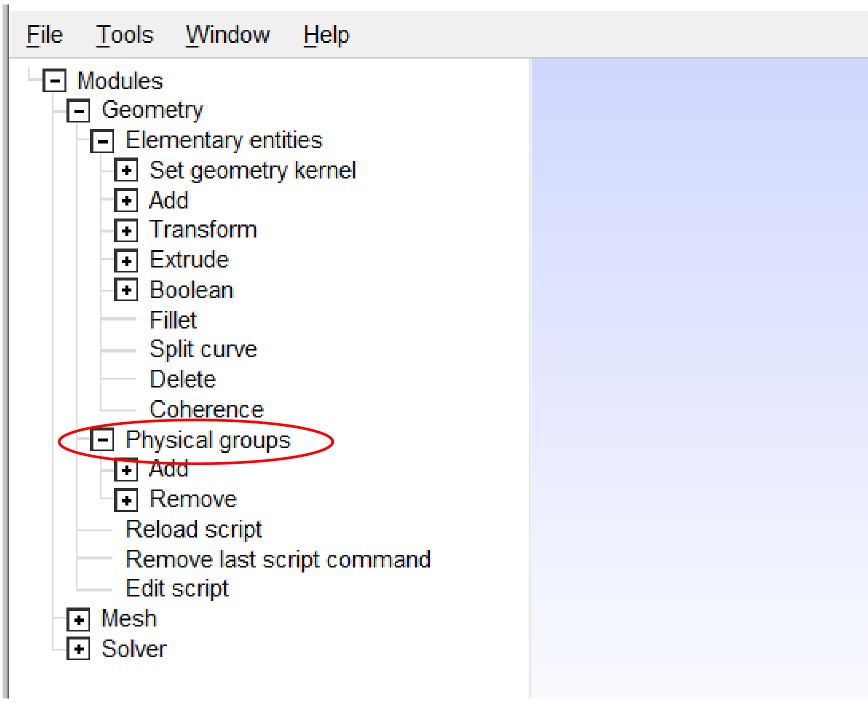

Figure 3: location of "physical groups" in the tool window

A physical group is defined by:

- ✓ Its name, to be chosen by the user.
- ✓ The list of elementary entities that are part of the group.
- ✓ The SAFIR properties[1], with their value, allocated to the group.

The interest of physical groups is that SAFIR properties (see section 1.4) can be allocated to a physical group, which will allocate this SAFIR property to all elementary entities of the group. If this SAFIR property is deleted or modified for the physical group, it is deleted or modified for all elementary entities of the group. It is also possible to add an entity to the physical group and this entity will receive the SAFIR properties defined for this group.

3.4 Entities and groups Versus F.E. objects

It is essential, conceptually speaking and also for comprehension of this manual to clearly distinguish between elementary entities (and physical groups), on one hand, and Finite element objects, on the other hand.

- ➢ Elementary entities exist in the GMSH model.
- ➢ Finite element objects exist in the SAFIR model, after meshing the GMSH model.

The elementary entities and the physical groups are used as generators of F.E. objects created through the meshing process; each cell of the meshing becomes a F.E. object which inherits the SAFIR properties from its mother elementary entity or physical group.

[1] See Section 2.5

We will thus distinguish between:

- "points" which are elementary entities used, for example, to be the end of a curve
- "F.E. NODES" which can, for example, be all the nodes of the SAFIR model after discretisation of the curve.

Similarly, we will use the concept of "BEAM F.E." for the finite elements resulting from the discretisation of a curve, "SHELL F.E." or "2D SOLID F.E." for the finite elements resulting from the discretisation of a surface, "3D SOLID F.E." for the finite elements resulting from the discretisation of a volume.

3.5 SAFIR properties

SAFIR properties are any information that can be added to elementary entities or to physical groups to be used in the SAFIR simulation.

Typical SAFIR properties are (for elementary entities or physical groups):

- For points: node loads, node masses, support conditions, restrain for torsion, "SAME" conditions with another point…
- For curves: Distributed loads on the F.E. BEAMS of this curve, distributed mass on the F.E. BEAMS of this curve, section type of the F.E. BEAMS of this curve, "SAME" condition between all F.E. nodes of the curve, thermal condition in a 2D thermal analysis….
- For surfaces: distributed load on the SHELL F.E. of this surface, distributed mass on the SHELL F.E. of this surface, section type of the SHELL F.E. of this surface, material type of the surface in a thermal analysis, "SAME" conditions for all F.E. nodes of the surface, thermal condition in a 3D thermal model…
- For volumes: material type of the SOLID F.E. of this volume….

4) Flowchart of a first model

4.1 File management

Immediately after being opened, GmSAFIR must be indicated the folder in which information will be found or will be created, and thereafter saved automatically as the model is being created or modified in the GUI.

Information about an existing model is indicated by clicking on "File/Open" whereas a new model is created by clicking on "File/New".

Important note: Any other action made before "File/Open" or "File/New", such as, for example, changing the "Problem Type" may close the code or, at best, issue a warning message.

If a new problem is to be created, the command "New" in the scroll down menu 'File" of the tool bar allows selected the folder in which the files created will be located.

