GiD User Manual

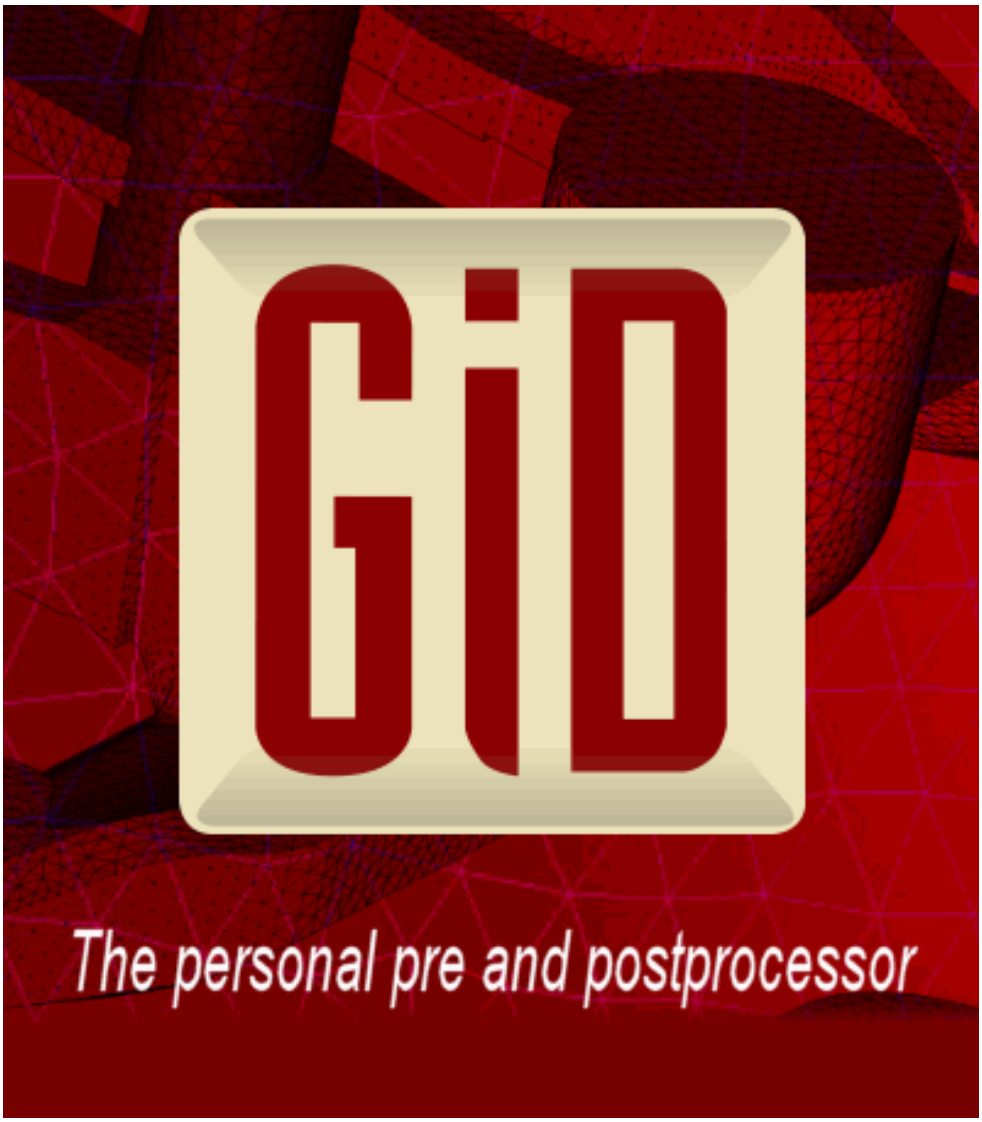


Table of Contents

1 PRESENTATION OF GiD

This chapter will introduce the user to the user-interface and graphic environment of **GiD**.

GiD is a general purpose pre-postprocessor for computer analysis.

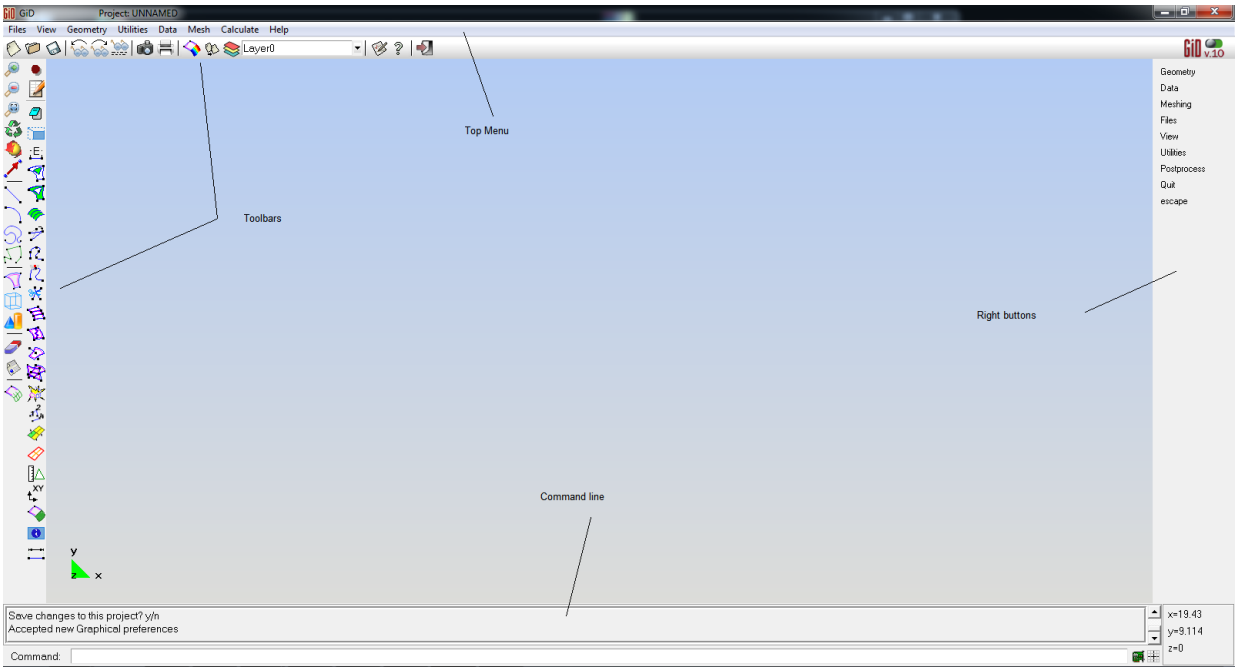
All the data, geometry and mesh generation can be performed inside. Also, the visualization of all types of results can be performed.

It can be adapted to a specific analysis module by the creation of a 'problem type'.

Typical problems that can be successfully tackled with GiD include most situations in solid and structural mechanics, fluid dynamics, electromagnetics, heat transfer, geomechanics, etc. using finite element, finite volume, boundary element, finite difference or point based (meshless) numerical procedures.

1.1 User interface

Upon opening **GiD,** the following window appears on the screen:



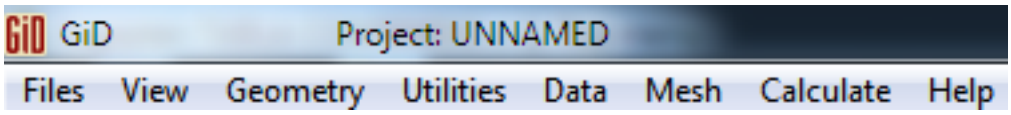
To change the configuration of toolbars and menus, use the toolbars option, located in **Utilities->Tools->Toolbars.**

1.2 Top menu

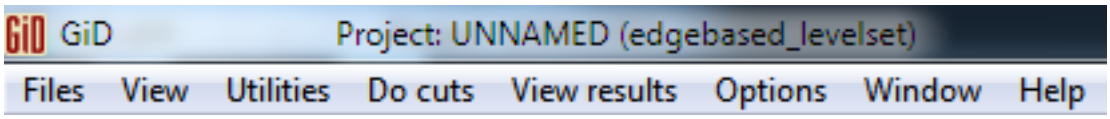
The **Top Menu** offers various types of commands.

It is important to note that these options will differ depending on the whether the user is performing a preprocessing or postprocessing analysis, and that the options needed in each case differ as well.

Two possible configurations of the **Top Menu** are presented below:



And in the postprocessing phase:

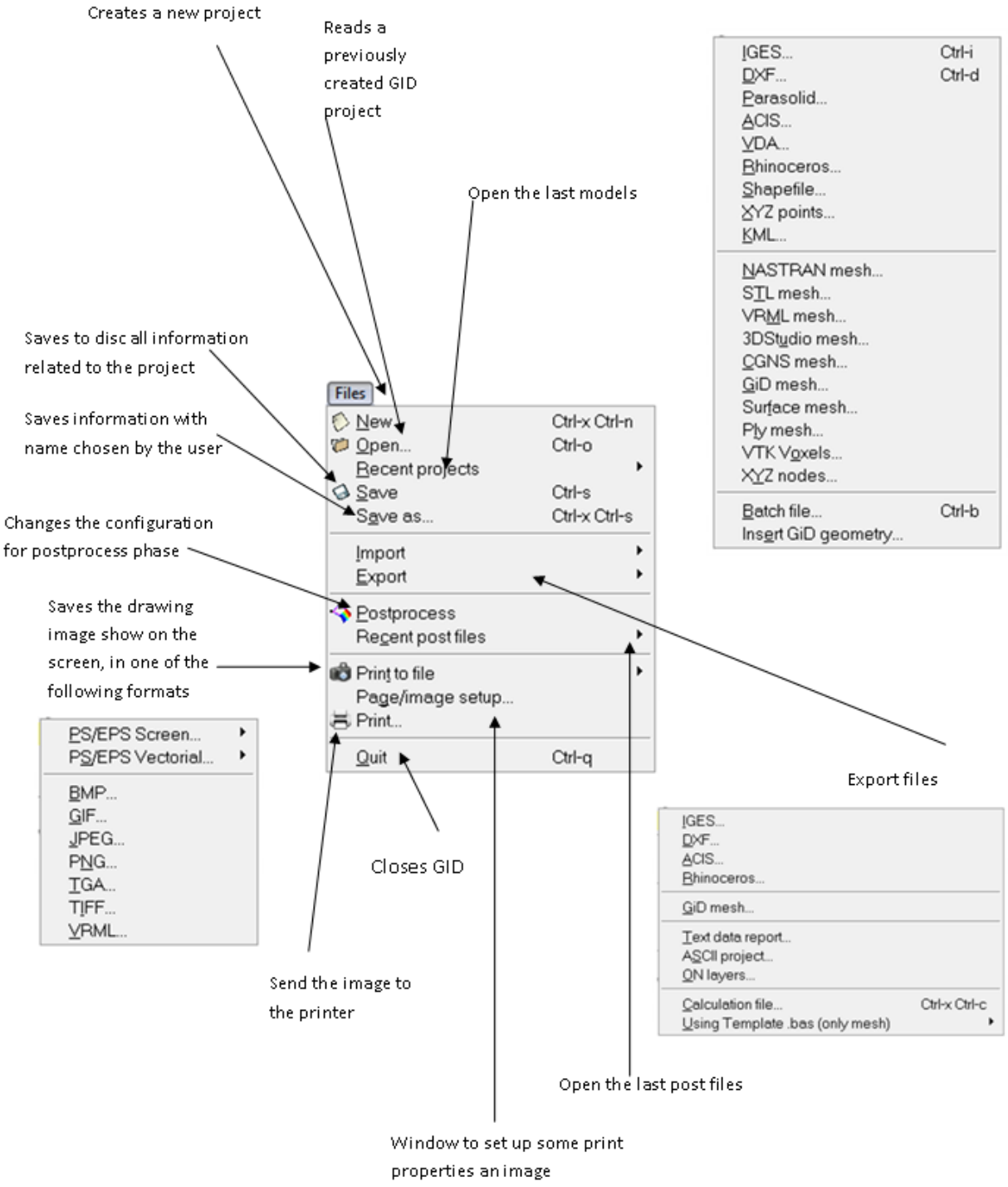


These two options will be presented in more detail later.

Next, each drop-down menu in the **Top Menu** will be described in detail.

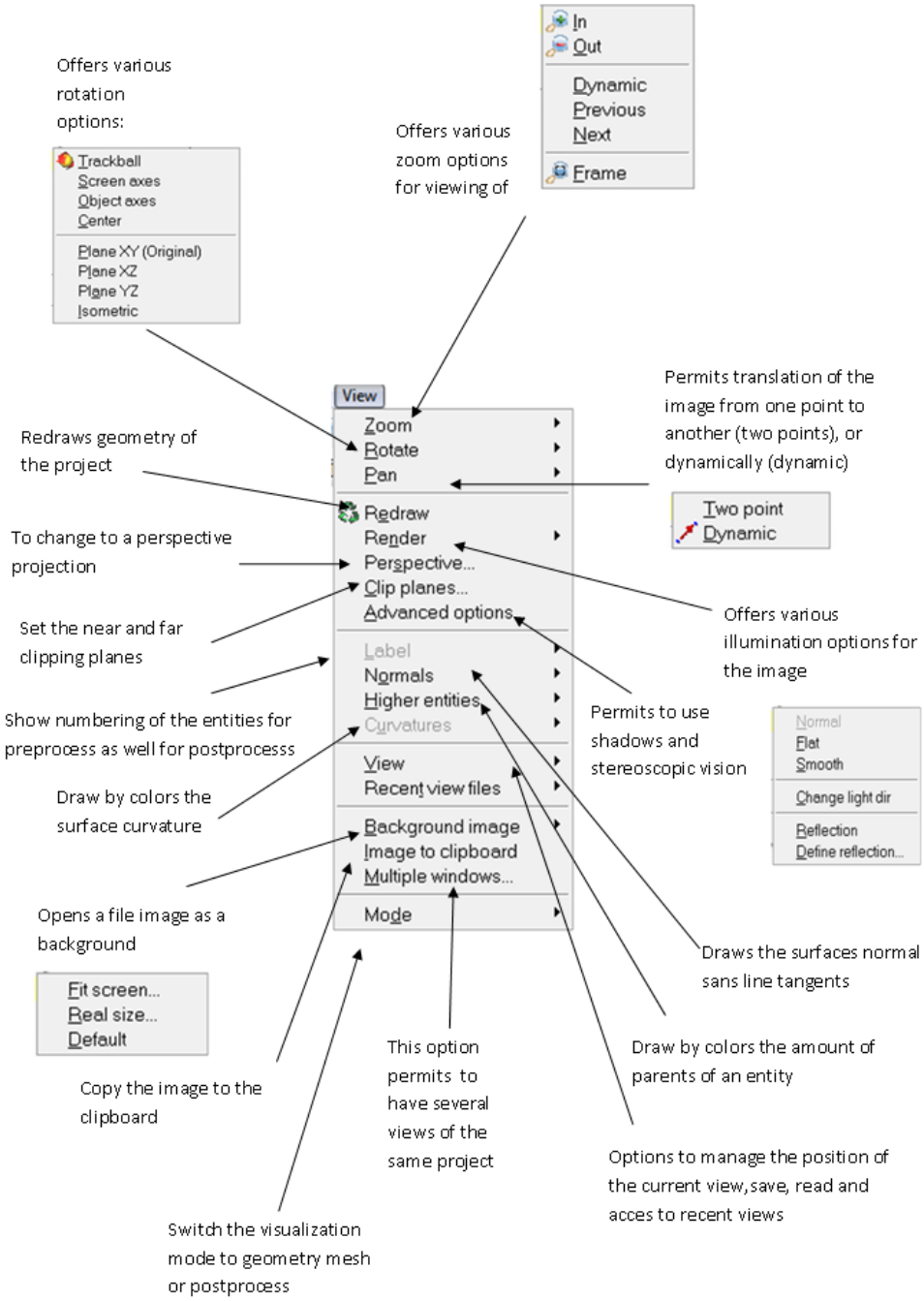
1.2.1 Files-Pre

Two main types of functions can be controlled in this menu: 1) the handling of files (i.e. create, read, save, etc.) of **GiD** projects;and, 2) the importing and exporting of files.


Creates a new project
Reads a previously created GID project
Open the last models
Saves to disc all information related to the project
Saves information with name chosen by the user
Changes the configuration for postprocess phase
Saves the drawing image show on the screen, in one of the following formats
Closes GID
Send the image to the printer
Open the last post files
Window to set up some print properties an image
Export files
Files
New Ctrl-x Ctrl-n
Open... Ctrl-o
Recent projects
Save Ctrl-s
Save as... Ctrl-x Ctrl-s
Import
Export
Postprocess
Recent post files
Print to file
Page/image setup...
Print...
Quit Ctrl-q
PS/EPS Screen...
PS/EPS Vectorial...
BMP...
GIF...
JPEG...
PNG...
TGA...
TIFF...
VRML...
IGES... Ctrl-i
DXF... Ctrl-d
Parasolid...
ACIS...
VDA...
Rhinoceros...
Shapefile...
XYZ points...
KML...
NASTRAN mesh...
STL mesh...
VRML mesh...
3DStudio mesh...
CGNS mesh...
GiD mesh...
Surface mesh...
Ply mesh...
VTK Voxels...
XYZ nodes...
Batch file... Ctrl-b
Insert GiD geometry...
IGES...
DXF...
ACIS...
Rhinoceros...
GiD mesh...
Text data report...
ASCII project...
ON layers...
Calculation file... Ctrl-x Ctrl-c
Using Template .bas (only mesh)

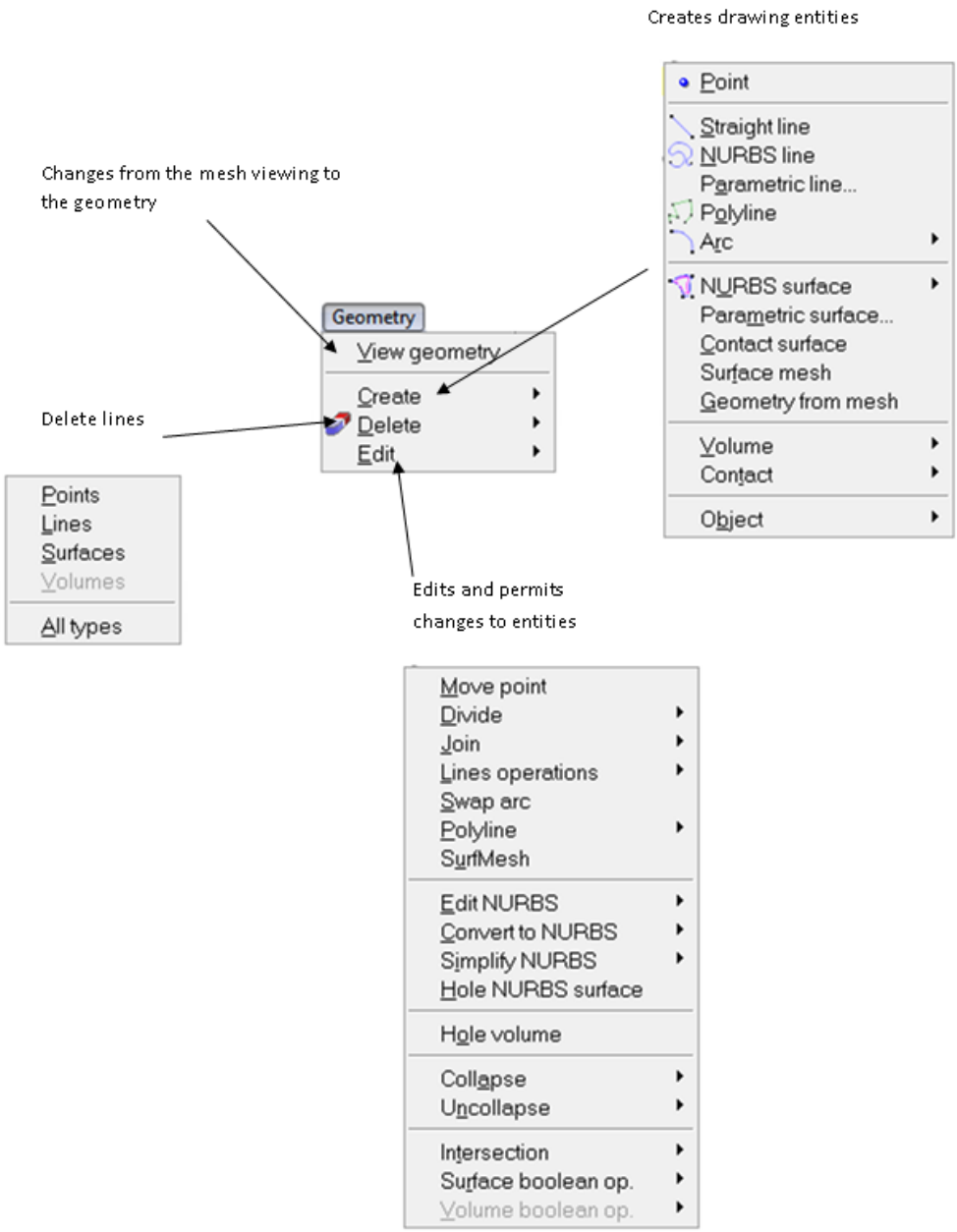
1.2.2 View

In the view menu (also available from the mouse menu) there are all the visualization commands. These commands change the way to display the information in the graphical window, but they do not change any definition of the geometry or any other data.



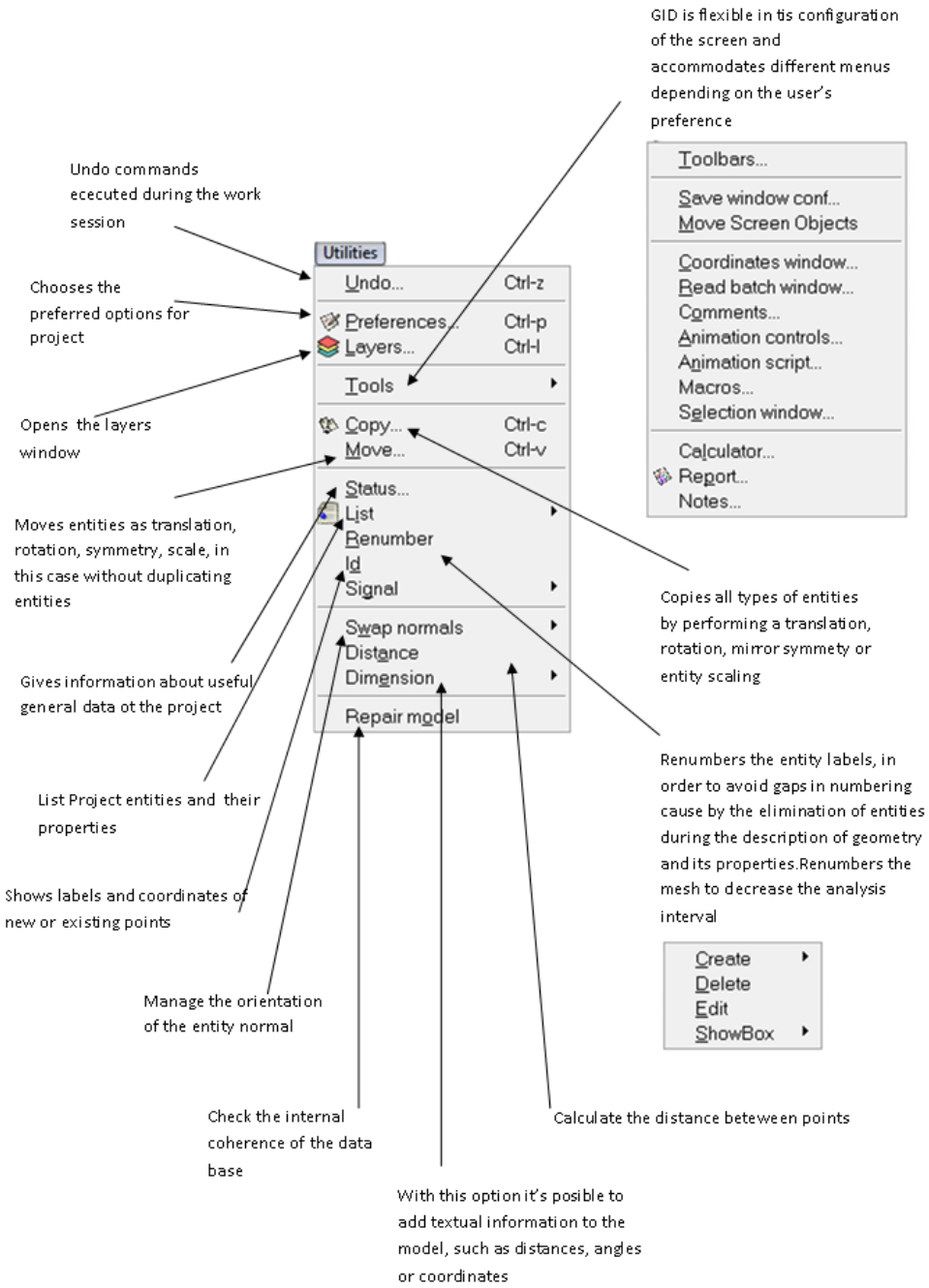
1.2.3 Geometry

Geometry permits the user to create, delete, edit and model geometry.