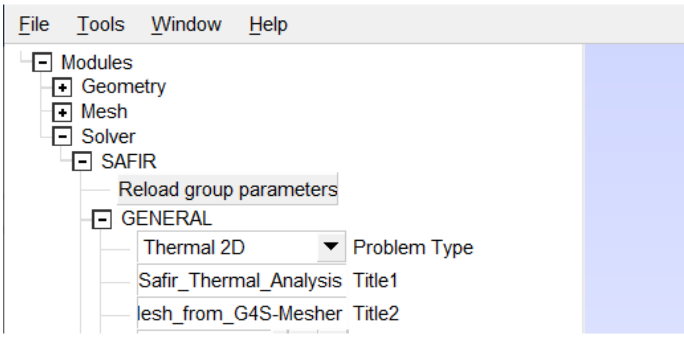

Figure 4: the "file" scroll down menu

The user is asked to give a file name of the kind "*name.geo*". This file will be created in the selected folder. It will contain information needed by GMSH to describe the geometry of your structure.

A similar file will automatically be created in the same folder with the name "*name.g4s*". It contains the SAFIR properties that are added to the geometry, see SAFIR properties.

If the user fails to select the appropriate folder by the "File/New" command, the model that will be created will be stored in the files "*untitled.geo*" and "*untitled.g4s*" that will be created in the same folder as the executable "*GmSAFIR.exe*". It will be possible to rename these two files and move them to the appropriate folder thereafter, but this implies file manipulations. It is perhaps good practice to create a specific and recognisable simple model and to leave it as "*untitled*" in the folder of "*GmSAFIR*". This will lead to the fact that this recognisable project will appear in the drawing window of GmSAFIR when the software is started, which will remind the user to either open another existing project with the "*File/Open"* command or to create a new one with the "*File/New"* command.

The two .GEO and .G4S files will not be used by SAFIR. The input file to be used by SAFIR will be created when the model is finished by clicking on "Create SAFIR .IN file" at the bottom of the tool window, see Figure 5. The SAFIR .IN file will be created in the same folder as the .GEO and the .G4S files, but it will have its own name as given in the window "Name of the .IN file" in the tool window, see Figure 5

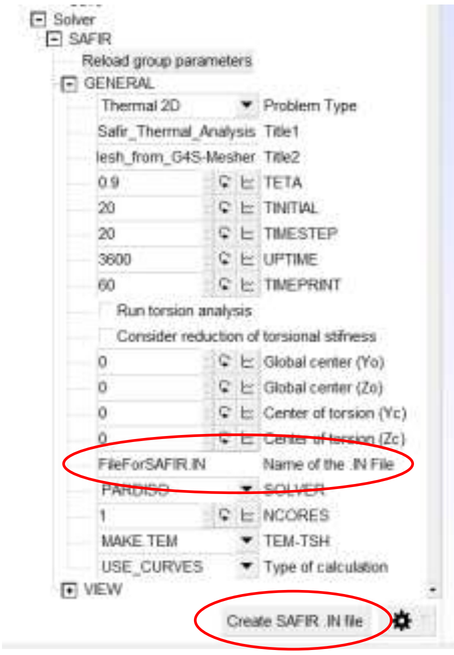

Figure 5: creating the SAFIR input file

The user who is familiar with most software used in the Windows environment such as Waord or Excel may be disoriented by the fact that there is no command such as “*save*” or “*save as*” in the “*file*” scroll down menu on top of the tool window.

This is because each command introduced that describes the geometry is directly written in the .GEO file and this file is saved automatically after each command. Figure 6, for example, shows the content of the GEO file that resulted from the creation of the 2 rectangles shown on the right hand side of the Figure. This file is called a *script* in GMSH langage.

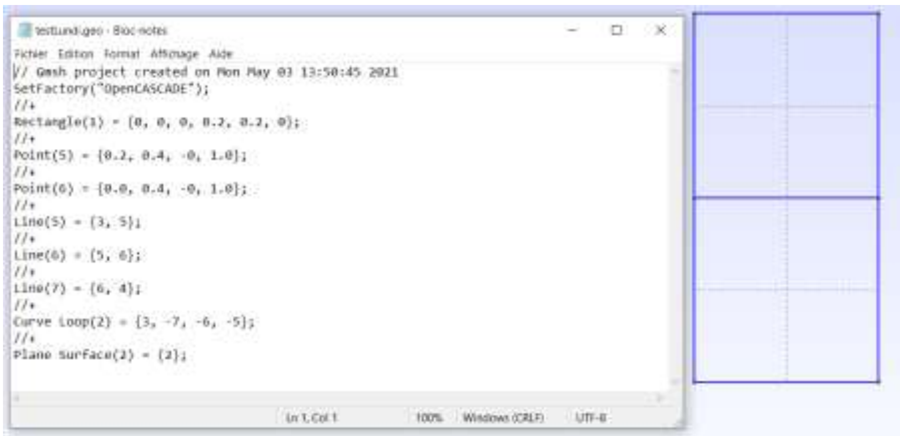

Figure 6: example of a script

The script of the GEO file can be opened and any time by the “*Edit script*” command in the tool window, see Figure 7. When it is open, the script can be verified and, most important and useful, it can be modified in the editor where it has been opened automatically (such as *nodepad*, for example). If the modifications are saved when closing the editor, the "*Reload script*" command will update the geometry of the model in GmSAFIR[2], see Figure 7.

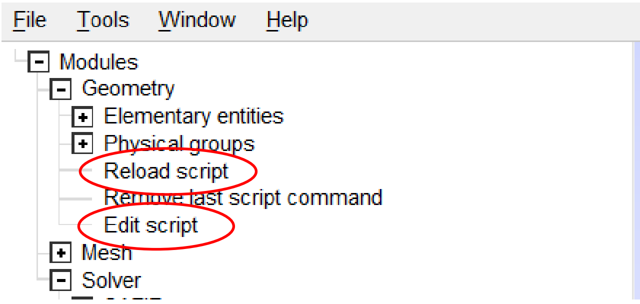

Figure 7: Script related commands

The SAFIR properties are also directly written in the .G4S file which is saved automatically after every entered command. Edition of this file is also possible, but not from within the GmSAFIR environment.