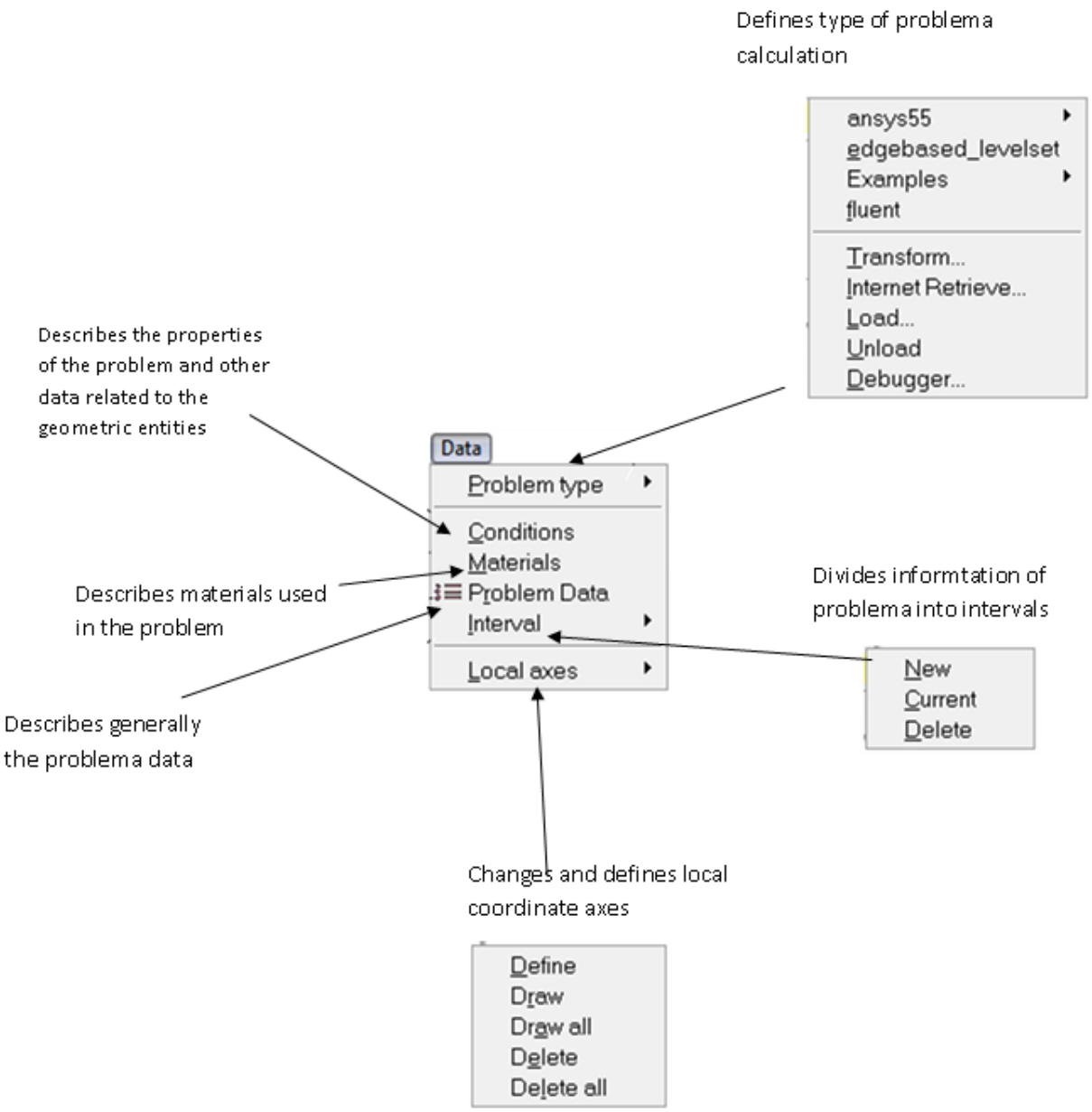
1.2.4 Utilities.Pre

In the **Utilities** menu, **GiD** allows the user to define preferences or perform operations on both the geometry and the mesh entities.



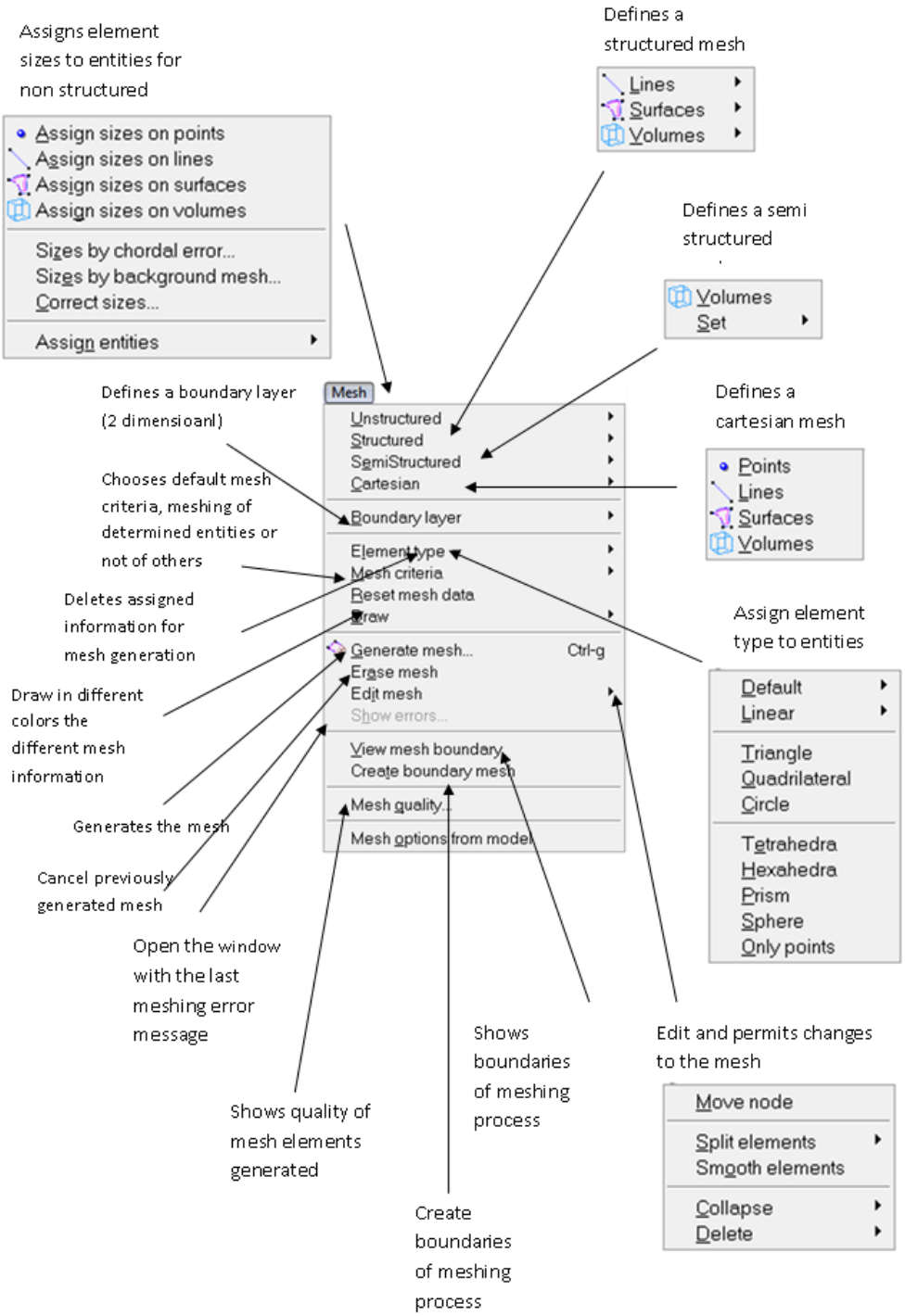
1.2.5 Data

This menu allows access to the definition of all data related to materials, boundary conditions, etc., which will be necessary for the calculations that follow. The form of this data will depend on the type of the analysis to be performed.



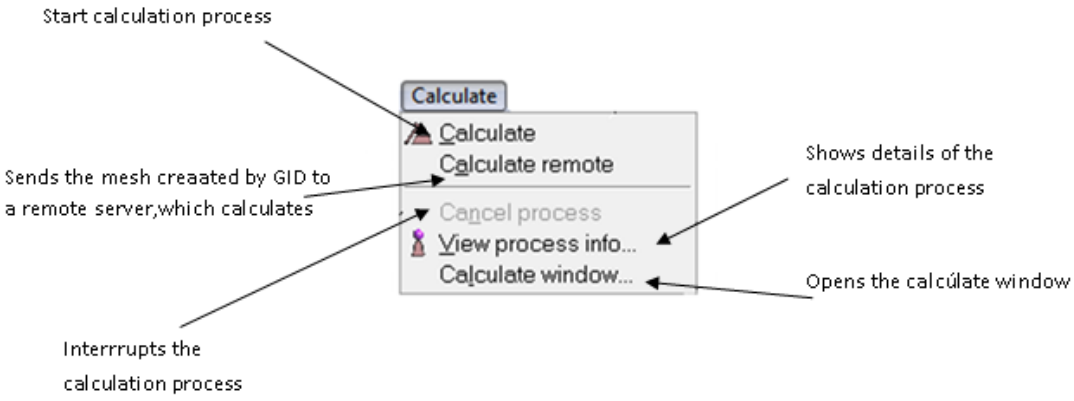
1.2.6 Mesh

Mesh permits the user to generate and edit the mesh, as well as to select mesh creation preferences.



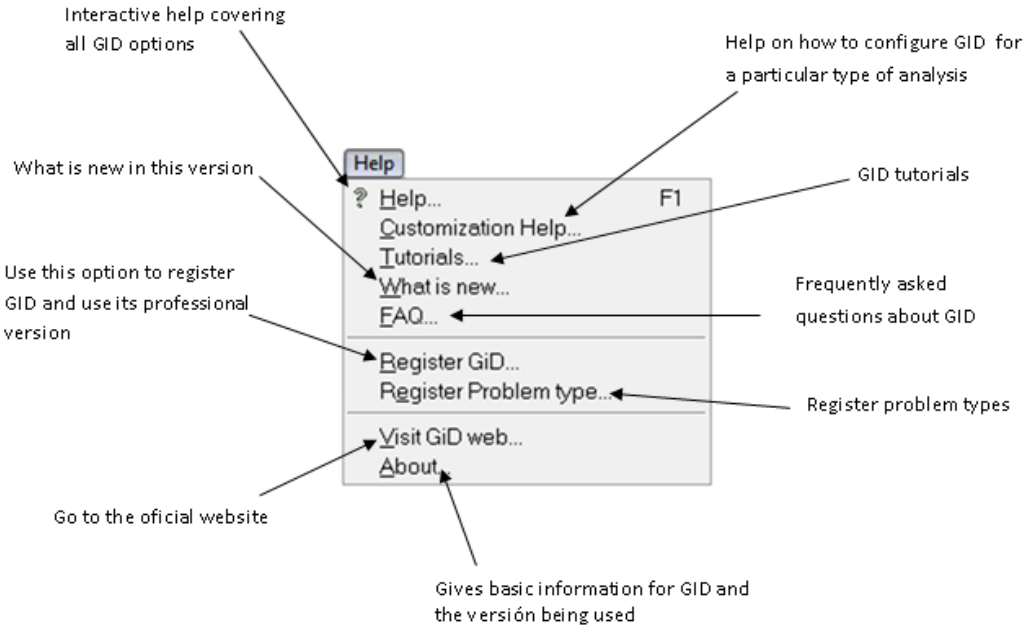
1.2.7 Calculate

This command calculates the problem, according to the type of problem defined. This option requires a previously activated *interface* between **GiD** and the corresponding calculation program.



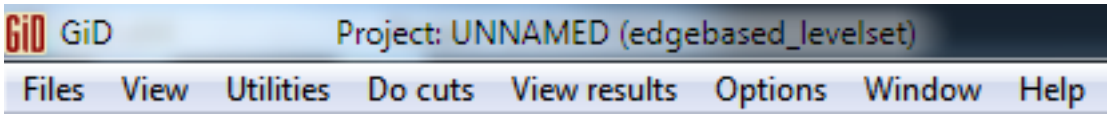
1.2.8 Help

This menu permits the user to obtain different types of help and information about **GiD**.

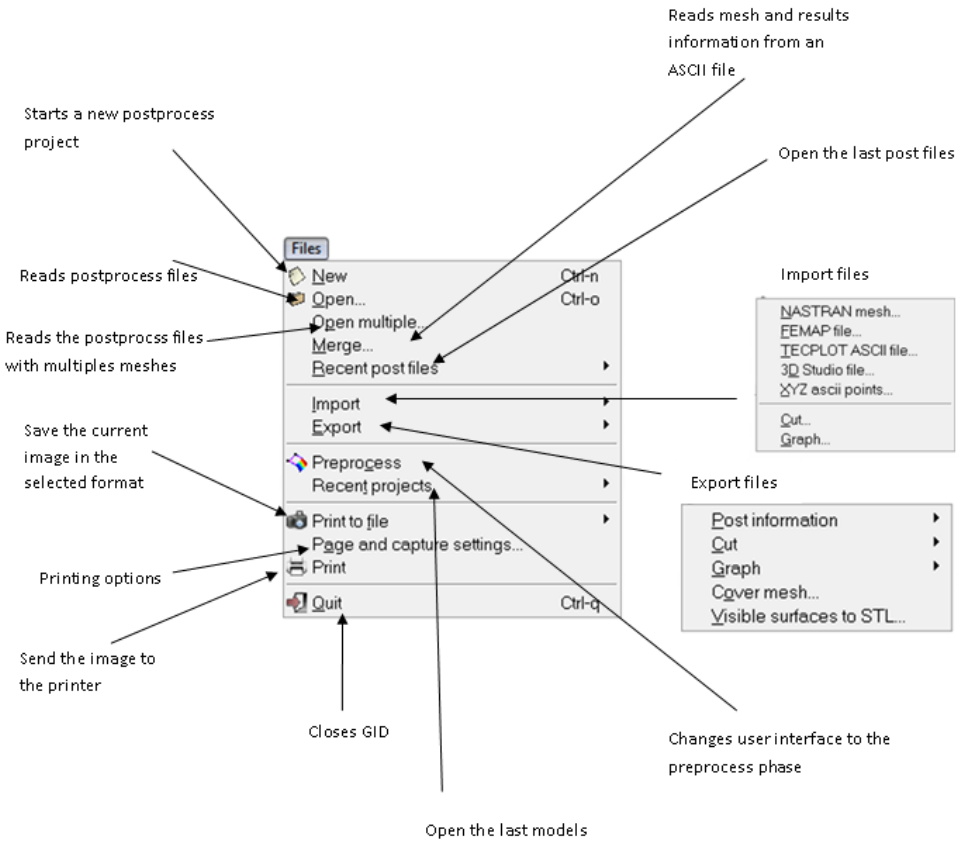


1.2.9 Files

GID PostProcess

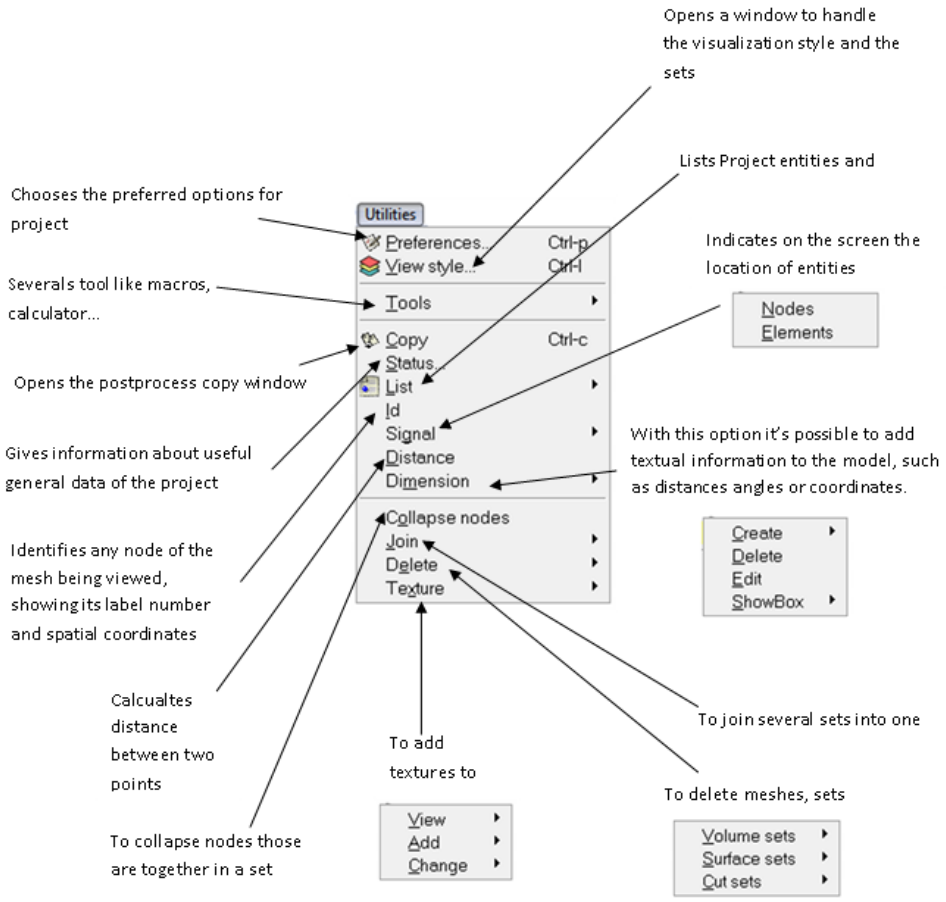


This **Top Menu** of the postprocess phase is the same of that as the preprocess phase and has the same name. The user can read and save files, save screen images, return to preprocess phase options and exit the program.



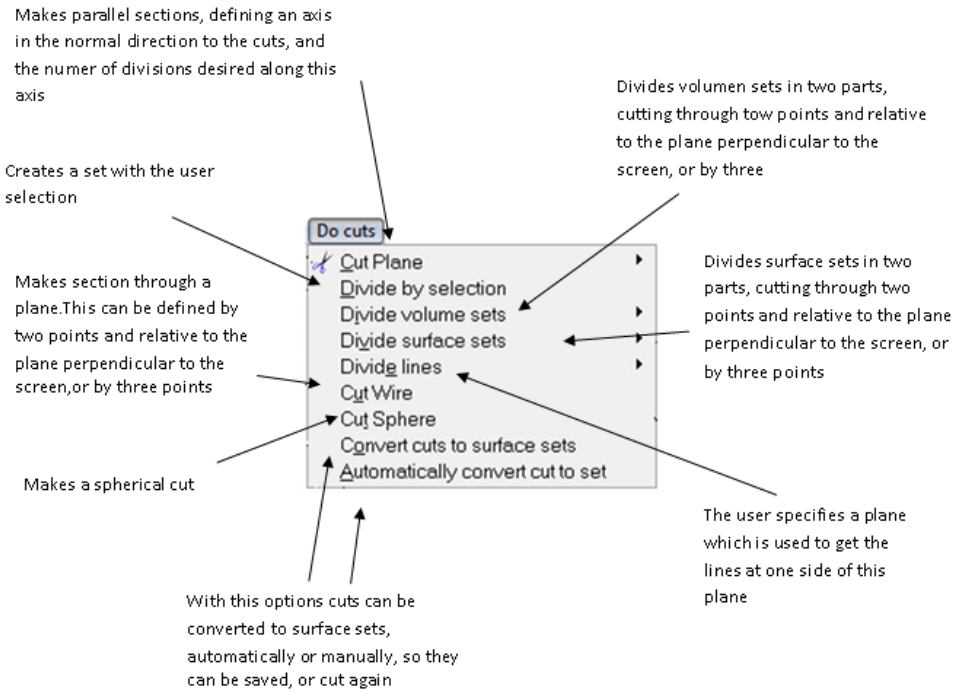
1.2.10 Utilities

In the postprocess phase, the **Utilities** command permits the user to obtain information about entities.



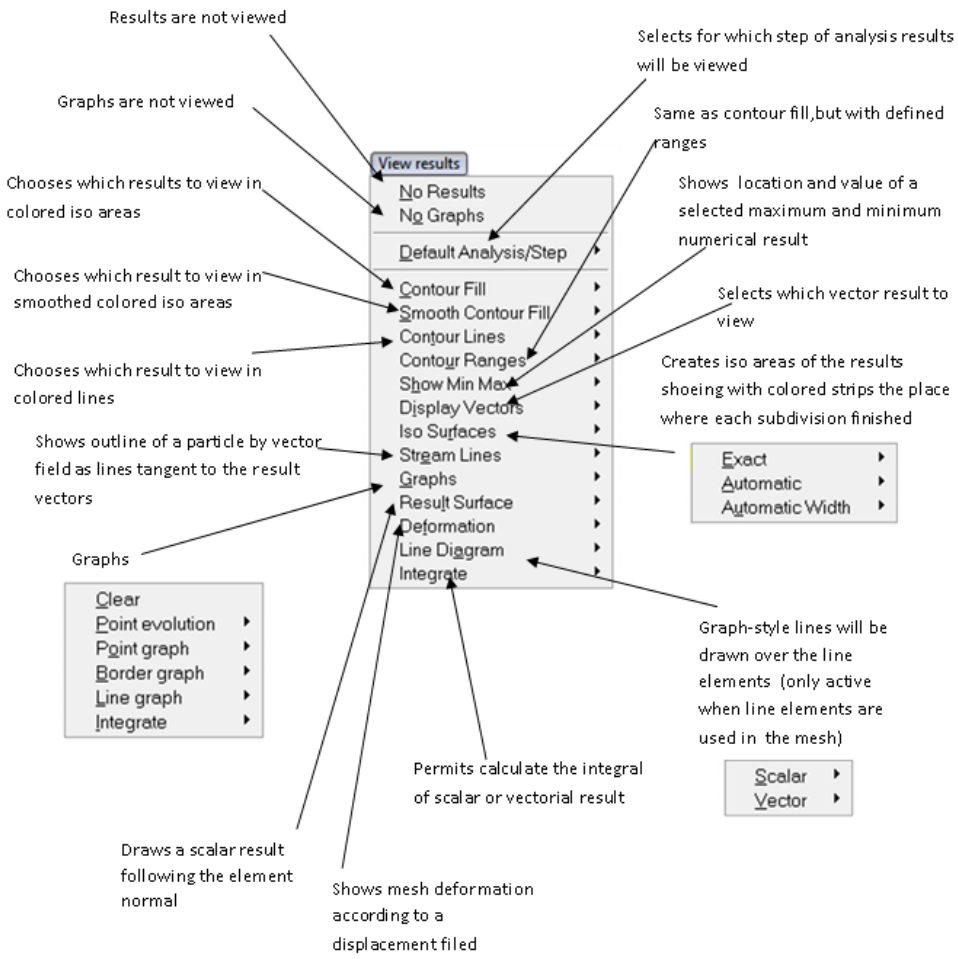
1.2.11 Do cuts

With the option **Do cuts** the user can make cuts through entities.