The operation equivalent to *"save as"*, if it has to be performed, must be done from outside the GmSAFIR environment, for example in the File Manager of Windows where the GEO and the G4S files can be copied and saved under a new name.

4.2 Initiation

It is good practice now to select the type of problem that will be solved, related to the type of model that will be created.

This is done in the tool window under "Modules/Solver/SAFIR/GENERAL/Problem type", see Figure 8.

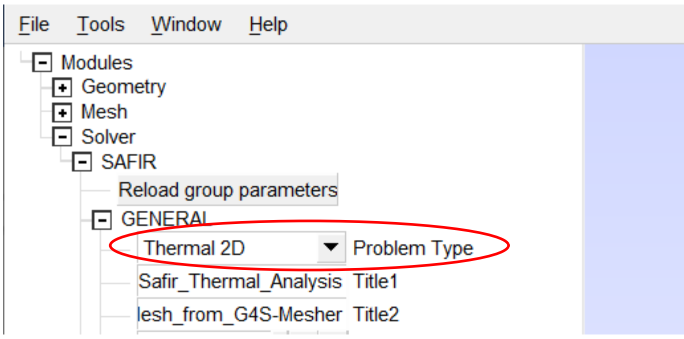

Figure 8: problem type command

The choice is between:

- Thermal 2D
- Thermal 3D
- Structural 2D
- Structural 3D

[2] Smart readers will immediately see the opportunity to write parametric models with any homemade code that can create the sciripts of the GEO files in the appropriate format.

Thermal analyses made for the cross section of SHELL F.E. will be found under “Thermal 2D”.

Torsion analyses of 2D sections for BEAM F.E. will be found under “Thermal 2D”.

4.3 Creating the geometry

4.3.1 In GmSAGIR

The geometry of the model can now be created using the different tools that are available under “Modules/Geometry/Elementary entities/Add”, see Figure 9.

The simplest example given here is for a “Thermal 2D” problem.

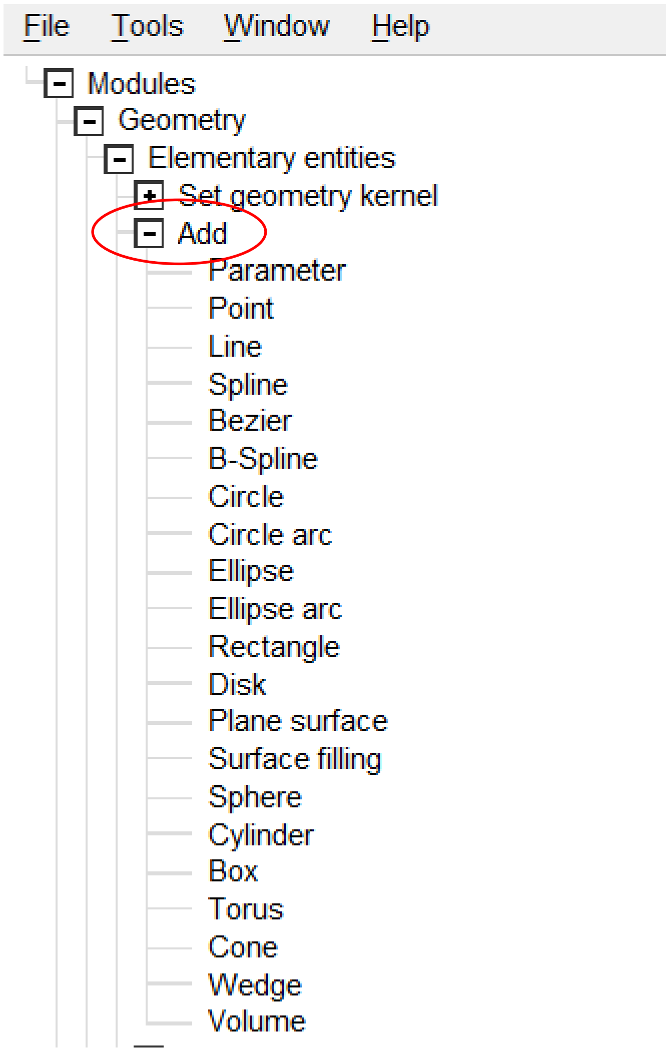

Figure 9: Add elementary entities

Different elementary entities can be added. In the most general way, points are first added; lines can be constructed on two points; plane surfaces can be added from lines which make a closed surface.

- Points added with “Add/Point” can be introduced in any order. The order has no consequence.
- Lines added with ‘Add/Line” request the two points on which the line is built. The order in which these points are given is important as it defines a direction for the line, from point 1 to point 2.

- Using “Add/Plane surface” requires some attention. For SAFIR, the surface has indeed to be described in counterclockwise direction. The first line that is selected to describe the surface must be a line that has the appropriate direction: the surface must be on the left if you imagine that you walk on the line from its point 1 to the point 2. How the lines are used to create the surfaces is illustrated by the example of Figure 10. On the top left are shown the 7 lines that have been created. They have been created as such:

  Line 3: from point 3 to point 4
  Line 5: from point 3 to point 5
  Line 6: from point 5 to point 6
  Line 7: from point 6 to point 4

  If you select “Add/Plane surface”, you are asked to select the first line of the surface. If you select line 5, GMSH understands that the surface must comprise also lines 6 and 7 and they are selected, see top right. At this stage, GMSH cannot guess whether the surface will be closed by line 3 or by lines 4, 1 and 2 and is therefore waiting for you to select the next line. If you select line 3, the result is shown on the bottom left of Figure 10.

  If we have a look on the script, we see the 2 following lines.

  ```
  //+
  Curve Loop(1) = {5, 6, 7, -3};
  ```

  It shows that the lines that define the surface are lines 5, 6, 7 and 3, in counter clockwise direction. Yet, line 3 has been inverted (see “-3”) because it has been defined originally from node 3 to node 4.

  If we start defining the surface by line 3, see bottom right in Figure 10, GMSH does not know whether our surface is based on lines 3, 7, 6 and 5 or on lines 3, 4, 1 and 2. Selecting the line 5 will close the surface based on the upper rectangle (same visual result as bottom left in Figure 10) but the script will now show:

  ```
  //+
  Curve Loop(1) = {3, -7, -6, -5};
  ```

  Note that the lines are defined in the order 3, 7, 6, 5 i.e. in clockwise order, as dictated by the direction of the first line selected, here line 3. Lines 7, 6 and 5 are inverted by the sign “-“ to revert their original direction. This surface will not be meshed properly for SAFIR.

  As a conclusion, the direction of the surface is dictated by the direction of the first line that is selected and not by the order in which the lines are selected.

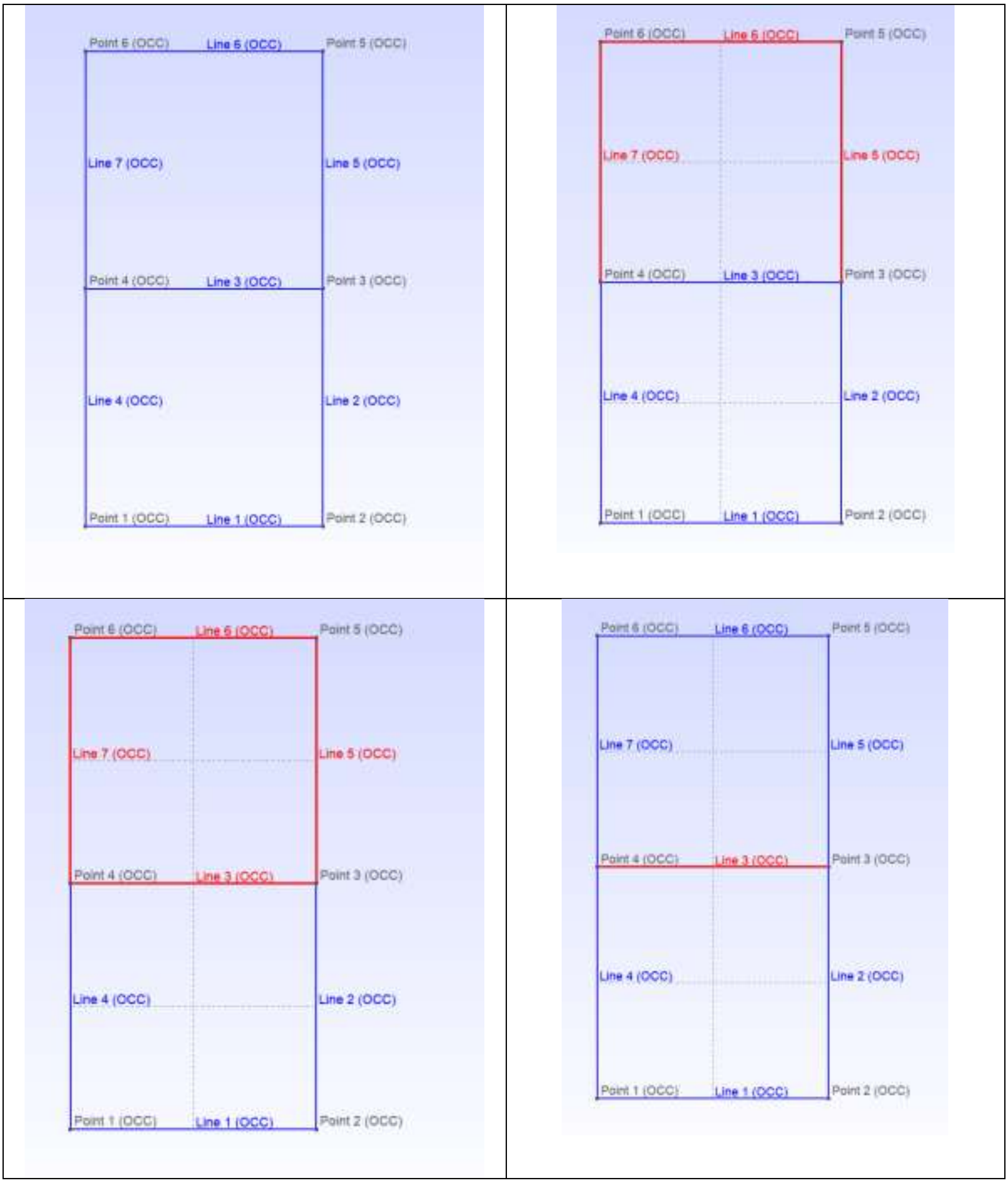

Figure 10: creation of surfaces

It is possible to construct regular surfaces directly, for example by the “Add/Rectangle” command. The rectangle created in this way will comprise the 4 lines and the 4 points on which it has been based. From what we could see, the direction of the lines that make the rectangle are counter clockwise. This is in fact how the rectangle based on lines 1, 2, 3, and 4 has been created in Figure 10 and this is why line 3 was defined from point 3 to 4.