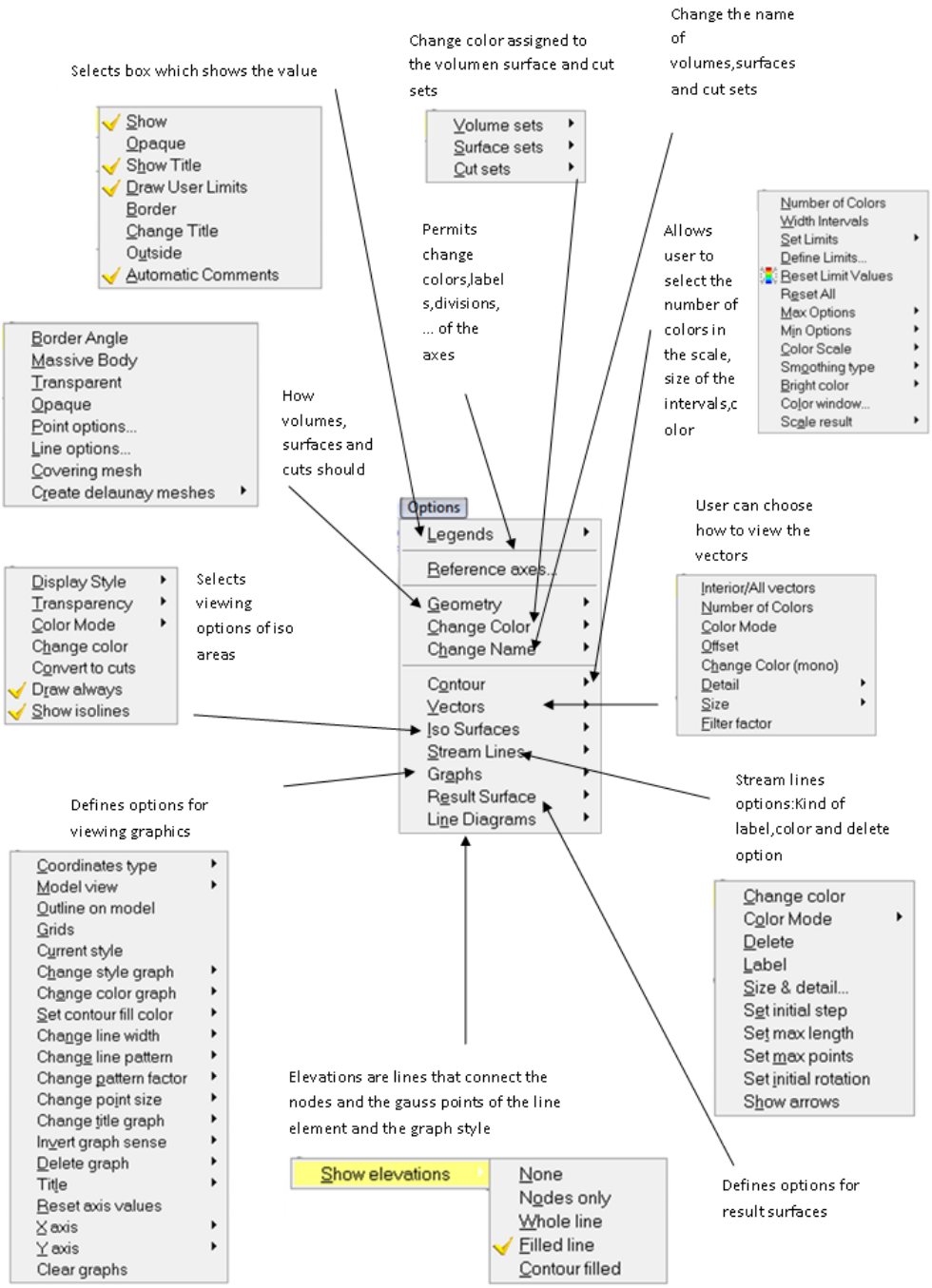
1.2.12 View results

This option permits the user to choose the viewing type in which the results of the postprocess calculation will be presented.

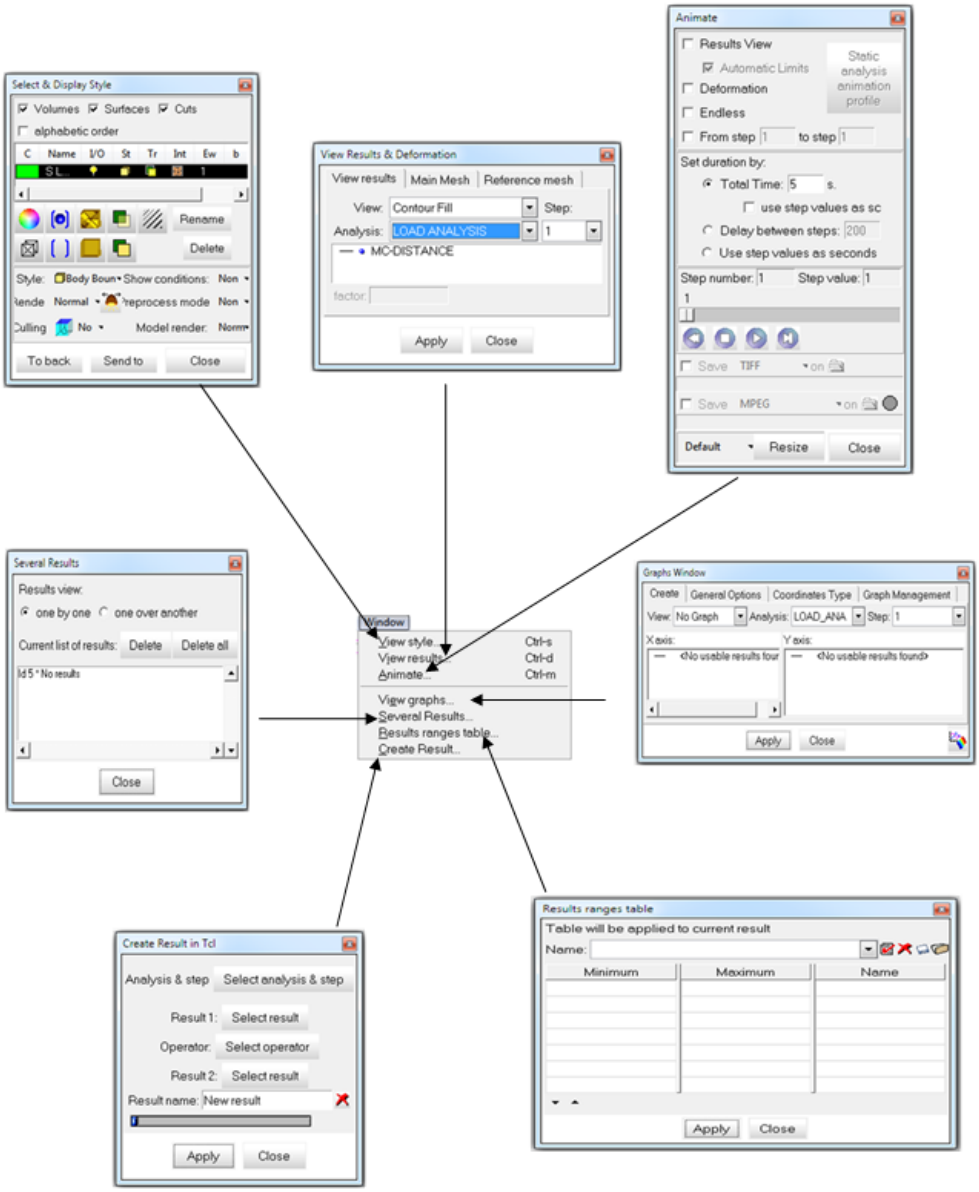


1.2.13 Options

Options permit the user to make choices related to the presentation of results: for example, color changes, number of result subdivisions, etc.



1.2.14 Postprocess windows

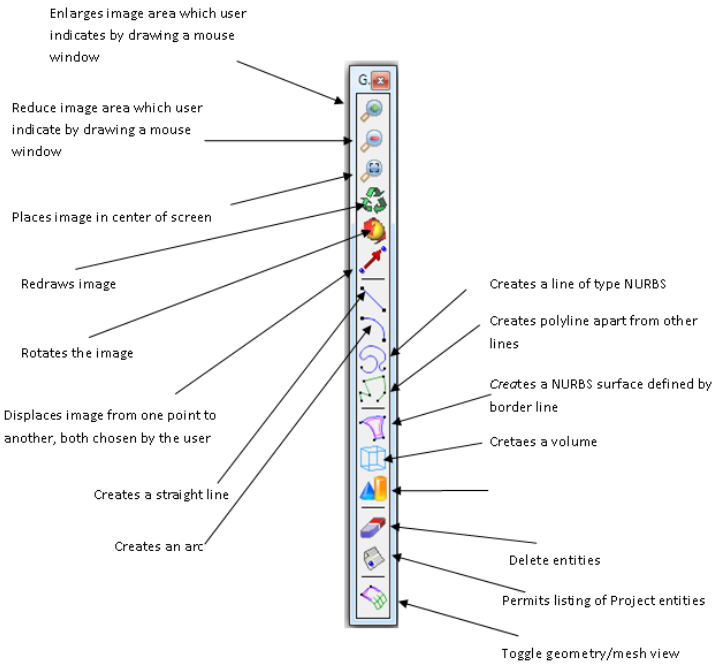


1.3 Toolbars

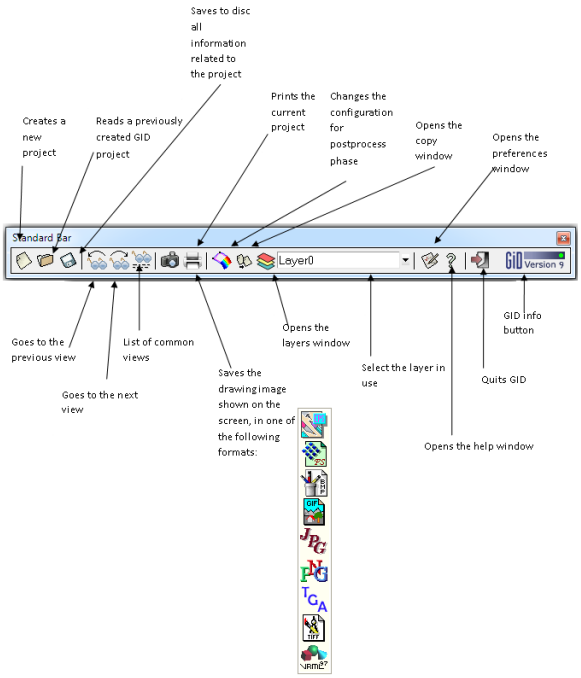
Option **Utilities->Tools->Toolbars...** opens a window where it's possible to configure the toolbars position or switch them on and off.

Geometry and View operations

(preprocess)

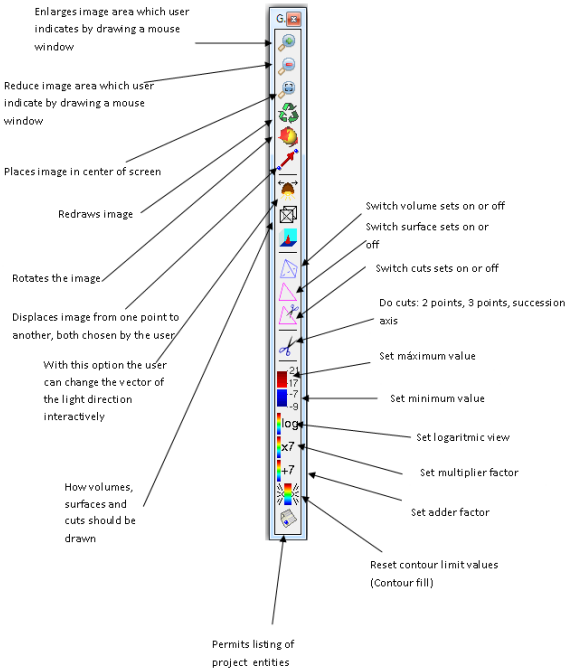


Standard toolbar



Geometry and View operations

(postprocess)

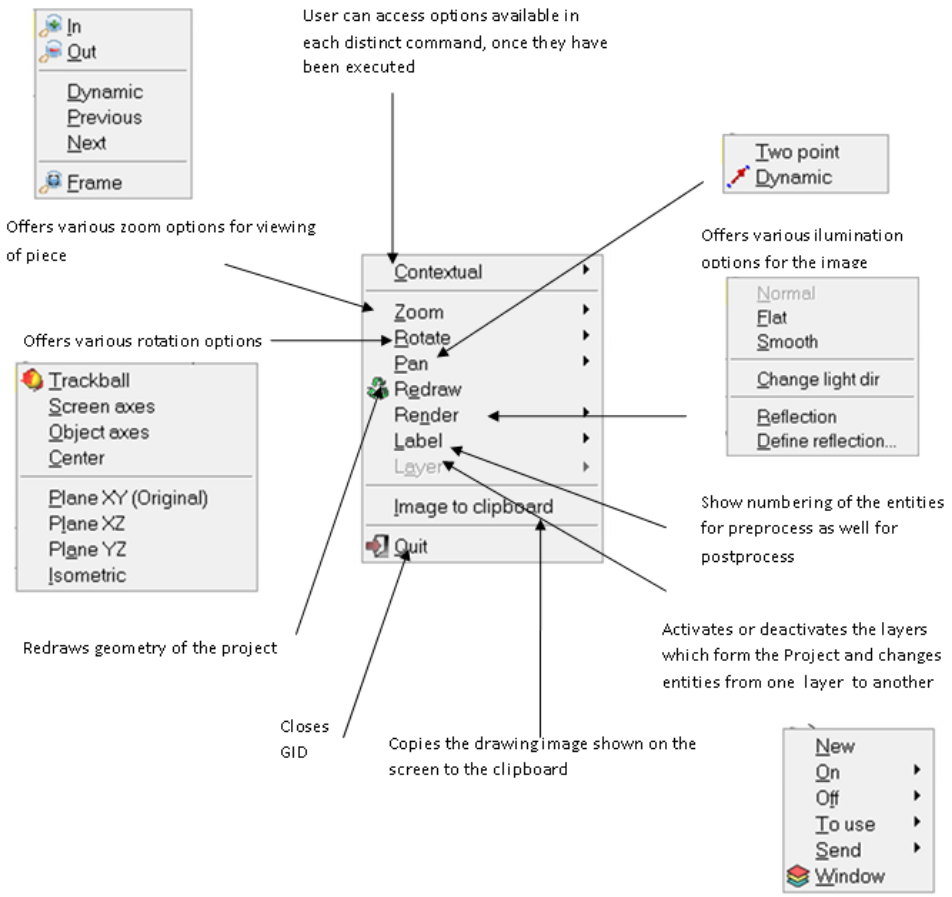


1.4 Mouse menu

The **Mouse Menu** is the auxiliary menu which appears by clicking on the right mouse button while the cursor is over the **GiD** screen.

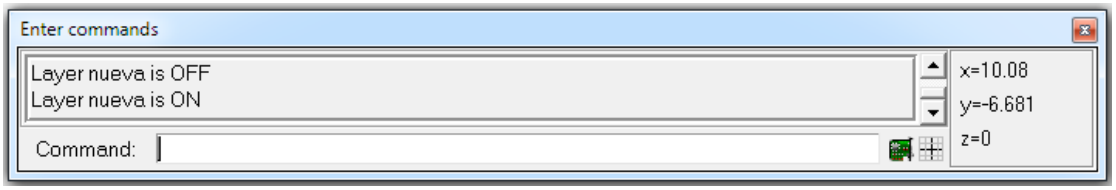
The **Mouse Menu** permits the user to quickly access various image placement and viewing commands, to facilitate easy management and definition of the project.

Furthermore, the **Mouse Menu** contains the **Contextual** menu, which permits the user to access to all options available in previously performed commands. The option **Contextual** is only available after the user has performed a command from the **Top Menu**.



1.5 Command line

The **Command Line** option allows the user to directly enter all executable **GiD** commands, without accessing the commands through drop-down menus.

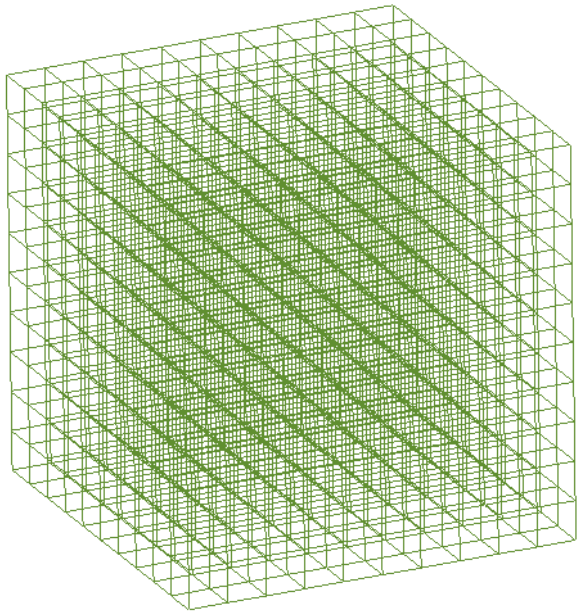


These commands should be written following the order which **GiD** would use to define them, according to the **Right buttons** menus.

A side comment in reference to the **Command Line**: **GiD** does not distinguish between the use of capital and small letters. In addition, in cases where ambiguities do not exist, commands need not be written in entire words, but can be written with the primary characters of each word.

2 INITIATION TO GiD

With this example, the user is introduced to the basic tools for the creation of geometric entities and mesh generation.



2.1 First steps

Before presenting all the possibilities that **GiD** offers, we will present a simple example that will introduce and familiarize the user with the **GiD** program.

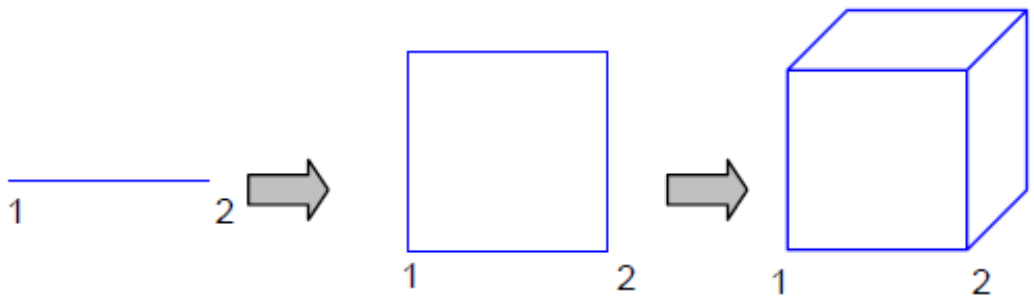
The example will develop a finite element problem in one of its principal phases, the preprocess, and will include the consequent data and parameter description of the problem. This example introduces creation, manipulation and meshing of the geometrical entities used in **GiD**.

First, we will create a line and the mesh corresponding to the line. Next, we will save the project and it will be described in the **GiD** data base form. Starting from this line, we will create a square surface, which will be meshed to obtain a surface mesh. Finally, we will use this surface to create a cubic volume, from which a volume mesh can then be generated.

2.2 Creation and meshing of a line

We will begin the example creating a line by defining its origin and end points, points 1 and 2 in the following figure, whose coordinates are (0,0,0) and (10,0,0) respectively.

It is important to note that in creating and working with geometric entities, **GiD** follows the following hierarchical order: point, line, surface, and volume.



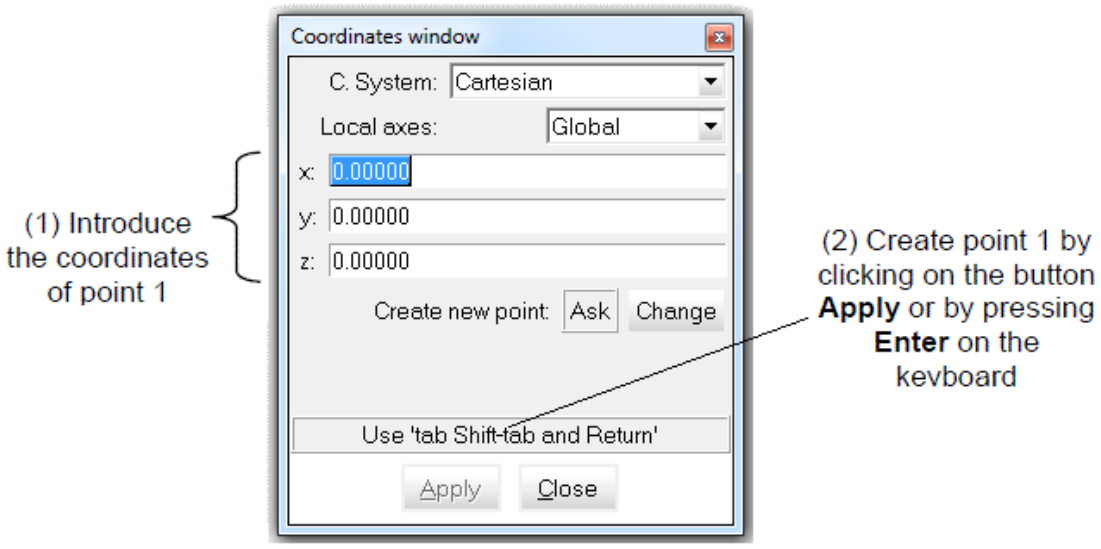
To begin working with the program, open **GiD**, and a new **GiD** project is created automatically.

From this new database, we will first generate points 1 and 2.

Next, we will create points 1 and 2. To do this, we will use an **Auxiliary Window** that will allow us to simply describe the points by entering coordinates. It is accessed by the following sequence: **Utilities->Tools->Coordinates Window**

Then, from the Top Menu, select GeometryCreatePoint

In the coordinate window opened previously, the following indicated steps should be used:

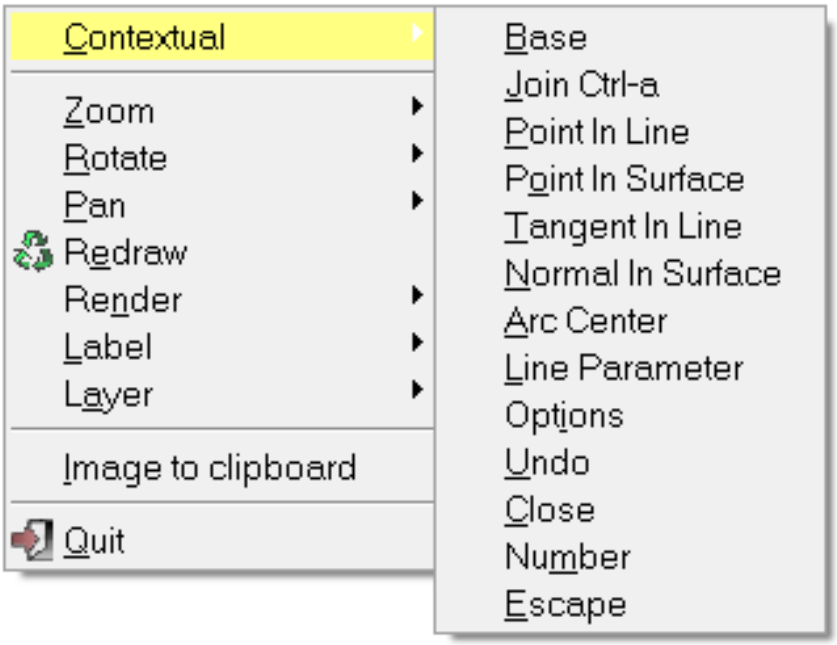


And create point 2 in the same way, introducing its coordinates in the **Coordinates Window**.

The last step in the creation of the points, as well as any other command, is to press **Escape**, either via the **Escape** button on the keyboard or by pressing the central mouse button. Select **Close** to close the **Coordinates Window**.

Now, we will create the line that joins the two points. Choose from the **Top Menu**: **Geometry->Create->Straight line.** Option in the **Toolbar** shown below can also be used.

Next, the origin point of the line must be defined. In the **Mouse Menu**, opened by clicking the right mouse button, select **Contextual->Join C-a.**



NOTE: With option Join, a point already created can be selected on the screen. The command No Join is used to create a new point that has the coordinates of the point that is selected on the screen. We can see that the cursor changes form for the Join and No Join commands.