A very important concept is the coherence of the model. A model is coherent when all surfaces are based on common lines at the interface. Two adjacent rectangles created by the “Add/Rectangle” command may look as adjacent in the drawing window but the 2 lines at the interface are not the same; these tow rectangles are not adjacent numerically speaking and, for example, heat will not circulate from one to the other. If coherence between the two surfaces is not ensured and, for example, the line 2 in Figure 10 is exposed to the fire, the isotherms will have the pattern shown in Figure 11.

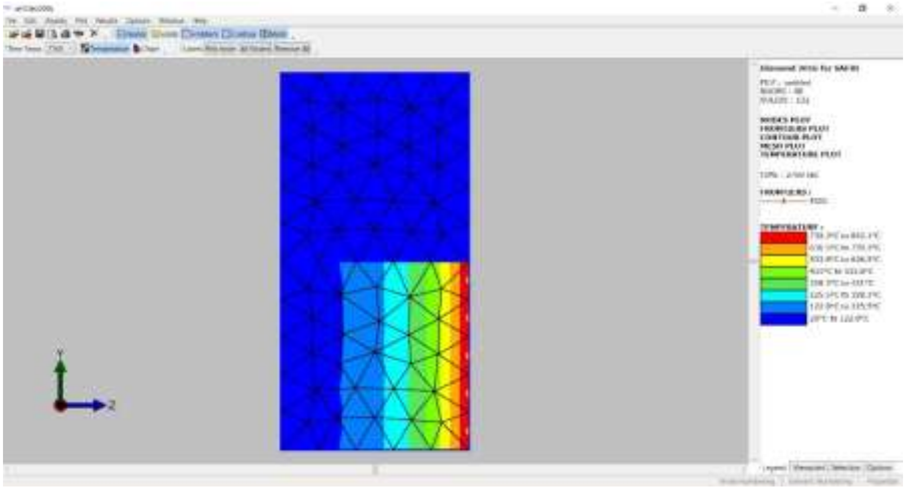

Figure 11: isotherms in two rectangles that are not coherent

The correct way to build adjacent surfaces is to create each of them with the "add/Plane surface" command, taking care of the direction of the first line that defines each surface. It is indeed possible to create adjacent rectangles by "Add/Rectangle" command and then make them coherent by the command "Coherence", see Figure 12.

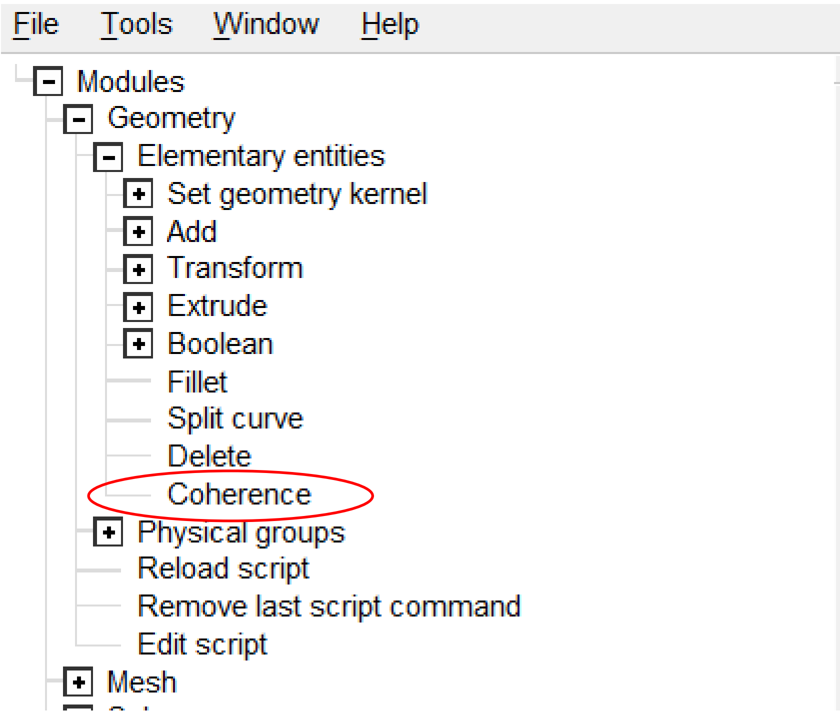

Figure 12: the "Coherence" command

This will merge the two lines which are superposed into one (and the same for the two pairs of superposed points) and the model will be coherent. Yet, there is no control as to the fact that the surfaces are described clockwise or counter clockwise.

4.3.2 In Autocad

The geometry can also be created with AUTOCAD.

1) Create the geometry of the model in Autocad using only points and lines. If you are using an existing Autocad model, clean it by erasing any information other than lines.
2) Export the model from Autocad in ".iges" or ".stp" format.
3) Open this file in GmSAFIR.
4) Convert the file in a new ".geo" file (Geometry / Elementary Entities /Set geometry kernel / OpenCASCADE); answer to the popup windows "Create .geo".
5) You can now continue by creating eventual surfaces, defining physical groups, meshing, etc.

If you have a file in the DWG format, you can convert it to a DXF format, for example in the free website CloudConvert.com. You must choose the ASCII DXF format, not the binary DXF Format.

1) Enter the full path (dirname+filename) of your DXF file in the textbox named « DXF File », see Figure 13, and, before quitting the textbox, you type « Enter » (which allows GmSAFIR to effectively know that the name was changed, otherwise the next step will not work properly).