Cursor during use of **Join** command

Cursor during use of No **Join** command

Now, choose on the screen the first point, and then the second, which define the line. Finally, press **Escape** to indicate that the creation of the line is completed.

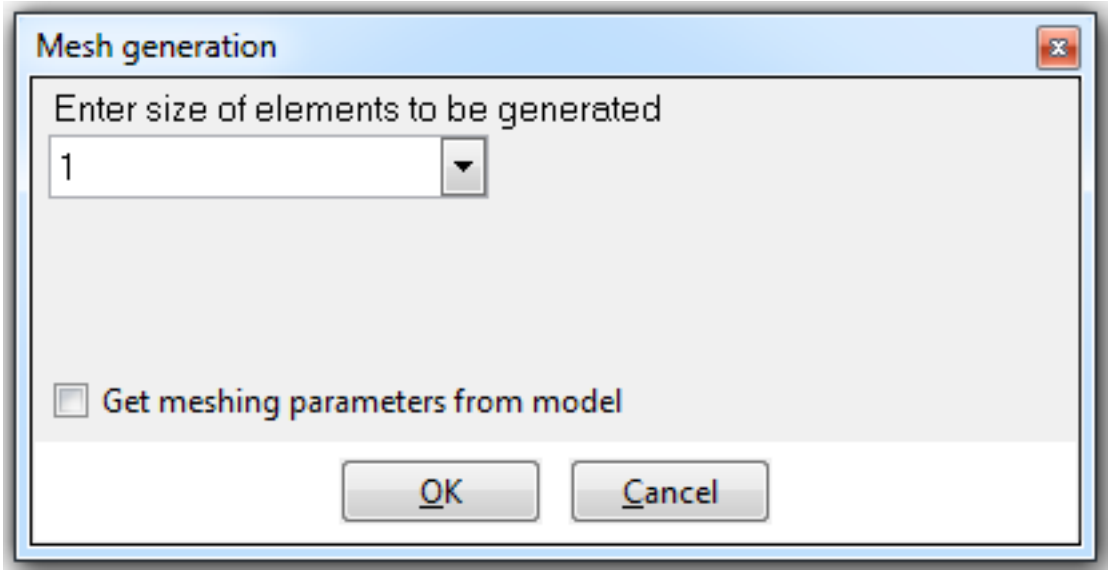
NOTE: It is important to note that the Contextual submenu in the Mouse Menu will always offer the options of the command that is currently being used. In this case, the corresponding submenu for line creation, has the following options:

Once the geometry has been created, we can proceed to the line meshing. In this example, this operation will be presented in the simplest and most automatic way that **GiD** permits. To do this, from the **Top Menu** select: **Mesh->Generate mesh.**

And an **Auxiliary Window** appears, in which the size of the elements should be defined by the user.

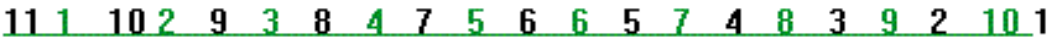
NOTE: The size of an element with two nodes is the length of the element. For, surfaces or volumes, the size is the mean length of the edge of the element.

In this example, the size of the element is defined in concordance with the length of the line, chosen for this case as size 1.



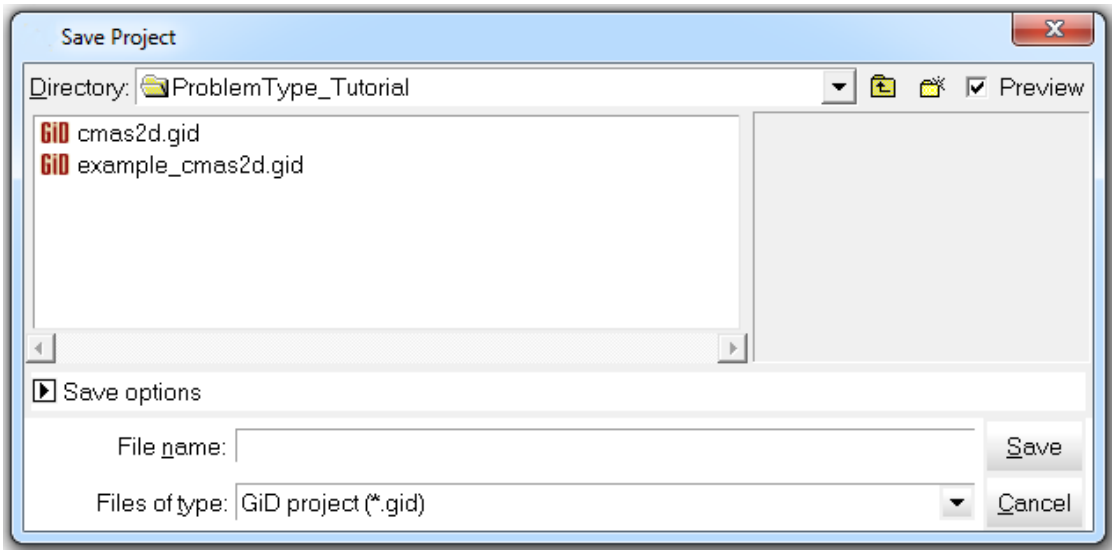
Automatically **GiD** generates a mesh for the line. The finite element mesh is presented on the screen in a green color.

The mesh is formed by ten linear elements of two nodes. To see the numbering of the nodes and mesh elements, select from the **Mouse Menu: LabelAll,** and the numbering for the 10 elements and 11 nodes will be shown, as below.



Once the mesh has been generated, the project should be saved. To save the example select from the **Top Menu: FilesSave.**

The program automatically saves the file if it already has a name. If it is the first time the file has been saved, the user is asked to assign a name. For this, an **Auxiliary Window** will appear which permits the user to browse the computer disk drive and select the location in which to save the file. Once the desired directory has been selected, the name for the actual project can be entered in the space titled **File Name**.



NOTE: Next, the manner in which **GiD** saves the information of a project will be explained. **GiD** creates

a directory with a name chosen by the user, and whose file extension is **.gid**. **GiD** creates a set of files in this directory where all the information generated in the present example is saved. All the files have the same name of the directory to which they belong, but with different extensions. These files should have the name that **GiD** designates and should not be changed manually.

Each time the user selects option **save** the database will be rewritten with the new information or changes made to the project, always maintaining the same name.

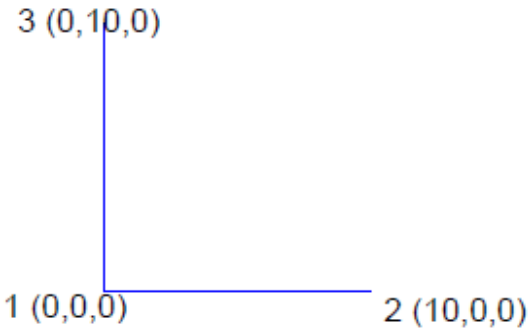
To exit **GiD**, simply choose **FilesQuit.**

To access the example, **ejemplo.gid**, simply open **GiD** and select from the **Top Menu: FilesOpen.** An **Auxiliary Window** will appear which allows the user to access and open the directory **iniciación.gid**.

2.3 Creation and meshing of a surface

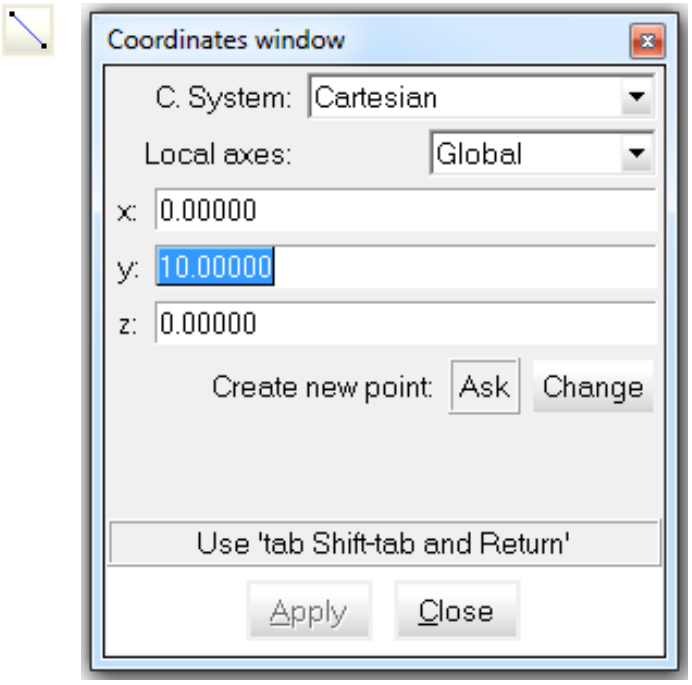
We will now continue with the creation and meshing of a surface.

First, we will create a second line between points 1 and 3.



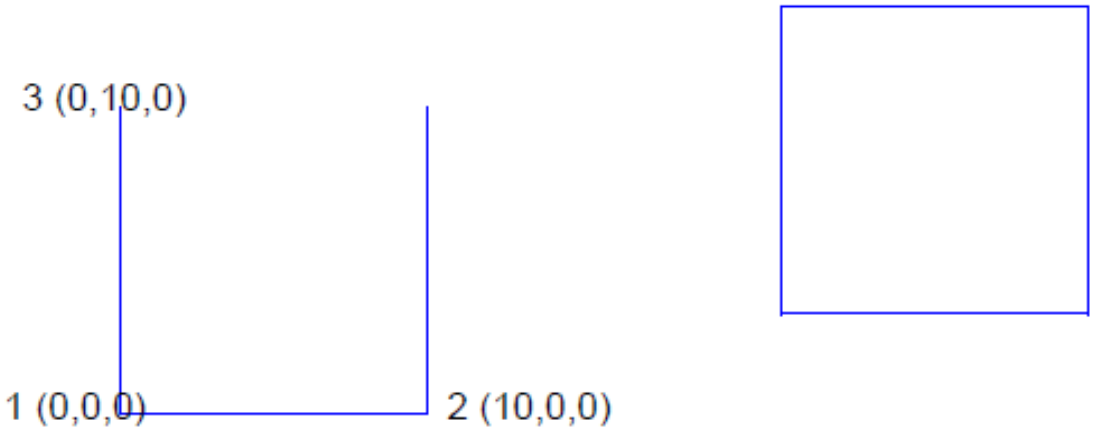
We will now generate the second line. We will now use again the **Coordinates Window** to enter the points. (**Utilities->ToolsCoordinates-> Window**)

Select the line creation tool in the toolbar. Enter point (0,10,0) in the **Coordinates Window** and click **Apply**.



With option **Join** (Contextual mouse menu) click over point 1. A line should be created between (0,10,0) and (0,0,0). Press **Escape**.

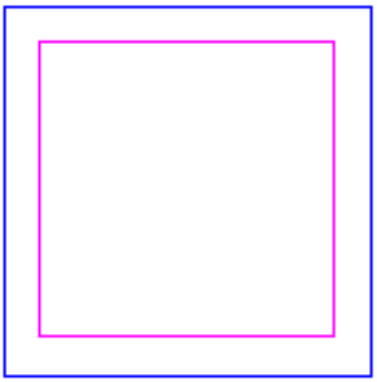
With this, a right angle of the square has been defined. If the user wants to view everything that has been created to this point, the image can be centered on the screen by choosing in the **Mouse Menu: Zoom->Frame**. This option is also available in the toolbar. Finish the square by creating point (10,10,0) and the lines that join this point with points 2 and 3.



Now, we will create the surface that these four lines define. To do this, access the create surface command by choosing: **Geometry->Create->NURBS surface->By contour**. This option is also available in the toolbar:

GiD then asks the user to define the 4 lines that describe the contour of the surface. Select the lines using the cursor on the screen, either by choosing them one by one or selecting them all with a window. Next, press **Escape**.

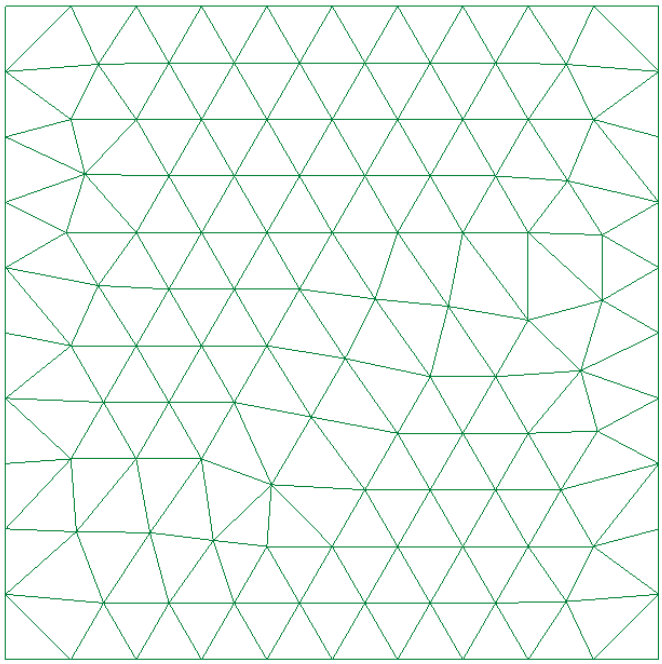
As can be seen below, the new surface is created and appears as a smaller, magenta-colored square drawn inside the original four lines.



Once the surface has been created, the mesh can be created in the same way as was done for the line. From the **Top Menu** select: **Mesh->Generate mesh.**

An **Auxiliary Window** appears which asks for the maximum size of the element, in this example defined

as 1.

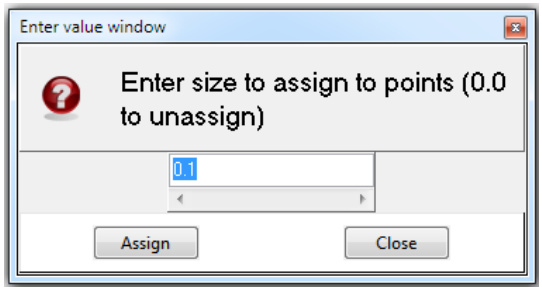


We can see that the lines containing elements of two nodes have not been meshed. Rather the mesh generated over the surface consists of planes of three-nodded, triangular elements.

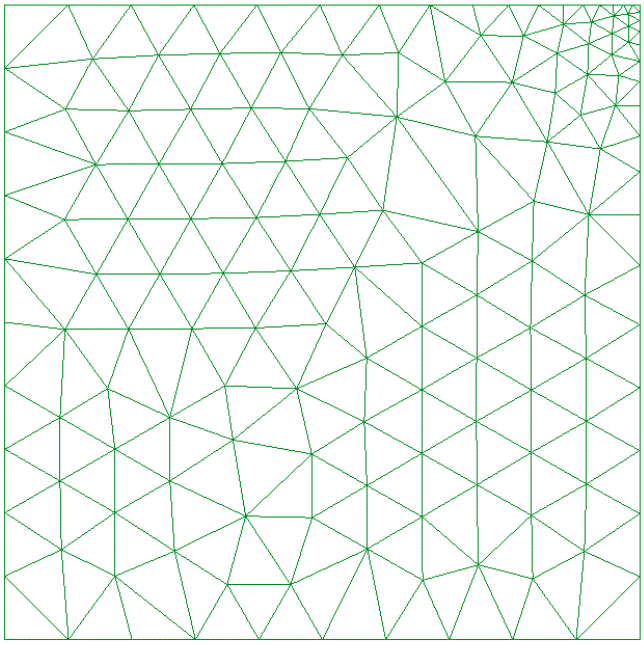
NOTE: GiD meshes by default the entity of highest order with which it is working.

GiD allows the user to concentrate elements in specified geometry zones. Next, a brief example will be presented in which the elements are concentrated in the top right corner of the square.

This operation is realized by assigning a smaller element size to the point in this zone than for the rest of the mesh. Select the following sequence: **Mesh->Unstructured->Assign sizes on points.** The following dialog box appears, in which the user can define the size:



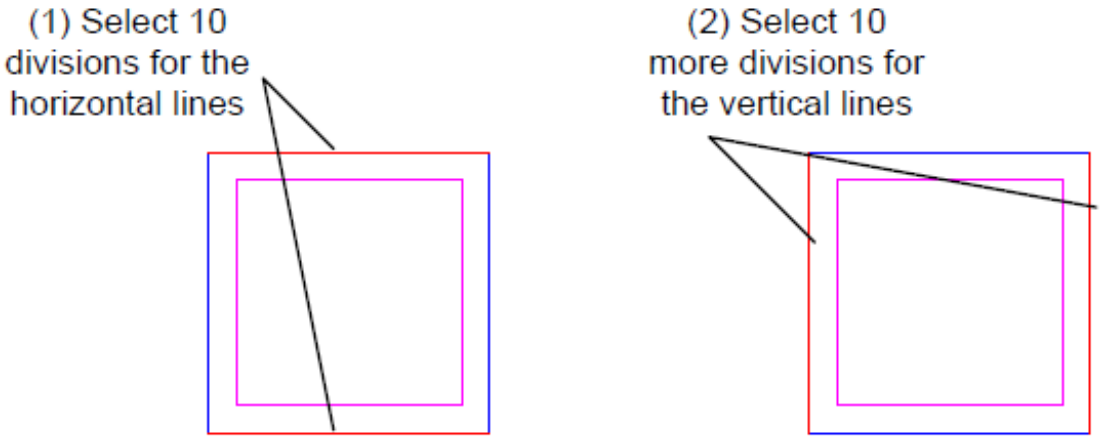
We must now regenerate the mesh, canceling the mesh generated earlier, and we obtain the following:



As can be seen in the figure above, the elements are concentrated around the chosen point. Various possibilities exist for controlling the evolution of the element size, which will be presented later in the manual.

To generate a surface mesh in which the elements are presented uniformly, the user can select the option for a structured mesh. This guarantees that the same number of elements appears around a node and that the element size is as uniform as possible. To generate this type of mesh, choose: **Mesh->Structured->Surfaces.**

Using this command, the user should first select the **4-sided NURBS surface** that will be defined by the mesh. Then, the number of subdivisions for the surface limit lines should be entered. Pairs of lines define the partitions in the following way:

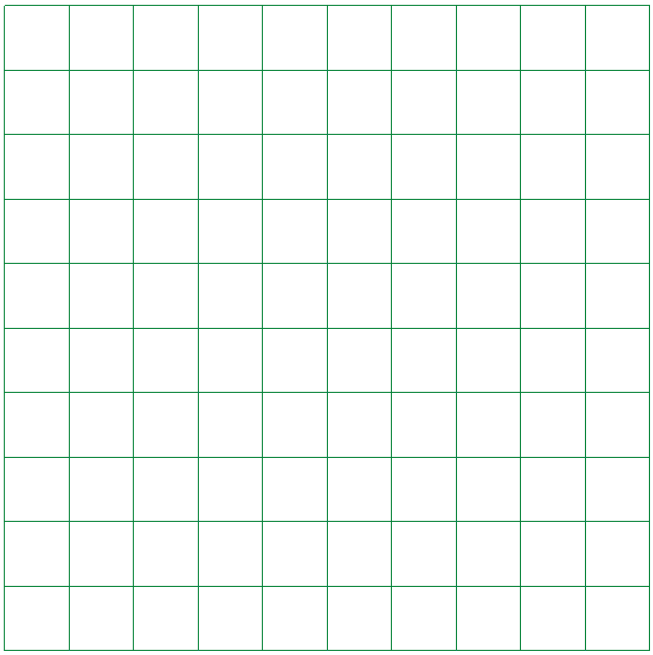


NOTE: GiD only generates structured meshes for surfaces of the type **4-sided surface** or **NURBS surface**.

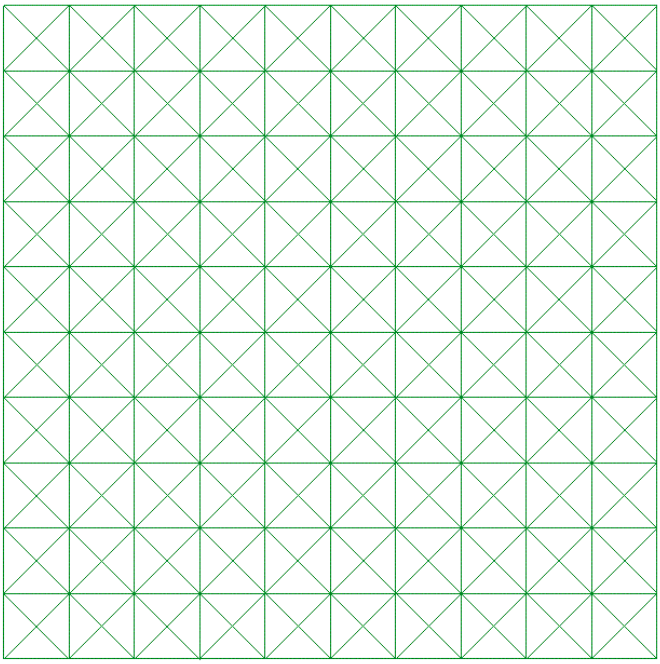
When this has been done, the mesh is generated in the same way as the unstructured mesh, by

choosing **Mesh->Generate mesh.**

Assign a general element size of 1, though in this case it is not necessary.



We can see here that the default element type used by **GiD** to create a structured mesh is a square element of four nodes rather than a three-nodded, triangular element. To obtain triangular elements, the user can specifically define this type of element, by choosing **Mesh->Element type->Triangle,** and selecting the surface to mesh as a triangular element. Regenerate the mesh, and the following figure is obtained:

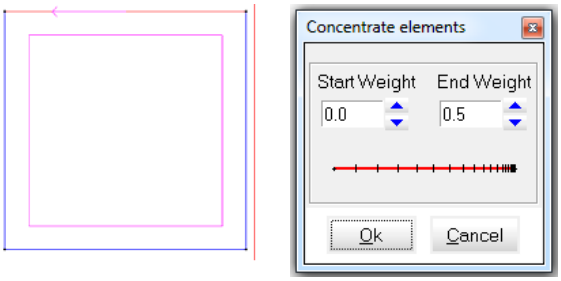


GiD also allows the user to concentrate elements in structured meshes. This can be done by selecting **Mesh->Structured->Lines->Concentrate elements**

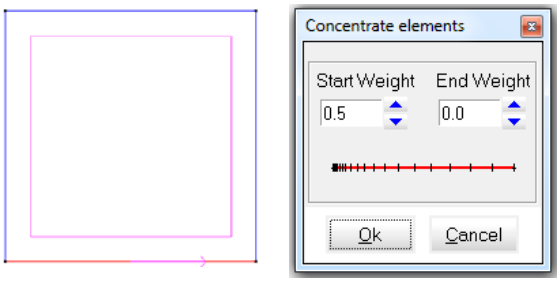
First, we must select the lines that need to be assigned an element concentration weight. The value of

this weight can be either positive or negative, depending on whether the user wants to concentrate elements at the beginning or end of the lines. Next, a vector appears which defines the start and end of the line and which helps the user assign the weight correctly.

Select the top line and assign a weight of 0.5 to the end of the line:

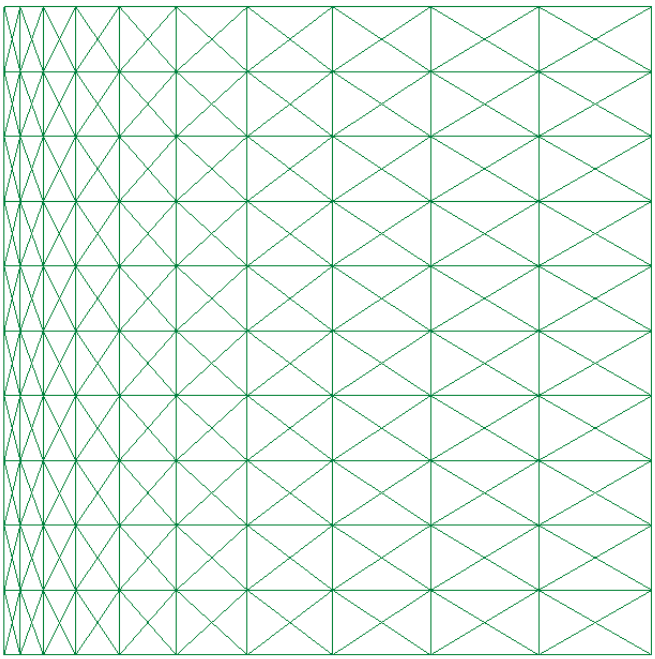


Select the bottom line and assign a weight of 0.5 to the beginning of the line:



From these operations, we obtain the following mesh:

We can see that in the figure above, the elements are concentrated in the left zone of the square.