Figure 13: enter the complete file name

2) Press the button « Convert DXF ». The GEO file will be created in the same directory of DXF file, with simply changing the filename extension to .GEO.
3) After conversion has been successfully done, see Figure 14 you need to open the .geo file in GmSAFIR.

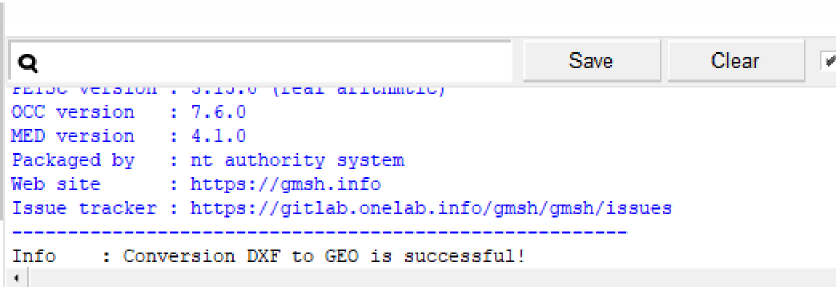

Figure 14: message window

4.4 Mesh

Module

To perform the mesh generation, go to the mesh module (by selecting 'Mesh' in the tree) and choose the dimension:

'1D' will mesh all the lines;

'2D' will mesh all the surfaces, as well as all the lines if '1D' was not called before;

'3D' will mesh all the volumes, and all the surfaces if '2D' was not called before.

It is possible to refine the mesh by clicking on "Refine by splitting".

This is the fastest way to generate a mesh and refine it, but it is here not possible to impose a size on the elements.

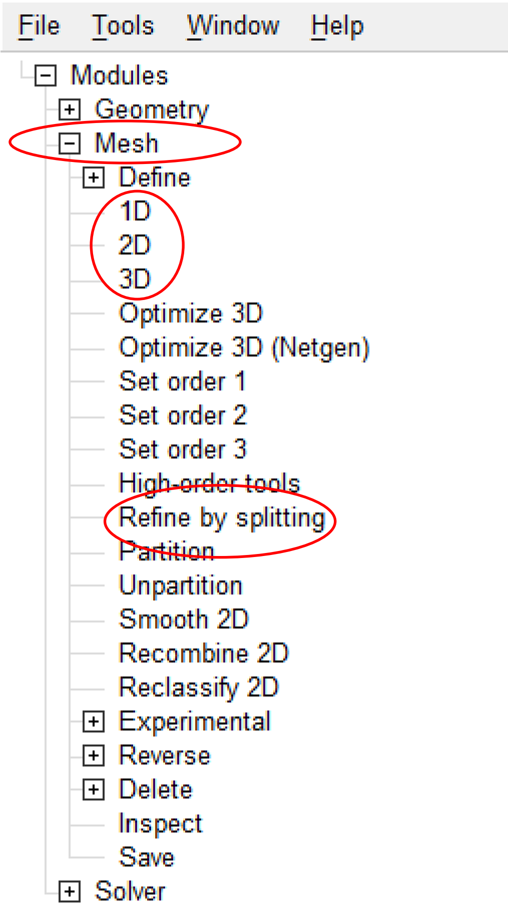

Figure 15: the "mesh" commands

Mesh size

The first method to define the size of the element is via the fourth coordinate of a node which is added via the command “Modules/Geometry/Elementary entities/Add/Point”, see “Prescribed mesh size at point” in Figure 16.

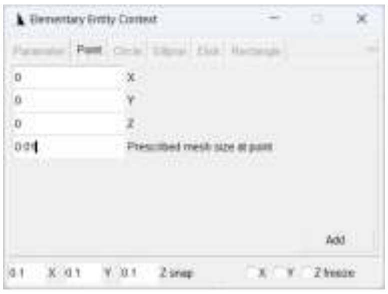

Figure 16: size of the elements near a point

Another possibility is to use the command “Module/Mesh/Define/Transfinite. Applying it to the curve which is the boundary of a circle (a reinforcing bar, for example) with 7 points will force this boundary to be made of 6 lines, thus representing the bar by the *inscribed* hexagon, see Figure 17

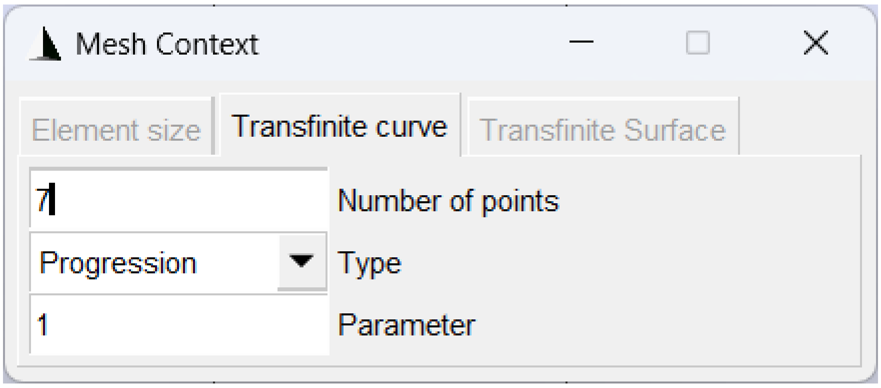

Figure 17: transfinite the curve

The inscribed polygon has a surface which is smaller than that of the circle, and this may be a problem if that circle is representing a steel rebar in a concrete section; some steel will be missing. If the objective is to have the surface of the inscribed polygon equal to that of a circle of diameter *D*, it is recommended to define, in the GmSAFIR model, a circle surface with a modified diameter $D^* = c\ D$, with the modification parameter *c* depending on the number of the number of sides of the inscribed polygon *n*, see Figure 18

n	*c*
4	1.253
5	1.149
6	1.100
8	1.054
10	1.034
12	1.023
14	1.017

Figure 18: Table for the equivalent diameter D^*

To impose the size on the elements of the mesh on the whole domain, go to Tools and click on Options. In the Options window, go to mesh and select the General tab. A maximum and minimum element size can be defined in the "Min/max element size" line, see Figure 19.