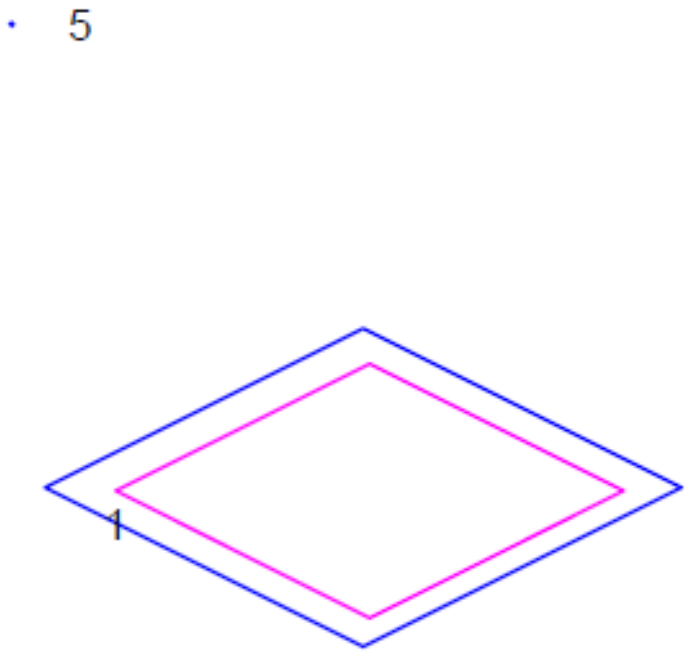
2.4 Creation and meshing of a volume

We will now present a study of entities of volume. To illustrate this, a cube and a volume mesh will be generated.

Without leaving the project, save the work done up to now by choosing **Files->Save,** and return to the geometry last created by choosing **Geometry->View geometry.**

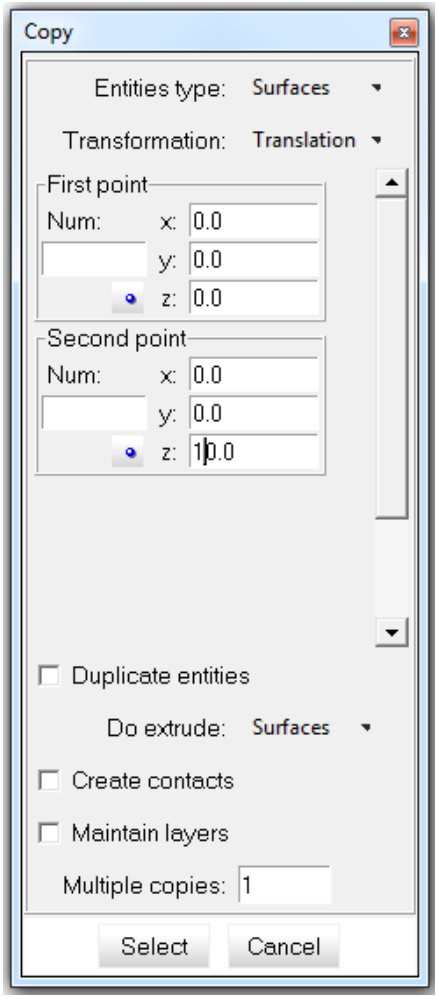
In order to create a volume from the existing geometry, firstly we must create a point that will define the height of the cube. This will be point 5 with coordinates (0,0,10), superimposed on point 1. To view the new point, we must rotate the figure by selecting from the **Mouse Menu, RotateTrackball.** This option is also available in the toolbar:

Rotate the figure until the following position is achieved:



Next, we will create the upper face of the cube by copying from point 1 to point 5 the surface created previously. To do this, select the copy command, **Utilities->Copy.**

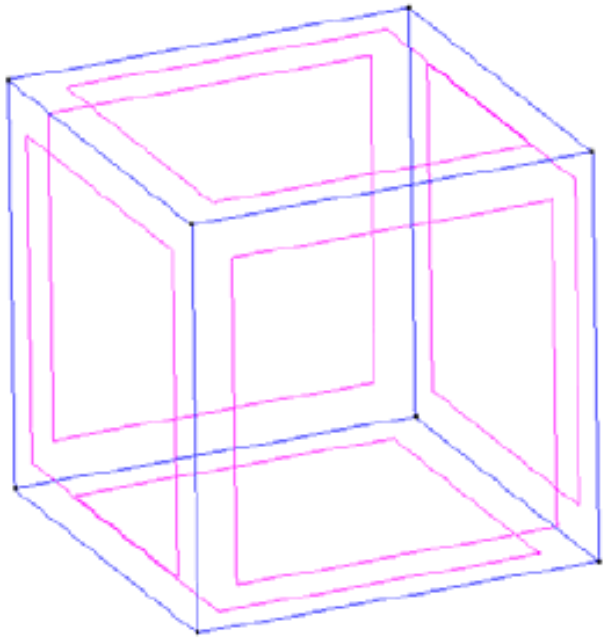
In the **Copy** window, we define the translation vector with the first and second points, in this case (0,0,0) and (0,0,10). Option **Do extrude surfaces** must be selected; this option allows us to create the lateral surfaces of the cube.



NOTE: If we look at the **Copy Window,** we can see an option called **Duplicate entities**. By activating this option, when the entities are copied (in this case from point 1 to point 5) **GiD** would create a new point (point 6) with the same coordinates as point 5.

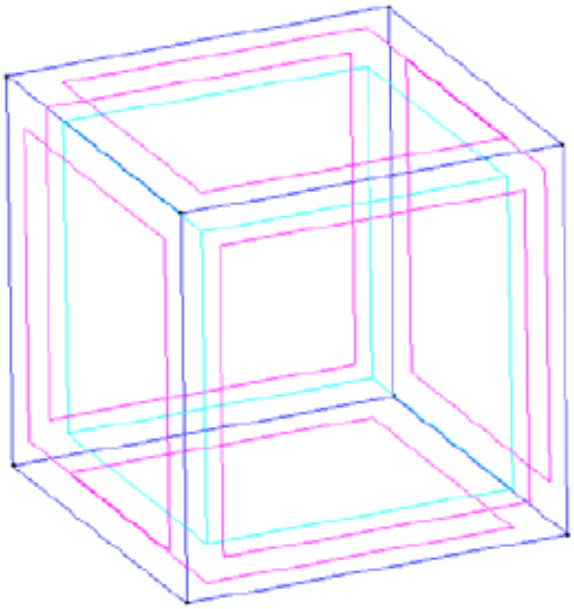
If the user does not choose option **Duplicate entities** , point 6 will be merged with point 5 when the entities are copied. By labeling the entities we could verify that only one point has been created.

Finishing the copy command for the surface, we obtain the following surfaces:

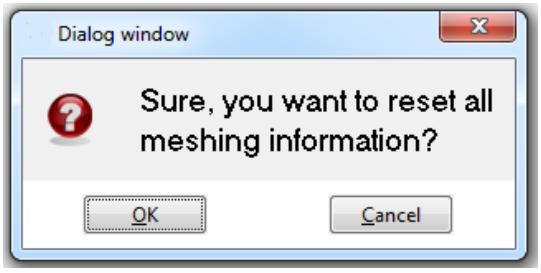


Now, we can generate the volume delimited by these surfaces. To create the volume, simply select the command **Geometry->Create->Volume->By contour** . This option is also available in the toolbar:

Select all the surfaces. **GiD** automatically generates the volume of the cube. The volume viewed on the screen is represented by a cube with an interior color of sky blue.



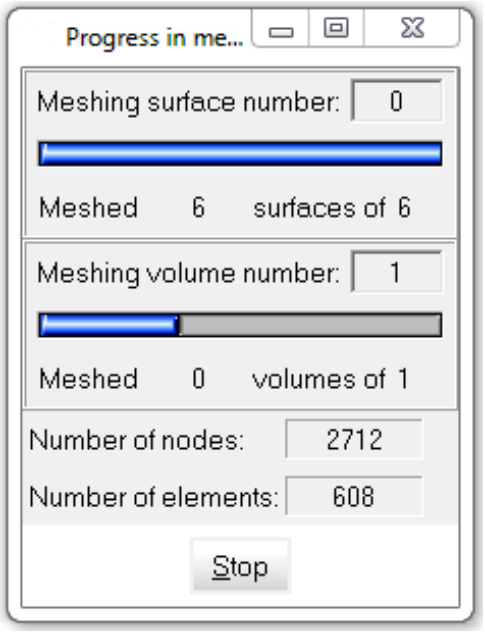
Before proceeding with the mesh generation of the volume, we should eliminate the information of the structured mesh created previously for the surface. Do this by selecting **Mesh->Reset mesh data,** and the following dialog box will appear on the screen:

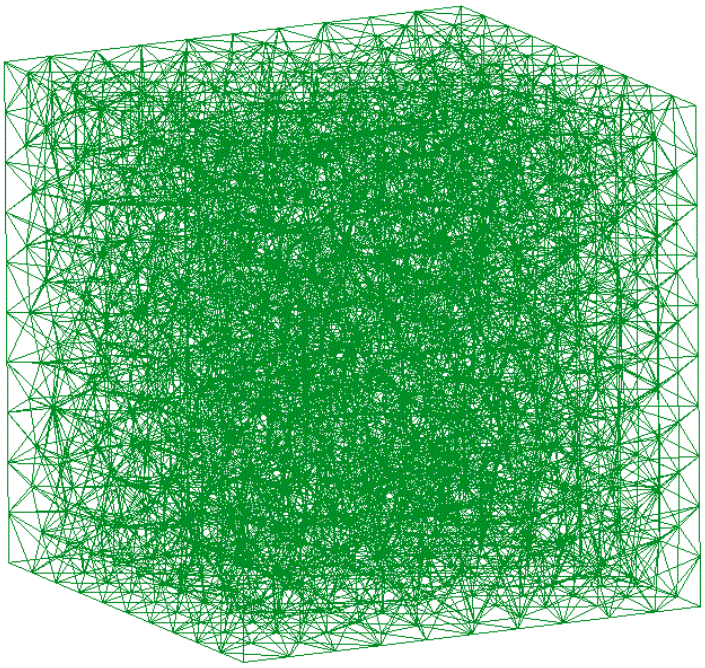


In which the user is asked to confirm the erasure of the mesh information.

NOTE: Another valid option would be to assign a size of 0 to all entities. This would eliminate all the previous size information as well as the information for the mesh, and the default options would become active.

Next, generate the mesh of the volume by choosing **Mesh->Generate mesh.** Another **Auxiliary Window** appears into which the size of the volumetric element must be entered. In this example, the value is 1.

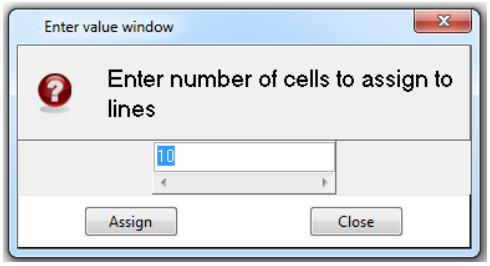


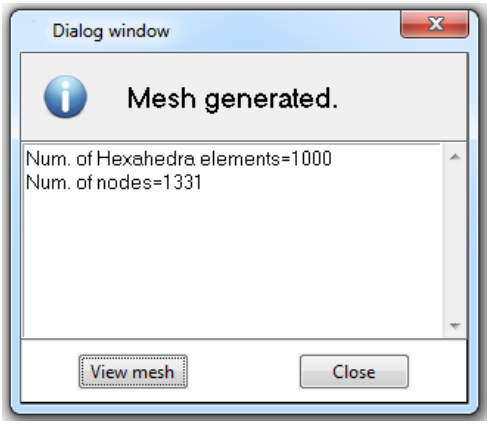


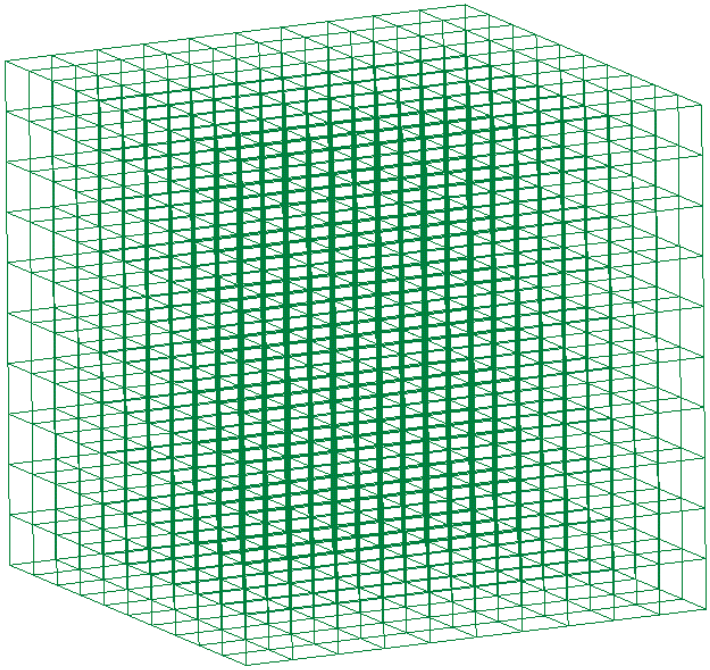
The mesh generated above is composed of tetrahedral elements of four nodes, but **GiD** also permits the use of hexahedral, eight-nodded structured elements.

We will generate a structured mesh of the volume of the cube. This is done by selecting **Mesh->Structured->Volumes.**

Now select the volume to mesh and enter the number of partitions in its edges which will be created. Then, create again the mesh.







NOTE: GiD only allows the generation of structured meshes of 6-sided volumes.

With this example, the user has been introduced to the basic tools for the creation of geometric entities and mesh generation.

3 IMPLEMENTING A MECHANICAL PART

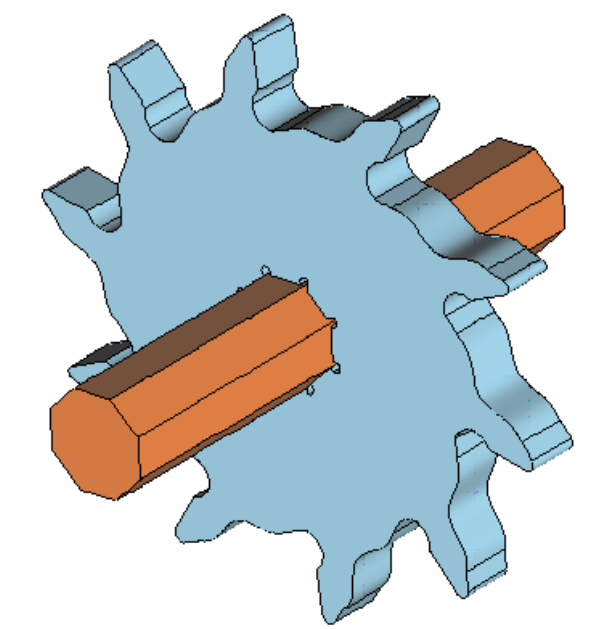
2D TOOLS, BASIC 3D TOOLS AND MESHING

IMPLEMENTING A MECHANICAL PART

The objective of this case study is implementing a mechanical part in order to study it through meshing analysis. The development of the model consists of the following steps:

- Creating a profile of the part
  - Generating a volume defined by the profile
  - Generating the mesh for the part

At the end of this case study, you should be able to handle the 2D tools available in GiD as well as the options for generating meshes and visualizing the prototype.



3.1 Working by layers

3.1.1 Defining the layers

A geometric representation is composed of four types of entities, namely points, lines, surfaces, and volumes.

A layer is a grouping of entities. Defining layers in computer-aided design allows us to work collectively with all the entities in one layer.

The creation of a profile of the mechanical part in our case study will be carried out with the help of auxiliary lines. Two layers will be defined in order to prevent these lines from appearing in the final drawing. The lines that define the profile will be assigned to one of the layers, called the "profile" layer, while the auxiliary lines will be assigned to the other layer, called the "aux" layer. When the design of the part has been completed, the entities in the "aux" layer will be erased.

3.1.2 Creating two new layers

1 . Open the layer management window. This is found in **Utilities->Layers**.

2 . Create two new layers called "aux" and "profile." Enter the name of each layer in the **Layers** window (Figure 1) and click **New**.

3 . Choose "aux" as the activated layer. To do this, click on "aux" to highlight it and then click on the **Layer To Use** button. (Next to this button the name of the activated layer will appear, "aux" in the present case.) From now on, all the entities created will belong to this layer.

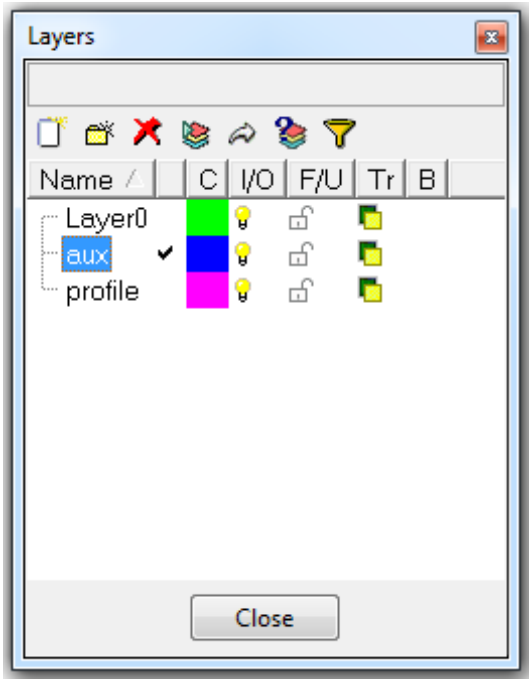


Figure 1. The Layers window

3.2 Creating a profile

In our case, the profile consists of various teeth. Begin by drawing one of these teeth, which will be copied later to obtain the entire profile.

3.2.1 Creating a size-55 auxiliary line

1 . Choose the **Line** option, by going to **Geometry->Create->Straight line** or by going to the GiD Toolbox[1].

2 . Enter the coordinates of the beginning and end points of the auxiliary line[2]. For our example, the coordinates are (0, 0) and (55, 0), respectively. Besides creating a straight line, this operation implies creating the end points of the line.

3 . Press **ESC** [3]to indicate that the process of creating the line is finished.

4 . If the entire line does not appear on the screen, use the **Zoom Frame** option,which is located in the GiD Toolbox and in **Zoom** in the mouse menu.

Figure 2. Creating a straight line

NOTE: The **Undo** option, located in **Utilities->Undo**, enables you to undo the most recent operations. When this option is selected, a window appears in which all the operations to be undone can be selected.

---

[1] The GiD Toolbox is a window containing the icons for the most frequently executed operations. For information on a particular tool, click on the corresponding icon with the right mouse button.

[2] The coordinates of a point may be entered on the command line either with a space or a comma between them. If the Z coordinate is not entered, it is considered 0 by default. After entering the numbers, press **Return**. Another option for entering a point is using the **Coordinates Window**, found in **Utilities->Tools->Coordinates Window**.

[3] Pressing the **ESC** key is equivalent to pressing the center mouse button.

3.2.2 Dividing the auxiliary line near "point" (coordinates (40, 0))

1 . Choose **Geometry->Edit->Divide->Lines->NearPoint.** This option will divide the line at the point ("element") on the line closest to the coordinates entered.
2 . Enter the coordinates of the point that will divide the line. In this example, the coordinates are (40, 0). On dividing the line, a new point (entity) has been created.
3 . Select the line that is to be divided by clicking on it.
4 . Press **ESC** to indicate that the process of dividing the line is finished.

Figure 3. Division of the straight line near "point" (coordinates) (40, 0)

3.2.3 Creating a 3.8-radius circle around point (40, 0)

1 . Choose the option **Geometry->Create->Object->Circle**.
2 . The center of the circle (40, 0) is a point that already exists. To select it, go to **Contextual->Join Ctrl-a** in the mouse menu (right-click). The pointer will become a square, which means that you may click an existing point.
3 . The **Enter Normal** window appears. Set the normal as Positive Z and press **OK**.
4 . Enter the radius of the circle. The radius is 3.8 [4]. Two circumferences are visualized; the inner circumference represents the surface of the circle.

Press **ESC** to indicate that the process of creating the circle is finished.

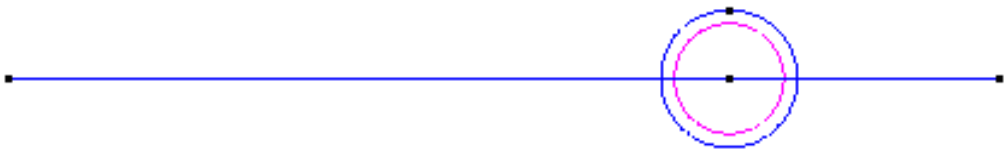


Figure 4. Creating a circle around a point (40, 0)

---

[4] In GiD the decimals are entered with a point, not a comma.

3.2.4 Rotating the circle -3 degrees around a point

1 . Use the **Move** window, which is located in **Utilities->Move**.

2 . Within the **Move** menu and from among the **Transformation** possibilities, select **Rotation**. The type of entity to receive the rotation is a surface, so from the **Entities Type** menu, choose **Surfaces**.

3 . Enter -3 in the **Angle** box and check the **Two dimensions** box. (Provided we define positive rotation in the mathematical sense, which is counter-clockwise, -3 degrees equates to a clockwise rotation of 3 degrees).

4 . Enter the point (0, 0, 0) under **First Point**. This is the point that defines the center of rotation.

5 . Click **Select** to select the surface that is to rotate, which in this case is that of the circle.

6 . Press **ESC** (or **Finish** in the **Move window**) to indicate that the selection of surfaces to rotate has been made, thus executing the rotation.

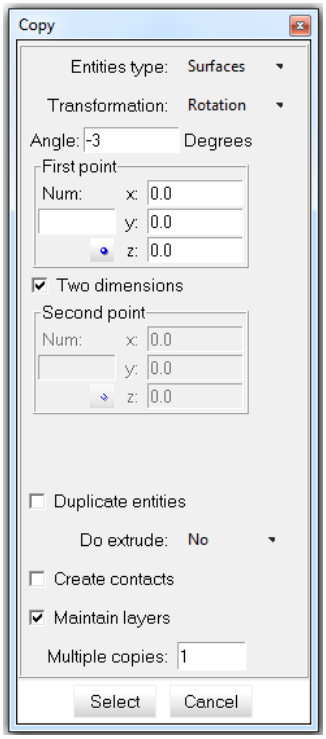


Figure 5. The Move window

3.2.5 Rotating the circle 36 degrees around a point and copying it.

1 . Use the **Copy** window, located in **Utilities->Copy**.

2 . Repeat the rotation process explained in section 2.4, but this time with an angle of 36 degrees (see Figure 6).

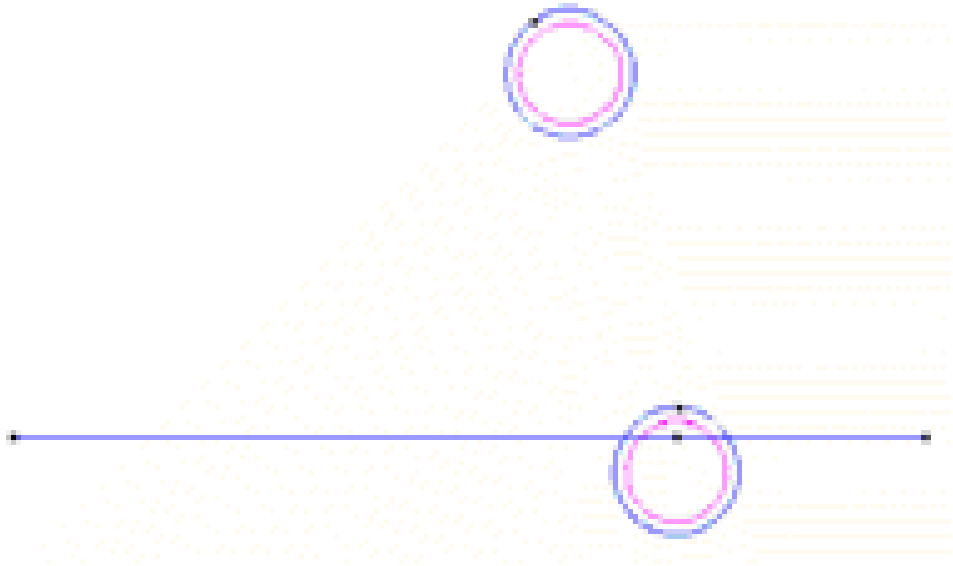


Figure 6. Result of the rotations

NOTE: The **Move** and **Copy** operations differ only in that **Copy** creates new entities while **Move** displaces entities.