This size criteria will be imposed on the entire model.

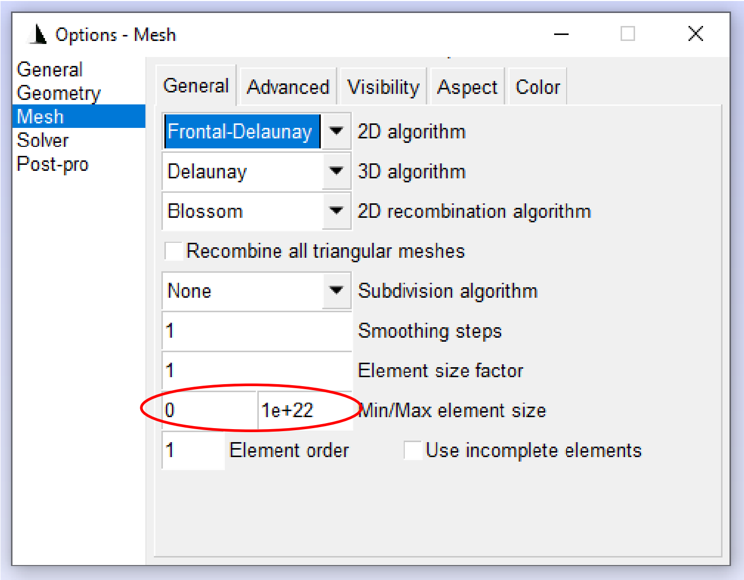

Figure 19: Min/Max element size

The same effect can be obtained by the following lines written in the script.

```
//+
Mesh.MeshSizeMin=0.0;
Mesh.MeshSizeMax=0.05;
//+
```

Advanced options

To impose element sizes on certain parts (surfaces, volumes, points, etc.), it is necessary to go through the "Fields". Follow these steps:

1) Go to Mesh in the tree ➔ Define ➔ click on Size fields
2) In the window, click on New and choose Constant. You must fill the following parameters:
   - Vin : size inside the model
   - Vout: size outside the model (you can write a default value as 10)
   - List of the surfaces, points or volumes that are affected by the size given:

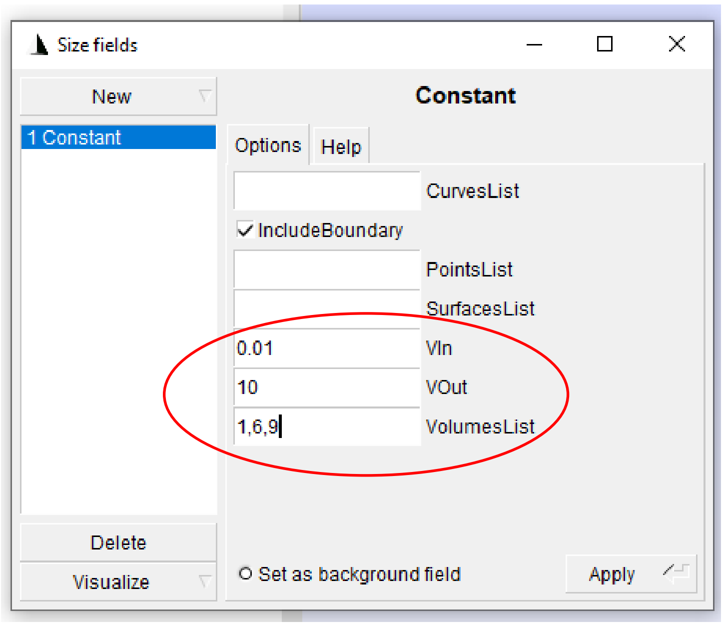

Figure 20: size of the elements in selected parts of the model

3) Repeat the previous operation for each list of parts to which you want to assign a specific size.
4) To finish, click on New and choose Min. You must write the list of the Fields you generated. Select "set as background field", see Figure 21.

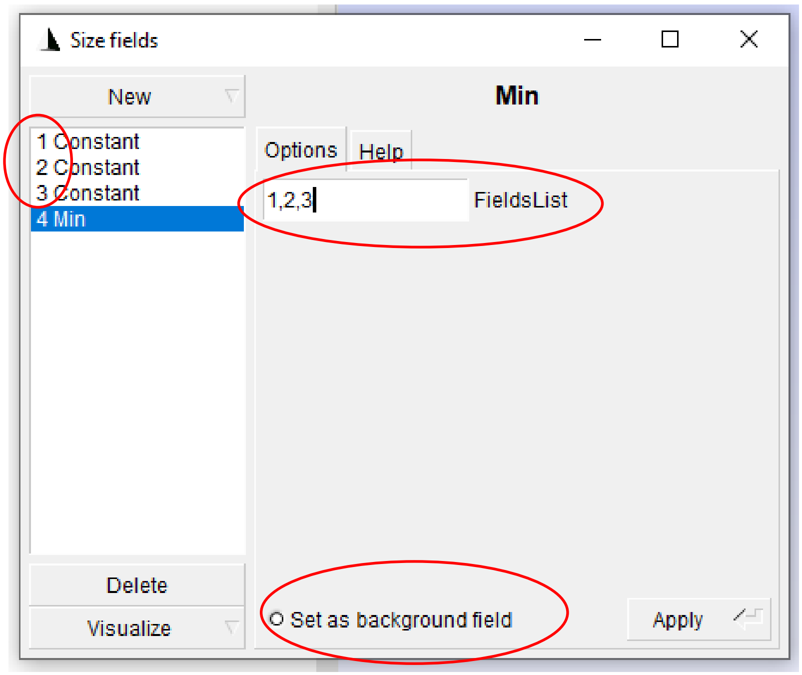

Figure 21: Set as background field

5) Click on Apply and close.