3.2.6 Rotating and copying the auxiliary lines

1 . Use the **Copy** window, located in **Utilities->Copy** (see Figure 9).
2 . Repeat the rotating and copying process from section 2.5 for the two auxiliary lines. Select the option **Lines** from the **Entities type** menuand enter an angle of 36 degrees.
3 . Select the lines to copy and rotate. Do this by clicking **Select** in the **Copy** window.
4 . Press **ESC** to indicate that the process of selecting lines is finished, thus executing the task (see Figure 7).

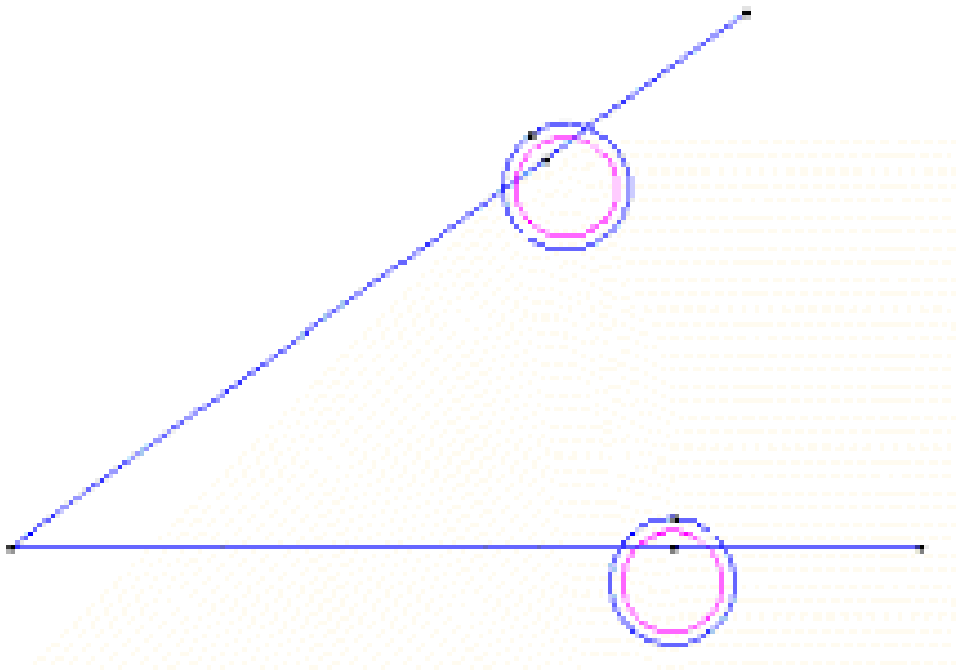


Figure 7. Result of copying and rotating the line.

5 . Rotate the line segment that goes from the origin to point (40, 0) by 33 degrees and copy it (see Figure 8).

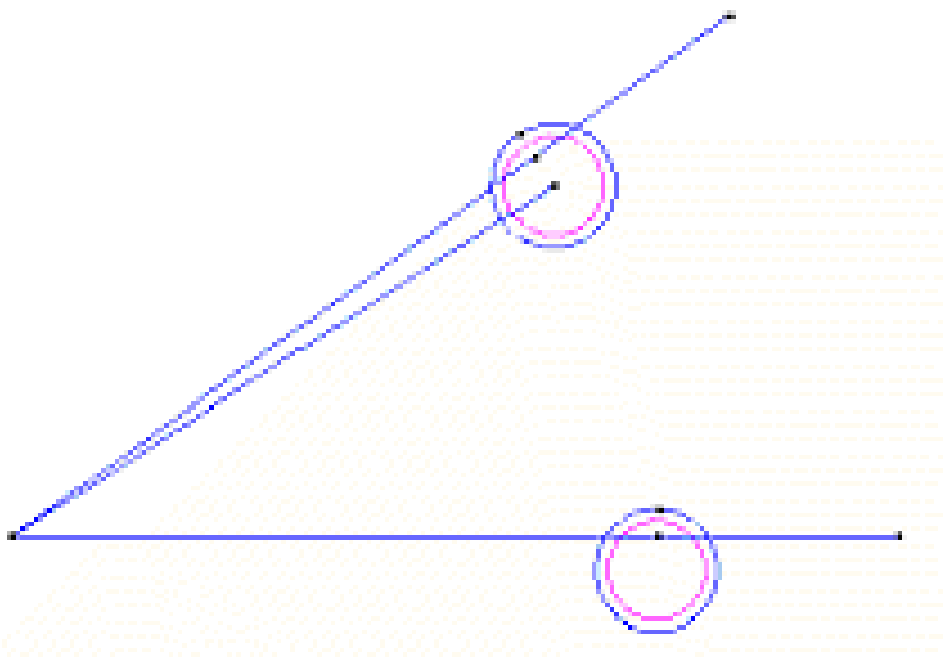


Figure 8. Result of the rotations and copies

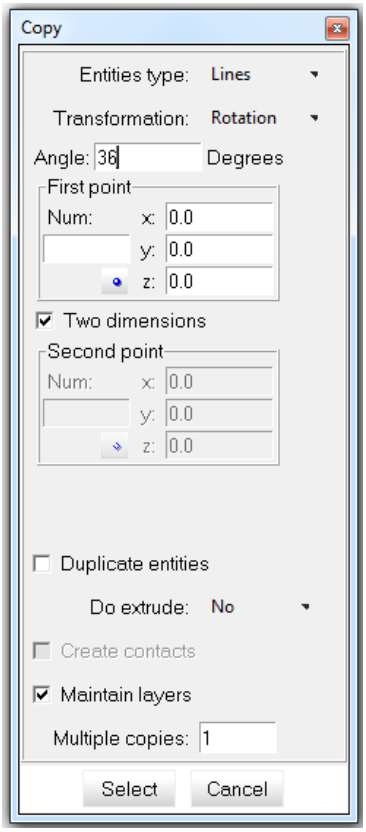


Figure 9. The copy window

NOTE: In the **Copy** and **Move** windows, the option **Pick** may be used to select existing points with the mouse.

3.2.7 Intersecting lines

1 . Choose the option **Geometry->Edit-> Intersection->Line-line.**
2 . Select the upper circle resulting from the 36-degree rotation executed in section 2.5.
3 . Select the line resulting from the 33-degree rotation executed in section 2.6 (see Figure 10).
4 . Press **ESC** to conclude the intersection of lines.
5 . Create a line between point (55, 0) and the point generated by the intersection. To select the points, use the option **Join Ctrl-a** in the **Contextual** menu.
6 . Choose the option **Geometry->Edit-> Intersection->Line-line** in order to make another intersection between the lower circle and the line segment between point (40, 0) and point (55, 0) (see Figure 11).
7 . Then continue selecting to make an intersection between the upper circle and the farthest segment of the line that was rotated 36 degrees (see Figure 12).

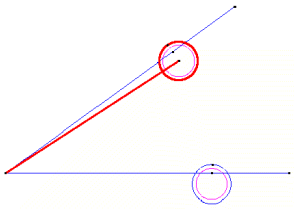


Figure 10. The two lines selected

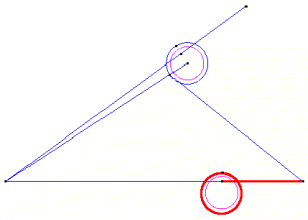


Figure 11. Intersecting lines

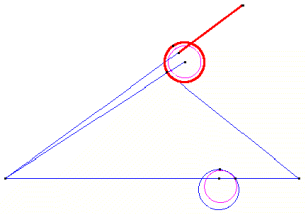


Figure 12. Intersecting lines

3.2.8 Creating an arc tangential to two lines

1 . Choose **Geometry->Create->Arc->Fillet curves**.

2 . Enter a radius of 1.35 inside the command line (see footnote 2).

3 . Now select the two line segments shown in Figure 13. Then press **ESC** to indicate that the process of creating the arcs is finished.

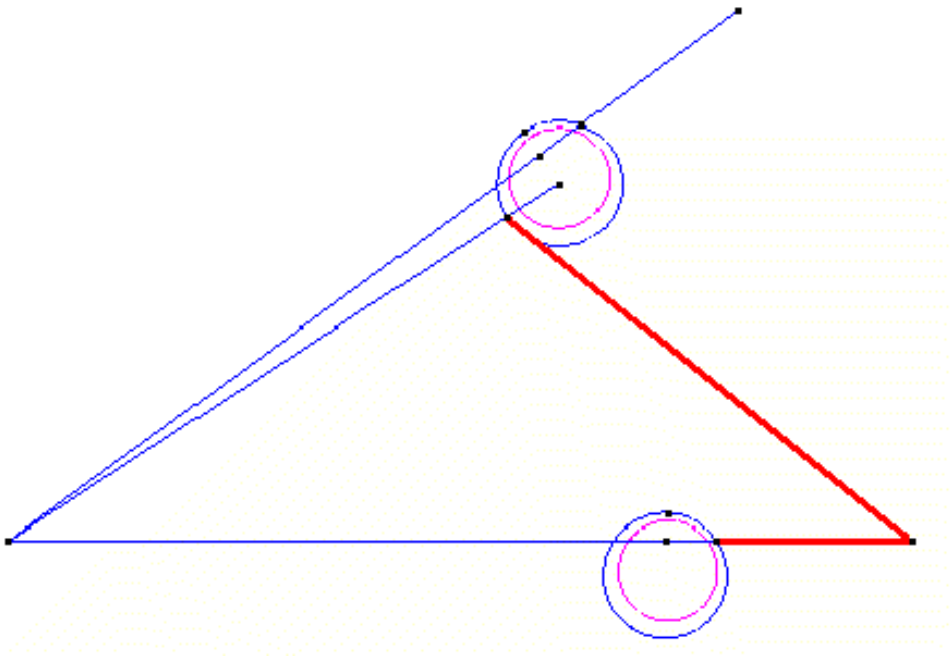


Figure 13. The line segments to be selected

3.2.9 Translating the definitive lines to the "profile" layer

1 . Select the "profile" layer in the **Layers** window. The auxiliary lines will be eliminated and the "profile" layer will contain only the definitive lines.

2 . In the **Sent To** menu of the **Layers** window, choose **Lines** in order to select the lines to be translated. Select only the lines that form the profile (Figure 14). To conclude the selection process, press the **ESC** key or click **Finish** in the **Layers** window.

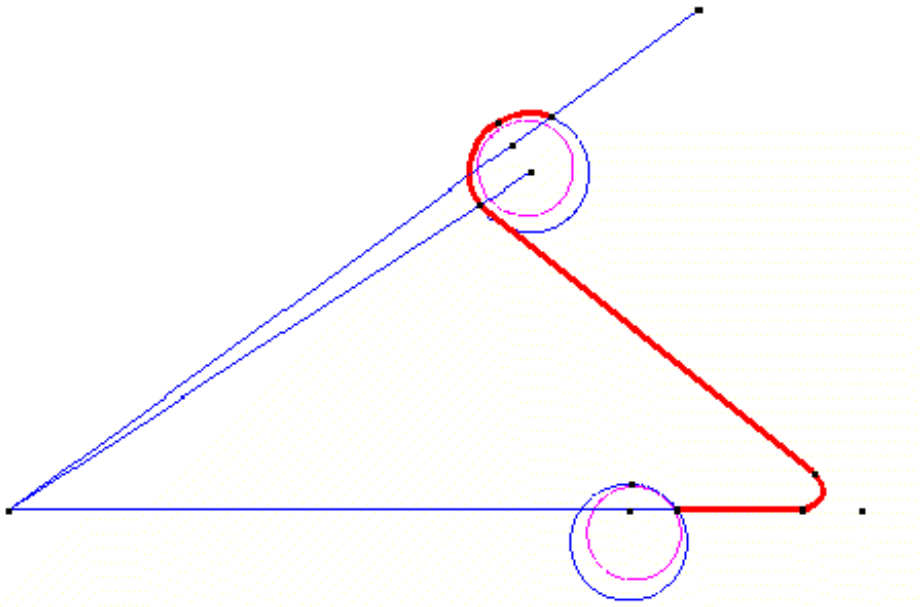


Figure 14. Lines to be selected

3.2.10 Deleting the "aux" layer

1 . Click **Off** the profile layer.
2 . Choose **Geometry->Delete->All Types** (or use the GiD Toolbox).
3 . Select all the lines and surfaces that appear on the screen. (The click-and-drag technique may be used to make the selection).
4 . Press **ESC** to conclude the selection of elements to delete.
5 . Select the "aux" layer in the **Layers** window and click **Delete**.
6 . Select the "profile" layer.

NOTE: When a layer is clicked **Off**, GiD reminds you of this. From this moment on, whatever is drawn does not appear on the screen since it is in the hidden layer.

NOTE: To cancel the deletion of elements after they have been selected, open the mouse menu, go to **Contextual** and choose **Clear Selection**.

NOTE: Elements forming part of higher level entities may not be deleted. For example, a point that defines a line may not be deleted.

NOTE: A layer containing information may not be deleted. First the contents must be deleted.

3.2.11 Rotating and obtaining the final profile

1 . Make sure that the activated layer is the "profile" layer. (Use the option **Layer To use**).
2 . In the **Copy** window, select the line rotation (**Rotation**, **Lines**).
3 . Enter an angle of 36 degrees. Make sure that the center is point (0, 0, 0) and that you are working in two dimensions.
4 . In the **Multiple Copies** box enter 9. This way, 9 copies will be made, thus obtaining the 10 teeth that form the profile of the model (9 copies and the original).
5 . Click **Select** and select the profile. Press the **ESC** key or click **Finish** in the **Copy** window in order to conclude the operation. The result is shown in Figure 15.

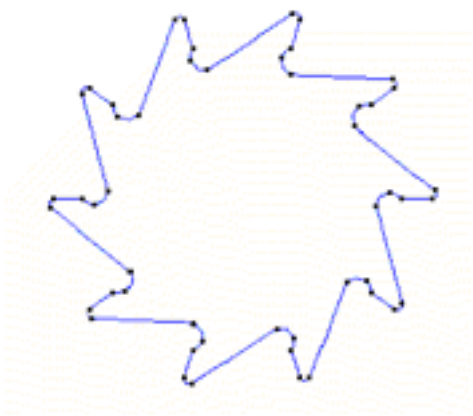


Figure 15. The part resulting from this process

3.2.12 Creating a surface

1 . Create a NURBS surface. To do this, select the option **Geometry->Create->NURBS Surface->By Contour.** This option can also be found in the GiD Toolbox.
2 . Select the lines that define the profile of the part and press **ESC** to create the surface.
3 . Press **ESC** again to exit the function. The result is shown in Figure 16.

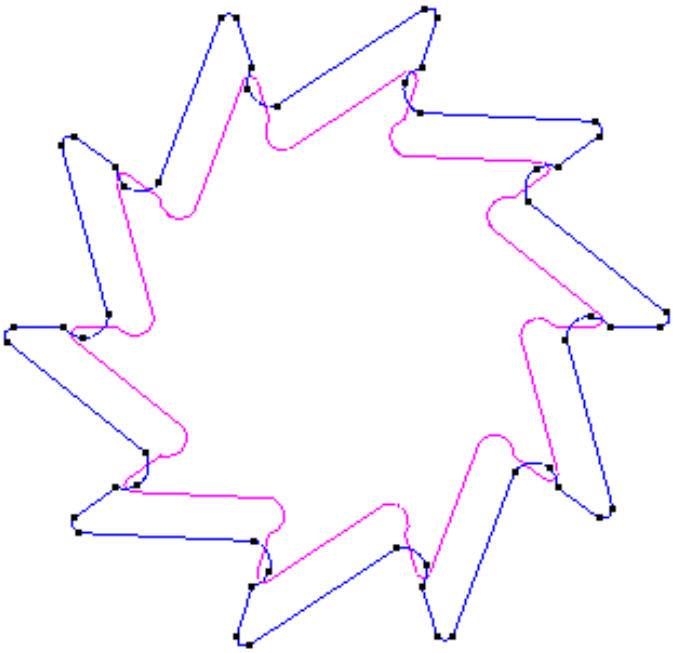


Figure 16. Creating a surface starting from the contour

NOTE: To create a surface there must be a set of lines that define a closed contour.

3.3 Creating a hole in the part

In the previous sections we drew the profile of the part and we created the surface. In this section we will make a hole, an octagon with a radius of 10 units, in the surface of the part. First we will draw the octagon.

1 . Select from the menu **Geometry->Create->Object->Polygon** to create a regular polygon.
2 . Enter 8 as the number of sides of the polygon.
3 . Enter (0,0,0) as the center of the polygon.
4 . Enter or select (0,0,1) (*Positive Z*) as the normal of the polygon.
5 . Enter 10 as the radius of the polygon and press **ENTER**. Press **ESC** to finish the action.

We get the result as shown in figure 20. As we only need the boundary we should remove the associated surface. Select the option **Geometry->Delete->Surfaces** and then select the surface of the octagon. Press **ESC** to finish.

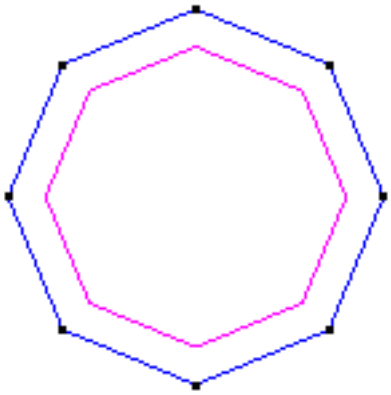


Figure 17. Regular 8-sided polygon

3.3.1 Creating a hole in the surface of the mechanical part

1 . Choose the option **Geometry->Edit->Hole NURBS Surface**.

2 . Select the surface in which to make the hole (Figure 18).

3 . Select the lines that define the hole (Figure 19) and press **ESC**.

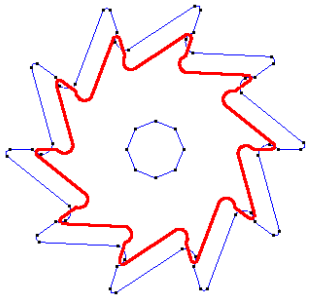


Figure 18. The selected surface in which to create the hole

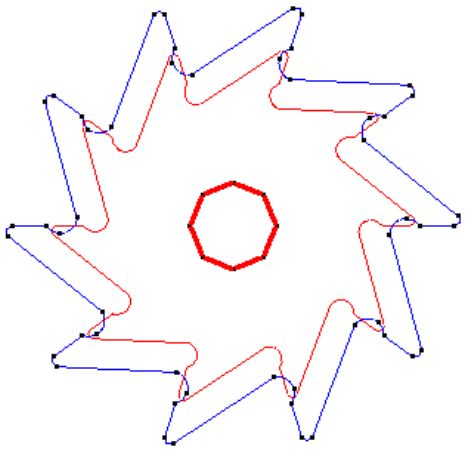


Figure 19. The selected lines that define the hole

4 . Again, press **ESC** to exit this function.

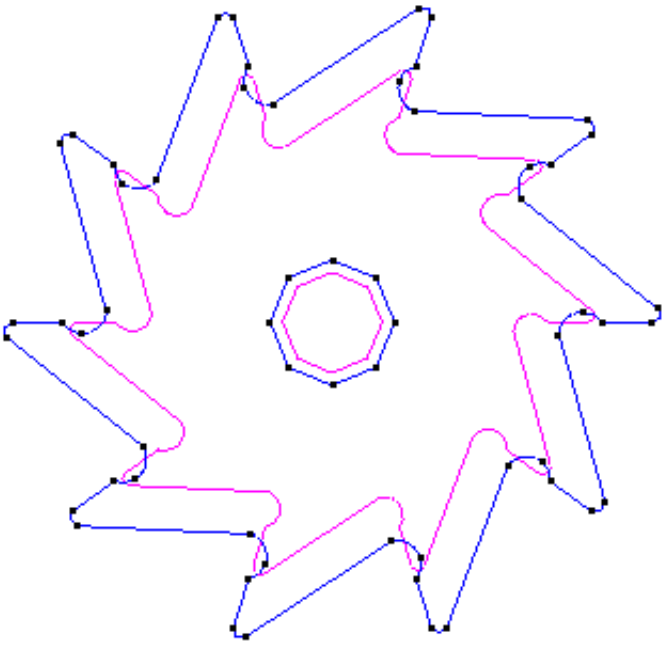


Figure 20. The model part with the hole in it

3.4 Creating volumes from surfaces

The mechanical part to be constructed is composed of two volumes: the volume of the wheel (defined by the profile), and the volume of the axle, which is a prism with an octagonal base that fits into the hole in the wheel. Creating this prism will be the first step of this stage. It will be created in a new layer that we will name "prism".

3.4.1 Creating the "prism" layer and translating the octagon to this layer

1 . In the **Layers** window, type the name of the new layer and click **New**.
2 . Select the "prism" layer and click **Layer To use** to choose it as the activated layer.
3 . Choose **Lines** in the **Sent To** menu in the **Layers** window. Select the lines that define the octagon. Press **ESC** to conclude the selection.

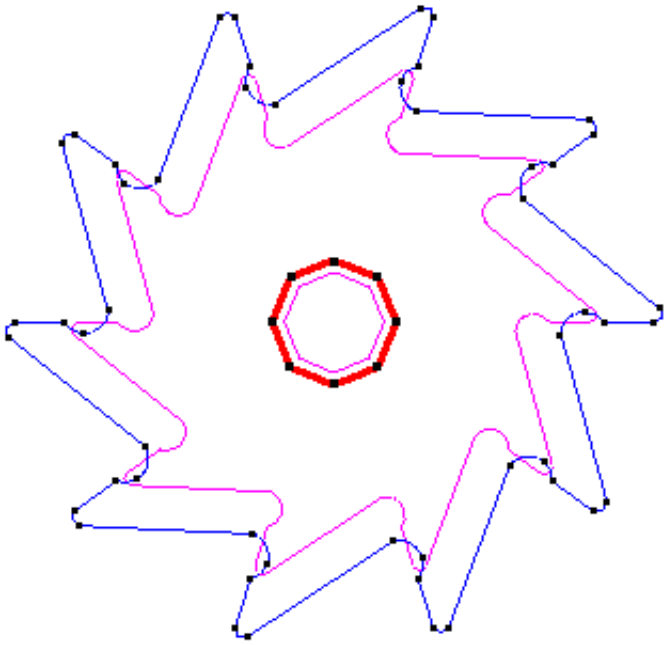


Figure 21. The lines that form the octagon

4 . Select the "profile" layer and click **Off** to deactivate it.

3.4.2 Creating the volume of the prism

1 . First copy the octagon a distance of -50 units relative to the surface of the wheel, which is where the base of the prism will be located. In the **Copy** window, choose **Translation** and **Lines**. Since we want to translate 50 units, enter two points that define the vector of this translation, for example (0, 0, 0) and (0, 0, 50). (Make sure that the Multiple Copies value is 1, since the last time the window was used its value was 9).

2 . Choose **Select** and select the lines of the octagon. Press **ESC** to conclude the selection.

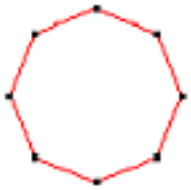


Figure 22. Selection of the lines
that form the octagon

3 . Since the Z axis is parallel to the user's line of vision, the perspective must be changed to visualize the result. To do this, use the **Rotate Trackball** tool, which is located in the GiD Toolbox and in the mouse menu.

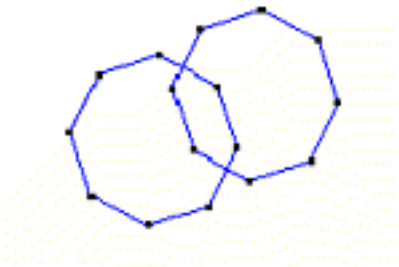


Figure 23. Copying the octagon and changing the perspective

4 . Choose **Geometry->Create->NURBS surface->By contour**. Select the lines that form the displaced octagon and press **ESC** to conclude the selection. Again, press **ESC** to exit the function of creating the surfaces.

Figure 24. The surface created on the translated octagon

5 . In the **Copy** window, choose **Translation** and **Surfaces**. Make a translation of 110 units. Enter two points that define a vector for this translation, for example (0, 0, 0) and (0, 0, -110).

6 . To create the volume defined by the translation, select **Do Extrude Volumes** in the **Copy** window.

7 . Click **Select** and select the surface of the octagon. Press **ESC**. The result is shown in Figure 25.

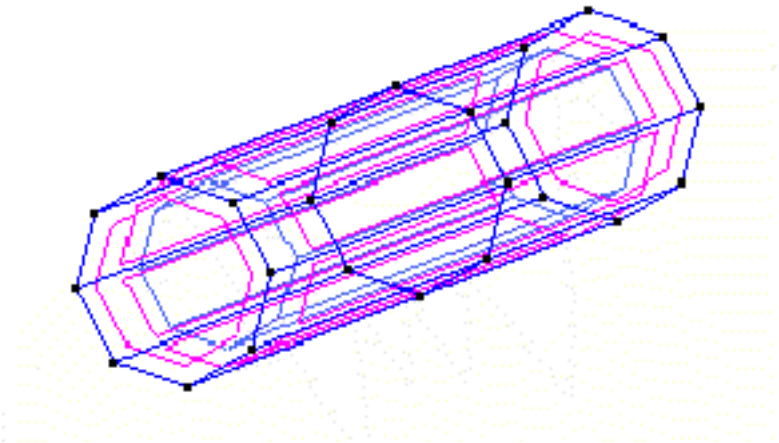


Figure 25. Creation of the volume of the prism

8 . Choose the option **Render->Flat** from the mouse menu to visualize a more realistic version of the model. Then return to the normal visualization using **Render->Normal**.

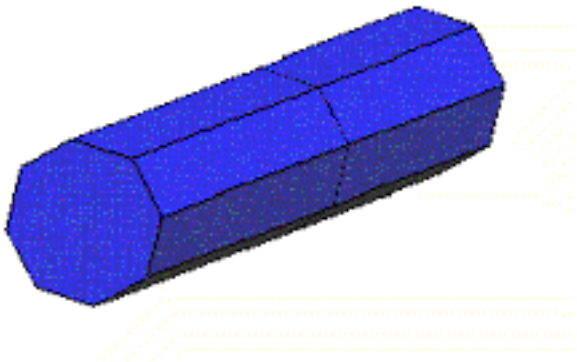


Figure 26. Visualization of the prism with the option RenderFlat.

NOTE: The **Color** option in the **Layers** window lets you define the color and the opacity of the selected layer. This color is then used in the rendering of elements in that layer.

3.4.3 Creating the volume of the wheel

1 . Visualize the "profile" layer and activate it. The volume of the wheel will be created in this layer. Deactivate the "prism" layer in order to make the selection of the entities easier.

2 . In the **Copy** window, choose **Translation** and **Surfaces**. A translation of 10 units will be made. To do this, enter two points that define a vector for this translation, for example (0, 0, 0) and (0, 0, -10).

3 . Choose the option **Do Extrude Volumes** from the **Copy** window. The volume that is defined by the translation will be created.

4 . Make sure that the **Maintain Layers** option is not checked, hence the new entities created will be placed in the layer to use; otherwise, the new entities are copied to the same layers as their originals

5 . Click **Select** and select the surface of the wheel. Press **ESC**.

6 . Select the two layers and click them **On** so that they are visible.

7 . Choose **Render->Flat** from the mouse menu to visualize a more realistic version of the model (Figure 27).

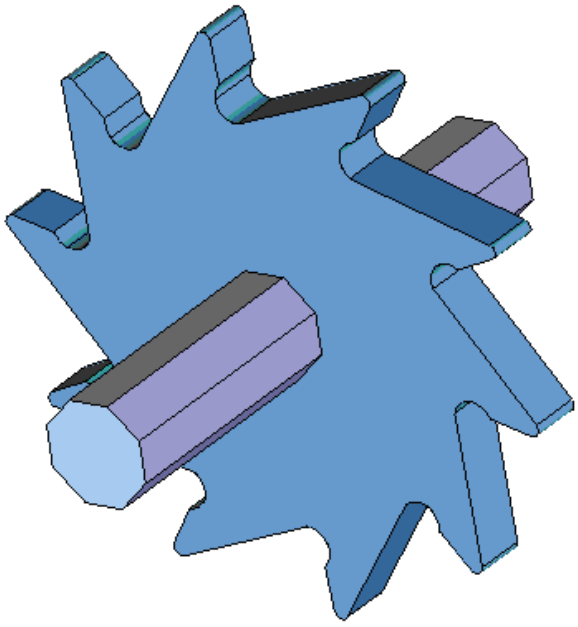


Figure 27. Image of the wheel

3.5 Generating the mesh

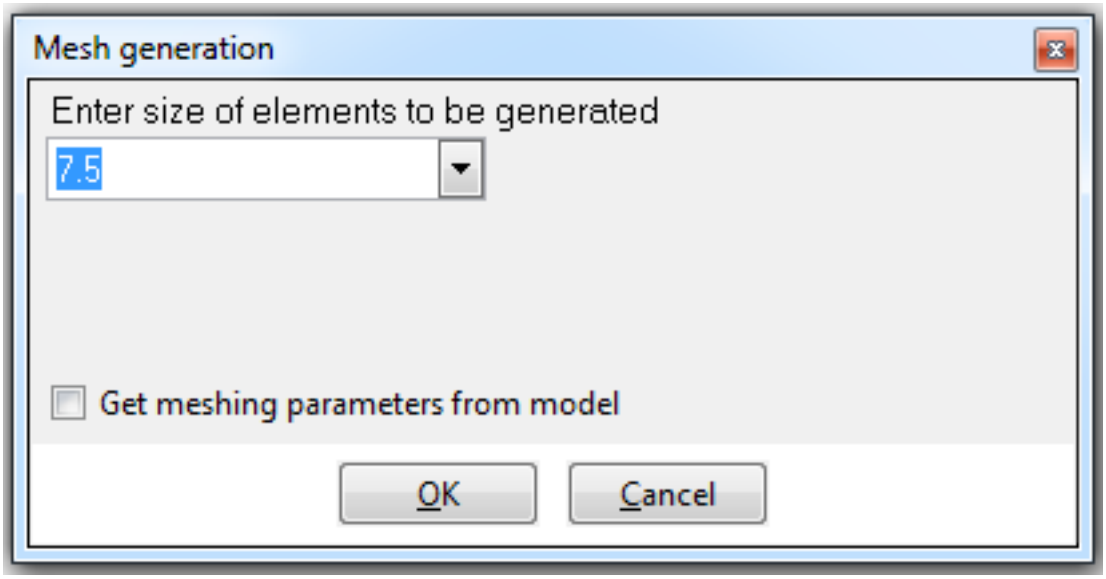
Now that the part has been drawn and the volumes created, the mesh may be generated. First we will generate a simple mesh by default.

Depending on the form of the entity to be meshed, GiD performs an automatic correction of the element size. This correction option, which by default is activated, may be modified in the **Meshing** card of the **Preferences** window, under the option **Automatic correct sizes**. Automatic correction is sometimes not sufficient. In such cases, it must be indicated where a more precise mesh is needed. Thus, in this example, we will increase the concentration of elements along the profile of the wheel by following two methods: 1) assigning element sizes around points, and 2) assigning element sizes around lines.

3.5.1 Generating the mesh by default

1 . Choose **Mesh->Generate Mesh**.

2 . A window comes up in which to enter the maximum element size of the mesh to be generated (Figure 28). Leave the default value given by GiD unaltered and click **OK**.



3 . A window appears showing how the meshing is progressing. Once the process is finished, another window opens with information about the mesh that has been generated (Figure 29). Click **OK** to visualize the resulting mesh (Figure 30).

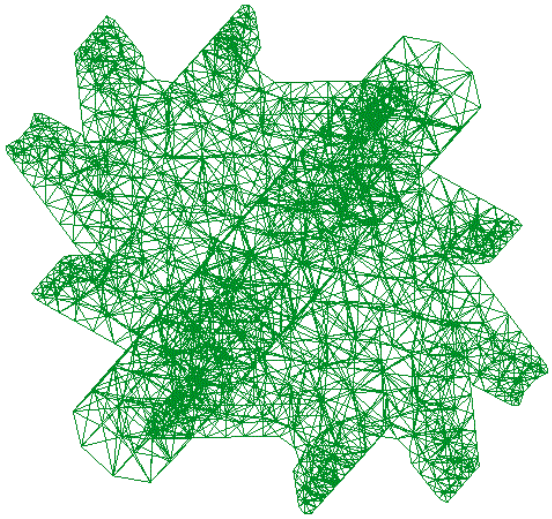


Figure 30. The mesh generated with default settings

4 . Use the **Mesh->View mesh boundary** option to see only the contour of the volumes meshed without the interiors (Figure 31). This visualization mode may be combined with the various rendering methods.

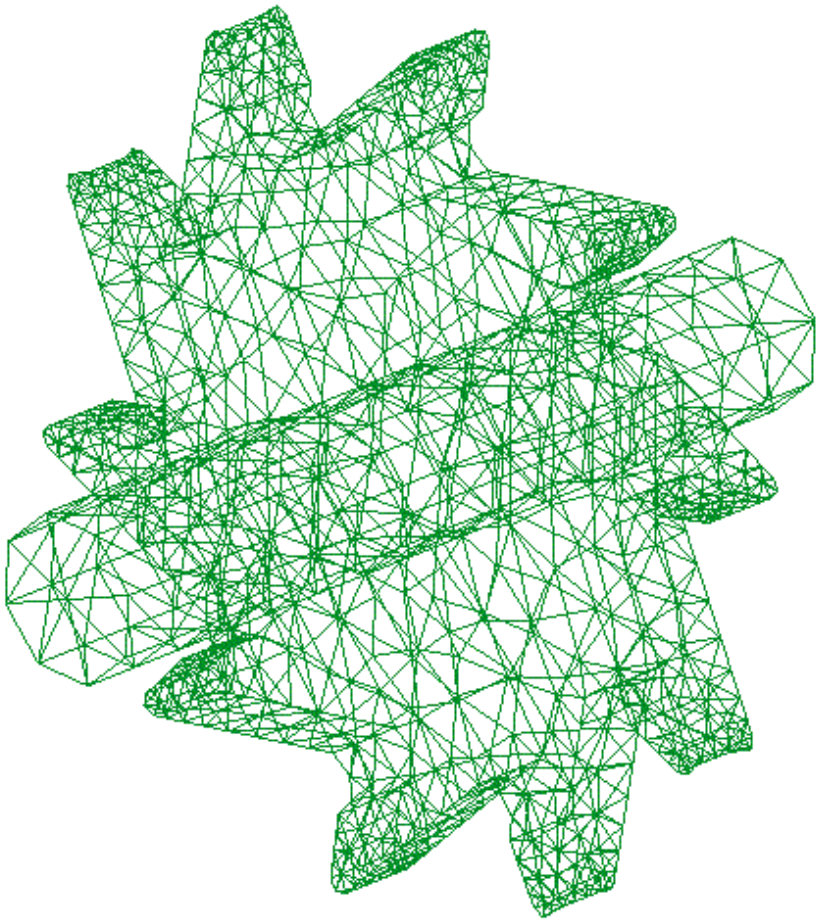


Figure 31. Mesh visualized with the Mesh->View mesh boundary option

5 . Visualize the mesh generated with the various rendering options in the **Render** menu, located in the mouse menu.

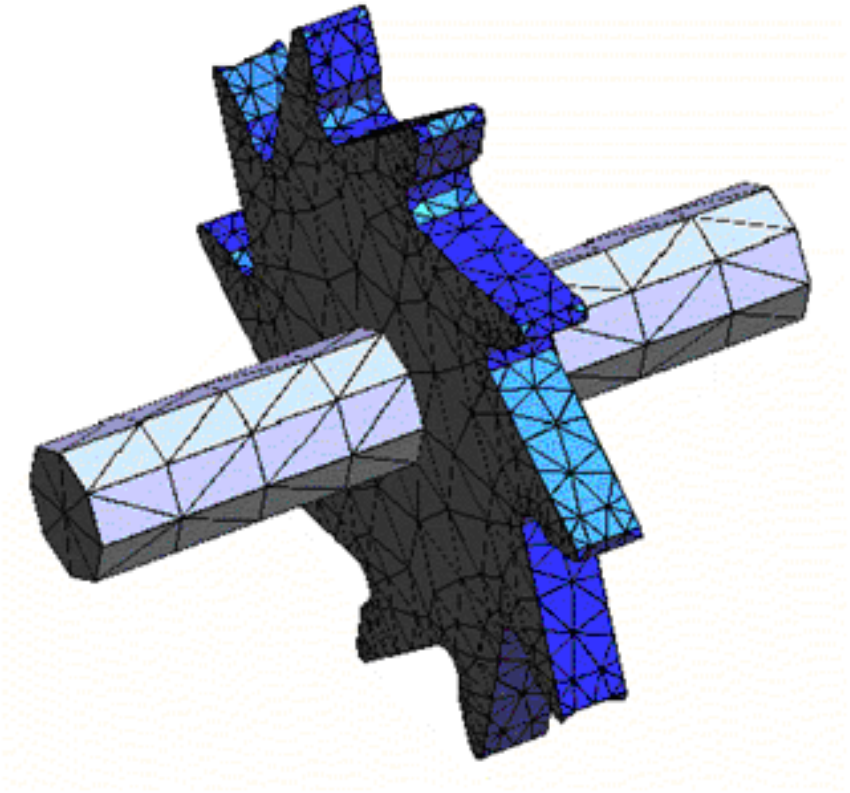


Figure 32. Mesh visualized with MeshView mesh boundary combined with RenderFlat.

6 . Choose **View->Mode->Geometry** to return to the normal visualization.

NOTE: To visualize the geometry of the model use **View->Mode->Geometry**. To visualize the mesh use **View->Mode->Mesh**.

3.5.2 Generating the mesh with assignment of size around points

1 . Enter **view rotate angle -90 90 ESC**[5] in the command line. This way we will have a side view.

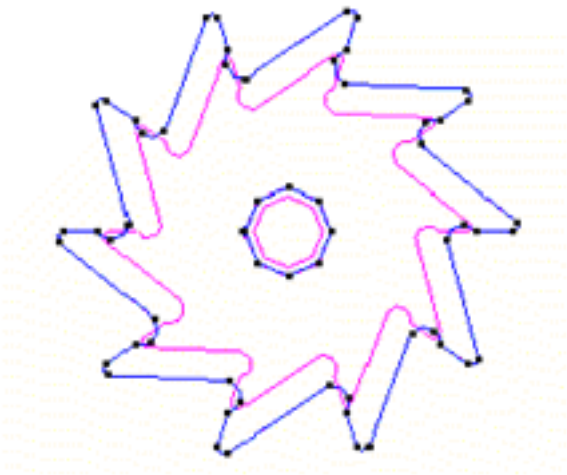


Figure 33. Side view of the part.

2 . Choose **Mesh->Unstructured->Assign sizes on points**. A window appears in which to enter the element size around the point to be selected. Enter 0.7.

3 . Select only the points on the wheel profile (Figure 34). One way of doing this is to select the entire part and then deselect the points that form the prism hole. Press **ESC** to conclude the selection process.

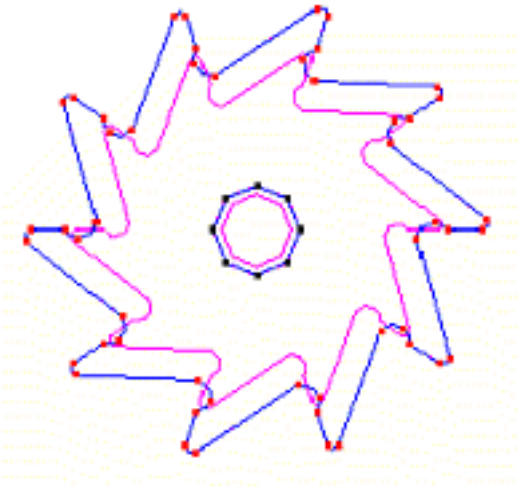


Figure 34. The selected points of the wheel profile

4 . Choose **Mesh->Generate mesh**.

5 . A window opens asking if the previous mesh should be eliminated (Figure 35). Click **Yes**. Another window appears in which the maximum element size should be entered. Leave the default value unaltered.

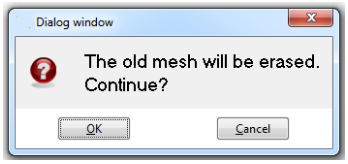


Figure 35

6 . A third window shows the meshing process. Once it has finished, click **OK** to visualize the resulting mesh (Figure 36).

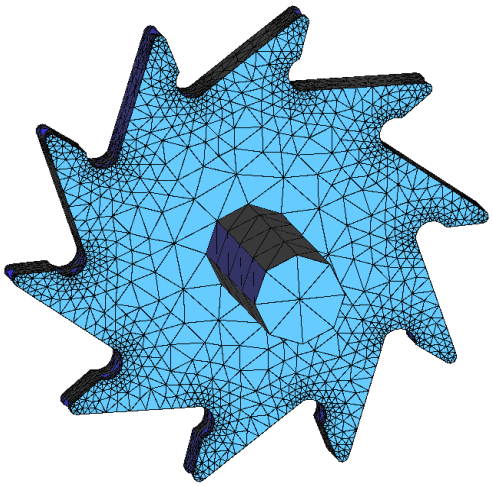


Figure 36. Mesh with assignment of sizes around the points on the wheel profile

7 . A greater concentration of elements has been achieved around the points selected.

8 . Choose **View->Mode->Geometry** to return to the normal visualization.

[5] Another option equivalent to **view rotate angle -90 90** is **RotatePlane XY**,located in the mousemenu.

3.5.3 Generating the mesh with assignment of size around lines

1 . Open the **Preferences** window, which is found in **Utilities**, and select the **Meshing** card. In this window there is an option called **Unstructured Size Transitions** which defines the size gradient of the elements. A high gradient number means a greater concentration of elements on the wheel profile. To do this, select a gradient size of 0.8. Click **Accept**.

2 . Choose **Mesh->Reset mesh data** to delete the previously assigned sizes.

3 . Choose **Mesh->Unstructured->Assign sizes on lines**. A window appears in which to enter the element size around the lines to be selected. Enter size 0.7. Select only the lines of the wheel profile (Figure 37) in the same way as in previous section.

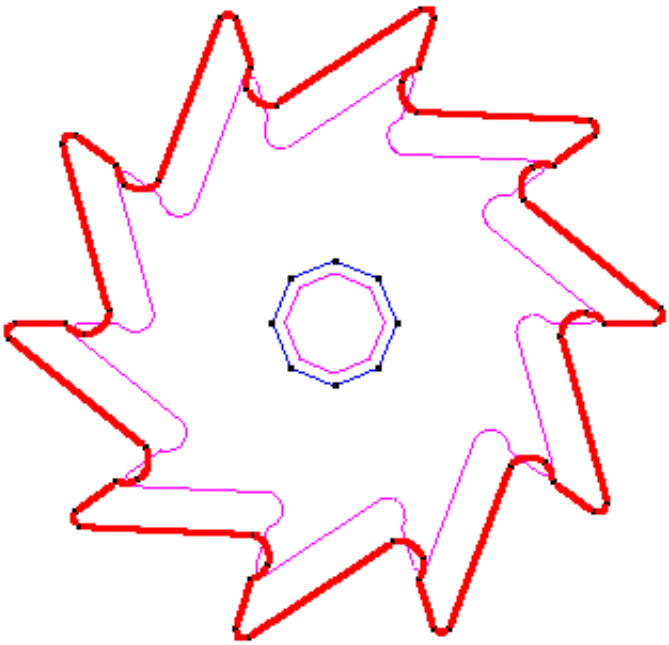


Figure 37. Selected lines of the wheel profile

4 . Choose **Mesh->Generate mesh**. A window appears asking if the previous mesh should be eliminated. Click **Ok**.

5 . Another window opens in which the maximum element size should be entered. Leave the default value unaltered.

6 . A greater concentration of elements has been achieved around the selected lines. In contrast to the case in previous section, this mesh is more accurate since lines define the profile much better than points do (Figure 38).

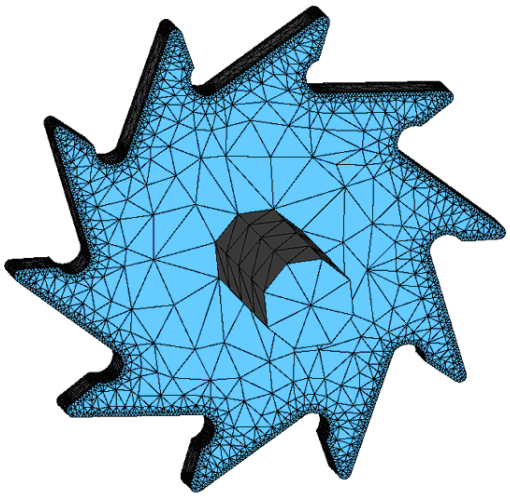


Figure 38. Mesh with assignment of sizes around lines

3.6 Optimizing the design of the part

The part we have designed can be optimized, thus achieving a more efficient product. Given that the part will rotate clockwise, reshaping the upper part of the teeth could reduce the weight of the part as well as increase its resistance. We could also modify the profile of the hole in order to increase resistance in zones under axle pressure.

To carry out these optimizations, we will use new tools such as NURBS lines. The final steps in this process will be generating a mesh and visualizing the changes made relative to the previous design.

This example begins with a file named "optimizacion.gid".

3.6.1 Modifying the profile

1 . First download the Preprocess Tutorial 1 from our web, **www.gidhome.com** Inside Support->Tutorials. Choose **Open** from the **Files** menu and open the file "optimizacion.gid".

2 . The file contents appear on the screen. In order to work more comfortably, select **Zoom In,** thus magnifying the image. This option is located both in the GiD Toolbox and in the mouse menu under **Zoom**.

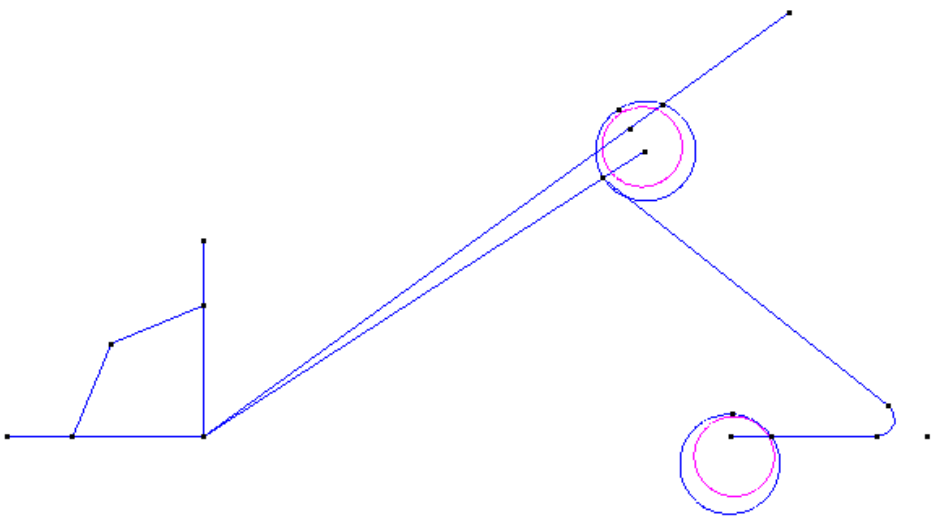


Figure 39. Contens of optimizacion.gid

3 . Make sure that the "aux" layer is activated.

4 . Choose **Geometry->Edit->Divide->Lines->Num Divisions** .This option divides a line into a specified number of segments.

5 . A window comes up in which to enter the number of partitions. Enter 8.