In the script, the operation would be performed by the following lines:

```
//+
Field[1] = Constant;
Field[1].VolumesList = {2,3,5};
```

```
Field[1].IncludeBoundary = 1;
Field[1].VIn = 0.05;
Field[1].VOut = 10;
//+
Field[2] = Constant;
Field[2].VolumesList = {4,6};
Field[2].IncludeBoundary = 1;
Field[2].VIn = 0.05;
Field[2].VOut = 10;
//+
Field[3] = Constant;
Field[3].VolumesList = {1,5,7,8,9};
Field[3].IncludeBoundary = 1;
Field[3].VIn = 0.05;
Field[3].VOut = 10;
//+
Field[4] = Min;
Field[4].FieldsList = {1, 2, 3};
//+
Background Field = 4;
//+
```

4.5 VOID and SYMVOID

In 2D models

The contour of a VOID is defined by a physical group of the type “curve”. This curve is made of several lines.
If the VOID is not closed, this is probably because the model represents one half, or one quarter of a bigger object and symmetry has been used to reduce the size of the mode. In this case, SYMVOID commands must be inserted in the input file. This is done in GmSAFIR by applying the property “Void SymAxis Constraint” to an elementary entity of the type “line”.
Figure 22 shows a rectangular hollow section of which only one half has been considered in the model. The “Void Boundary” property has been applied to the physical group which is made of 3 lines, whereas the “Void SymAxis” property has been applied to the elementary line 1 which is visible on the right side of the model.

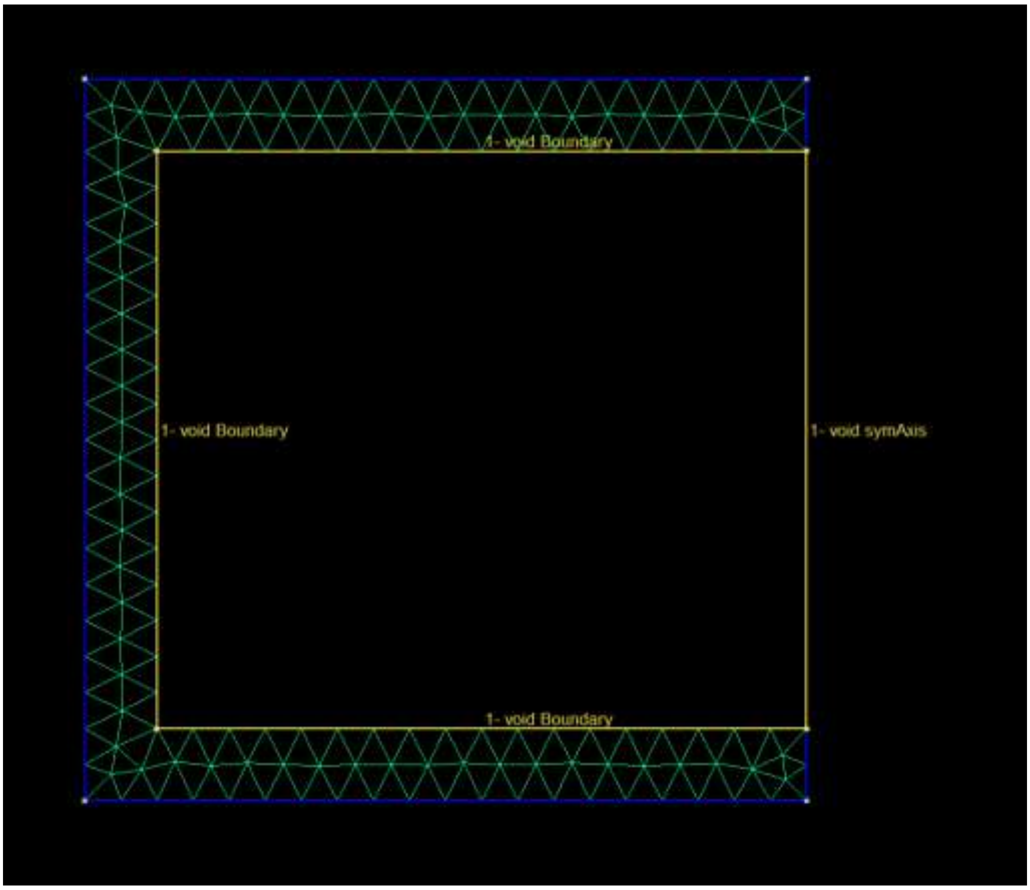

Figure 22: VOID and SYMVOID

In 3D models

The contour of a VOID is defined by a physical group of the type “surface”.
If the VOID is not closed, this is probably because the model represents one half, or one quarter of a bigger object and symmetry has been used to reduce the size of the mode. In this case, SYMVOID commands must be inserted in the input file. This is done in GmSAFIR by applying the property “Void SymAxis Constraint” to an elementary entity of the type “plane surface”.