6 . Select the line segment from the upper part of a tooth (Figure 39) and press **ESC**.

7 . Using the option **Geometry->Create->Point**, and create a point with the coordinates (40, 8.5).

8 . Choose **Geometry->Create->NURBS line** to create a NURBS curve. The NURBS line to be created will pass through the two first points which have been created on dividing the line at point (40, 8.5) and by the two last points of the divided line.

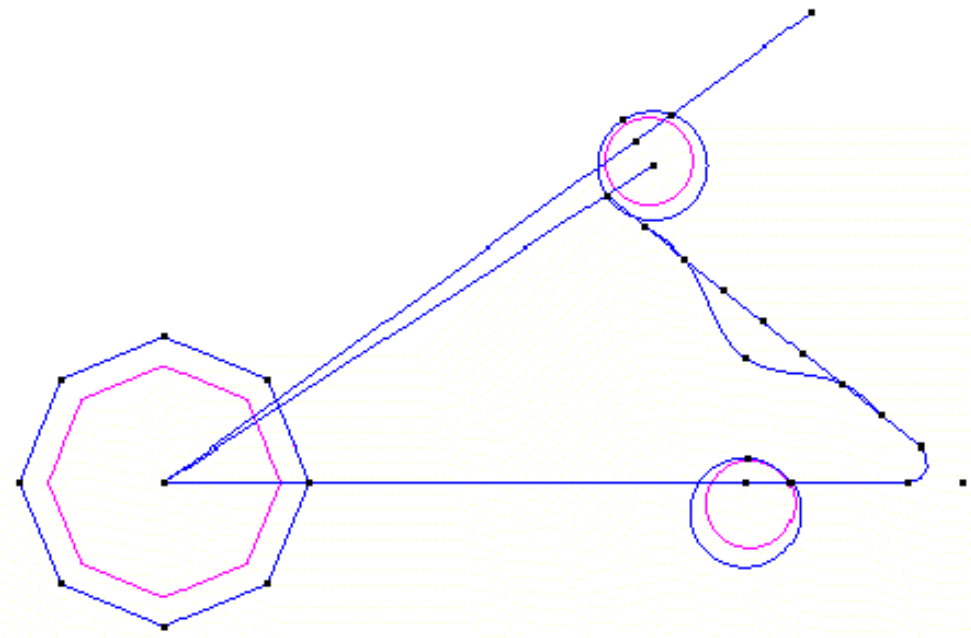


Figure 40. Optimizing the design

9 . Select the first point through which the curve will pass. To do this use **Join Ctrl-a**, located in **Contextual** in the mouse menu.

10 . One at a time, select the rest of the points except the last one. Use **Join Ctrl-a** each time in order to ensure that the line passes through the point.

11 . Before selecting the last point, choose **Last Point** in the **Contextual** menu. Then finish the NURBS line. The result is shown in Figure 40.

12 . Send the new profile (See Figure 41) to the "profile" layer and eliminate the auxiliary lines.

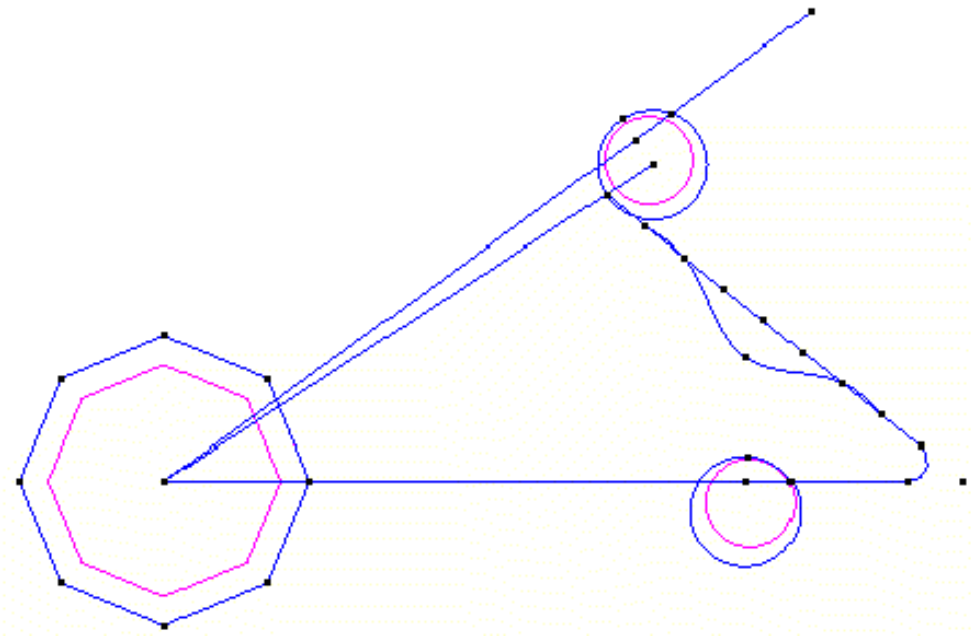


Figure 41. Optimizing the design

13 . Repeat the process explained in sections 2.11 and 2.12 to create the wheel surface: use the rotation tool to create the entire profile and, using **Geometry->Create->NURBS Surface->By contour**, select it to create a NURBS surface.

14 . Repeat the processes explained in section 3 (except section 3.1) and sections 4.1 and 4.2 to create the prismatic volume.

3.6.2 Modifying the profile of the hole

1 . Move the lines of the octagon placed in the profile surface to the "profile" layer (with the **Send To** button).

2 . Click **Off** the "prism" layer. Hiding it simplifies the space on the screen.

3 . Choose **Geometry->Create->Object->Circle**.

4 . Enter (-10.5, 0) as the center point. Enter a normal to the **XY** plane (Positive Z) and a radius of 1.5.

5 . From the Toolbox, use the **Delete->Surfaces** tool to delete the surface of the circle so that only the line is left. This way the **Geometry->Edit-> Intersection->Multiple Lines** option may be used to intersect the circle (circumference). Select only the circle and the two straight lines that intersect it.

6 . Choose **Copy** from the **Utilities** menu and make seven copies (**Multiple copies**=7), rotating the circle -45 degrees.

7 . Using the intersection options, delete the auxiliary lines leaving only the valid lines, thus obtaining the new profile of the hole. The result is illustrated in Figure 42.

8 . Create the hole in the surface of the wheel using **Geometry->Edit->Hole NURBS Surface** (the result is shown in Figure 43).

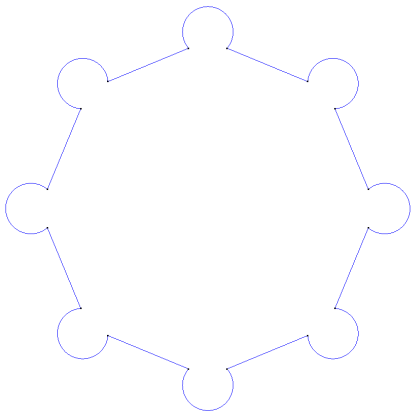


Figure 42. The new hole profile

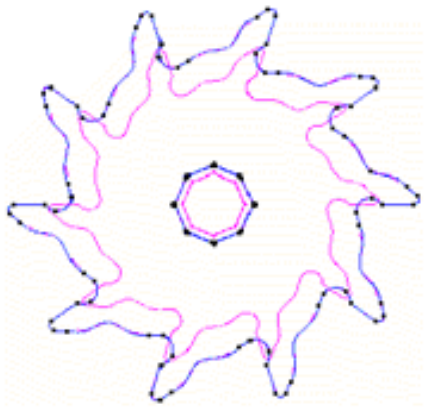


Figure 43. The surface of the new optimized design

3.6.3 Creating the volume of the new design

Repeat the same process as in section 4.3:

1 . In the **Copy** window, choose **Translation** and **Surfaces**. Enter two points that define a translation of 10 units, for example (0, 0, 10) and (0, 0, 0). (Make sure that the Multiple Copies value is 1).
2 . Choose **Do Extrude Volume** in the **Copy** window.
3 . Click **Select** and select the surface of the wheel. Press **ESC**.

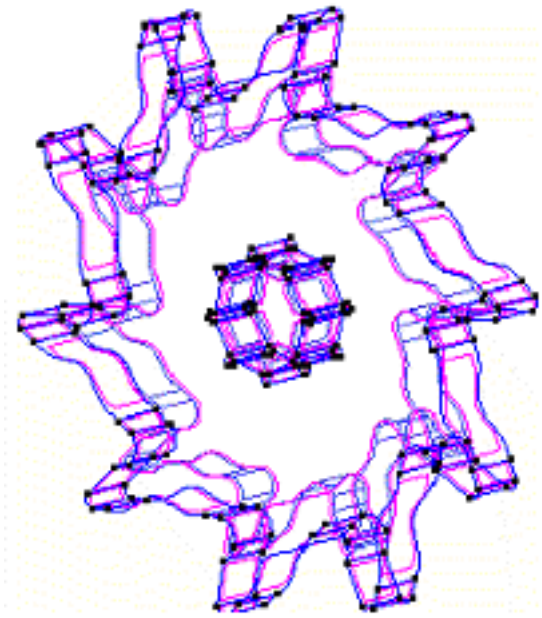


Figure 44. The volume of the optimized design

4 . Click **On** the "prism" layer.

3.7 Generating the mesh for the new design

Generating the mesh for the optimized design is more complex. In this geometry it is especially important to obtain a precise mesh on the surfaces around the hole and on the surfaces of the teeth.

Initially, we will generate a simple mesh by default. Then we will generate a mesh using Chordal Error[6] to obtain a more accurate result.

[6] The Chordal Error is the distance between each element generated by the meshing process and the real profile.

3.7.1 Generating a mesh for the new design by default

1 . Choose the option **Mesh->Generate mesh**.

2 . A window appears in which to enter the maximum element size for the mesh to be generated. Leave the default value provided by GiD unaltered and click **OK**.

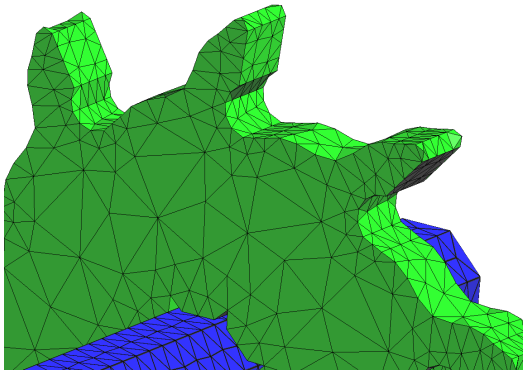


Figure 45. A detail of the mesh generated by default

3.7.2 Generating a mesh using "Chordal Error"

1 . Choose **Mesh->Unstructured->Sizes by Chordal error**.

2 . Provide the values shown in figure 46.

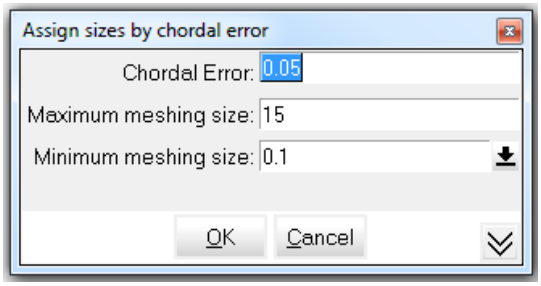


Figure 46. Chordal error window.

3 . Choose **Mesh->Generate mesh.**

4 . A greatly improved approximation has been achieved in zones containing curves and, more specifically, along the wheel profile and the profile of the hole (see Figure 47).

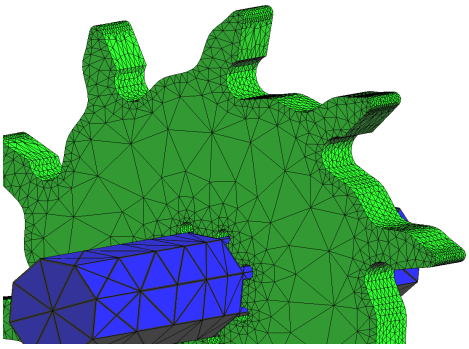


Figure 47. A detail of the mesh generated using Chordal Error

4 IMPLEMENTING A COOLING PIPE

ADVANCED 2D & 3D TECHNIQUES AND MESHING

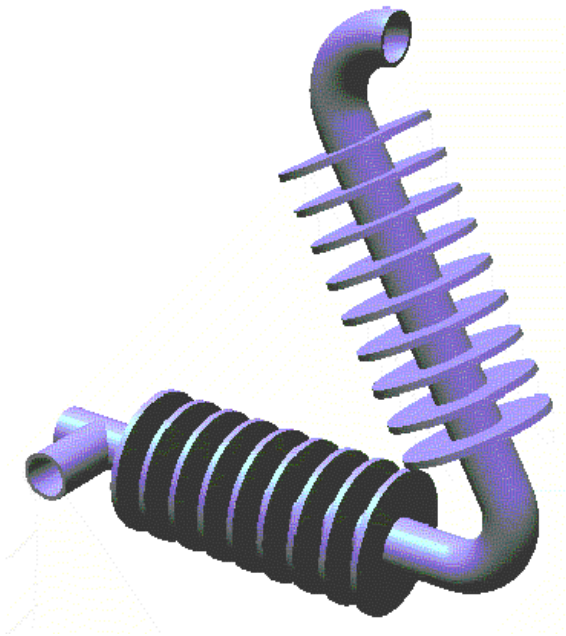
IMPLEMENTING A COOLING PIPE

This case study shows the modeling of a more complex piece and concludes with a detailed explanation of the corresponding meshing process. The piece is a cooling pipe composed of two sections forming a 60-degree angle.

The modeling process consists of four steps:

- Modeling the main pipes
  - Modeling the elbow between the two main pipes, using a different file
    - Importing the elbow to the main file
    - Generating the mesh for the resulting piece

At the end of this case study, you should be able to use the CAD tools available in GiD as well as the options for generating meshes and visualizing the result.



4.1 Working by layers

Various auxiliary lines will be needed in order to draw the part. Since these auxiliary lines must not appear in the final drawing, they will be in a different layer from the one used for the finished model.

4.1.1 Creating two new layers

1 . Open the layer management window, which is found in the **Utilities->Layers** menu.

2 . Create two new layers called "aux" and "ok". Enter the name for each layer in the Layers window (Figure 1) and click **New**.

3 . Choose "aux" as the activated layer. To do this, click on "aux" to highlight it and then click on the **Layer To Use** button. (The name of the activated layer will appear next to the button, "aux" in this case.) From now on, all the entities created will belong to this layer.

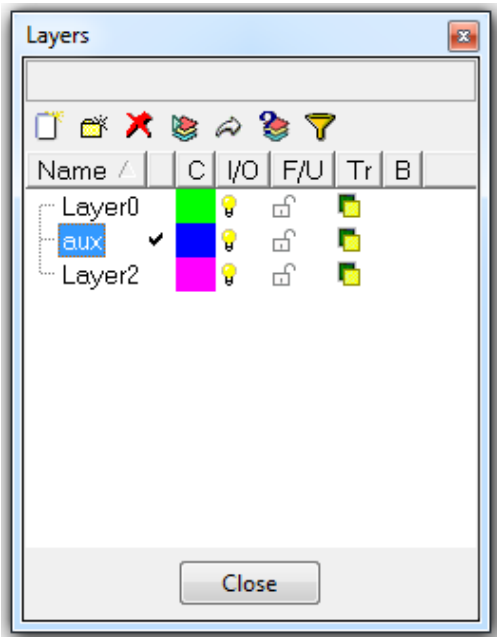


Figure 1. The Layers window.

4.2 Creating the auxiliary lines

The auxiliary lines used in this project are those that make it possible to determine the center of rotation and the tangential center, which will be used later to create the model.

4.2.1 Creating the axes

1 . Choose the **Line** option, by selecting **Geometry->Create->Straight line** [1].
2 . Enter the coordinate (0, 0) in the command line.
3 . Enter the coordinate (200, 0) in the command line.
4 . Press **ESC** [2] to indicate that the process of creating the line is finished.
5 . If the entire line does not appear on the screen, use the option **Zoom Frame,** which is located in the GiD Toolbox and in **Zoom** in the mouse menu.
6 . Again, choose **Line**. Draw a line between points (0, 25) and (200, 25). The result is shown in Figure 2.

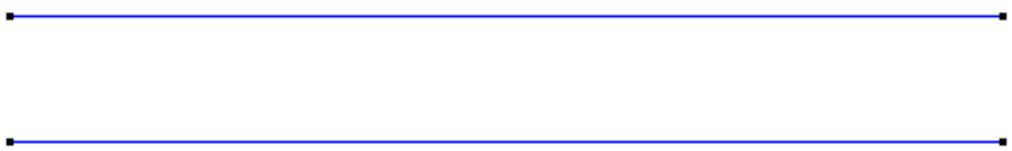


Figure 2

7 . Go to the **Copy** window (Figure 4), which is found in **Utilities->Copy**.
8 . Choose **Rotation** from the **Transformation** menu and **Lines** from the **Entities Type** menu.
9 . Enter an angle of -60 degrees and click on **Two dimensions**.
10 . Enter point (200, 0, 0) in **First Point**. This is the point that defines the center of rotation.
11 . Enter -60 degrees with option **Two dimensions** checked.
12 . Click **Select** to select the first line we drawn.

13 . After making the selection, press **ESC** (or **Finish** in the **Copy** window) to indicate that the selection of lines to be rotated is finished. The result is shown in Figure 3.

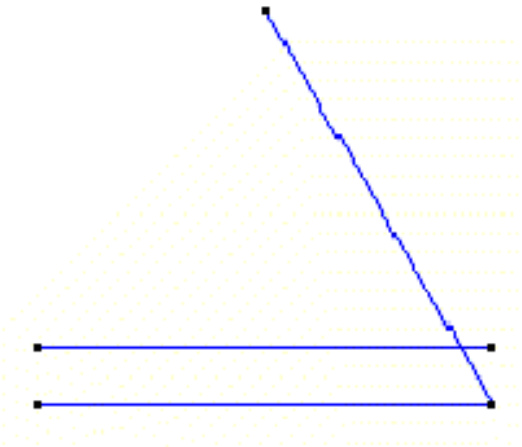


Figure 3. Creating the axes

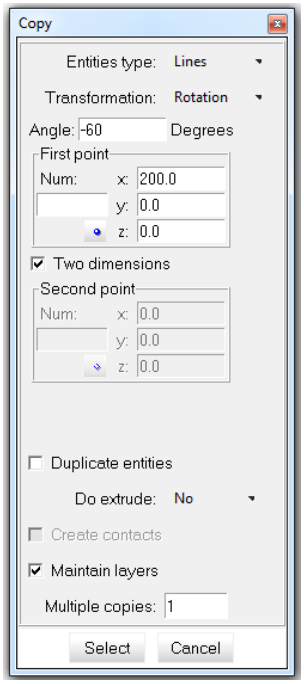


Figure 4. The Copy window

---

[1] This option can also be found in the GiD Toolbox.

[2] Pressing the **ESC** key is equivalent to pressing the center mouse button.

4.2.2 Creating the tangential center

1 . Choose the menu option **Geometry->Create->Straight line.** On the mouse menu, choose **Contextual** and use **Join Ctrl-a** to select points (0, 0) and (0, 25). Press **ESC.**

2 . In the **Copy** window, choose **Rotation** from the **Transformation** menu and **Lines** from the **EntitiesType** menu. Enter an angle of 120 degrees, and the coordinates (0, 25, 0) in **First point** also check the **Two dimensions** option. Finally select last line created.

3 . In the **Copy** window, choose **Translation** from the **Transformation** menu and **Lines** from the **EntitiesType** menu. The translation vector for the translation to be made is the line just created. As the first point of the translation, select the point farthest from this line segment. For the second point, select the other point of the line (Figure 5). To select an existing point, you must change the "No Join / Join" option located in the contextual submenu from 3rd mouse's button menu or use the shortcut (Control-a), notice that the cursor change from a cross to a box.

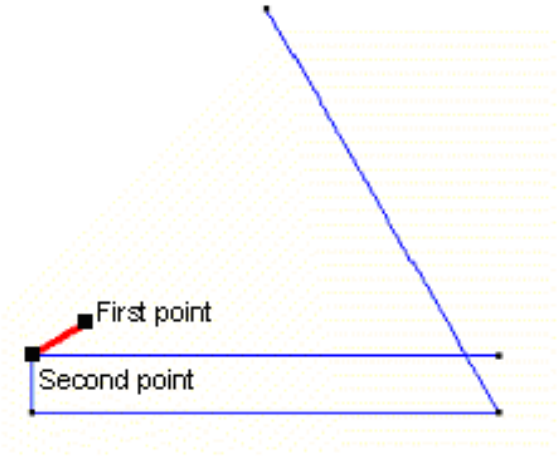


Figure 5. The line segment selected is the translation vector.

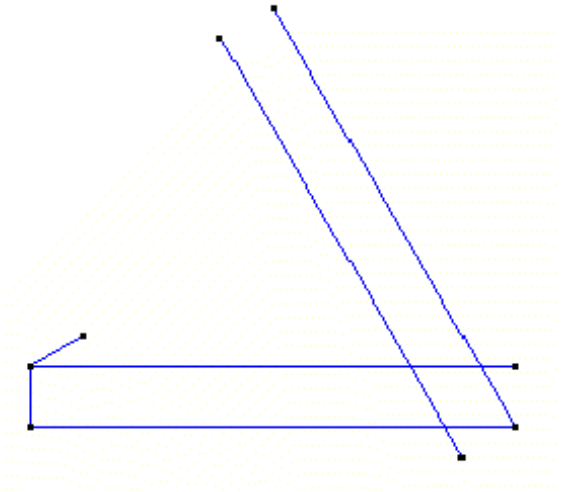


Figure 6. Result of the translation with copy

4 . Click **Select** to select the line segment that forms an angle of -60 degrees with the horizontal. Press **ESC** to indicate that the selection has been made.

5 . Choose **Geometry->Edit->Intersection->Line-line**.

6 . Select the two inner lines.

7 . Click **Yes** to confirm that there is an intersection and that, therefore, the shortest distance between the two entities is 0. The intersection between the two entities (lines) creates a point. This point will be the tangential center.

8 . Press **ESC** to indicate that the process of intersection between lines is finished.

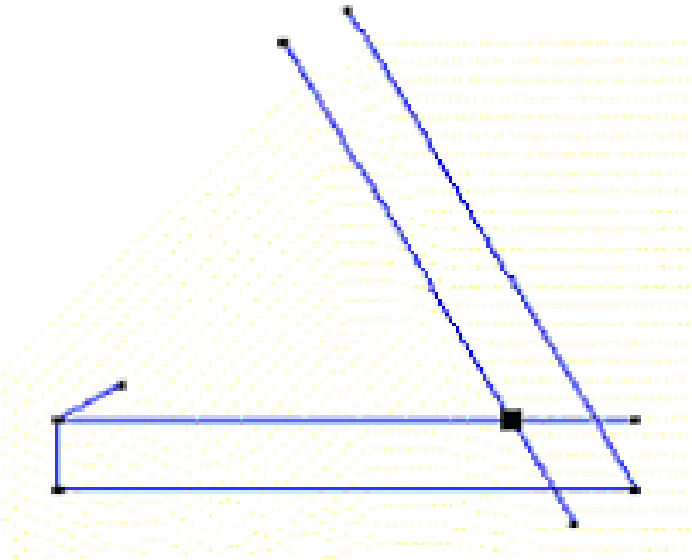


Figure 7. The auxiliary lines with tangencia center

NOTE: The **Undo** option allows you to undo the operations most recently carried out. If an error is made, go to **Utilities->Undo**; a window appears where you can select all the options to be eliminated.

4.3 Creating a component part

In this section the entire model, except the T junction, will be created. The model to be created is composed of two pipes forming a 60-degree angle. To start with, the first pipe will be created. This pipe will then be rotated to create the second pipe.

4.3.1 Creating the profile

1 . Select the **ok** layer and click on **Layer To use**. From now on, all entities created will belong to the **ok** layer.

2 . Choose the **Line** option, located in **Geometry->Create->Straight line.**

3 . Enter the following points: (0, 11), (8, 11), (8, 31), (11, 31), (11, 11) and (15, 11). Press **ESC** to indicate that the process of creating lines is finished.

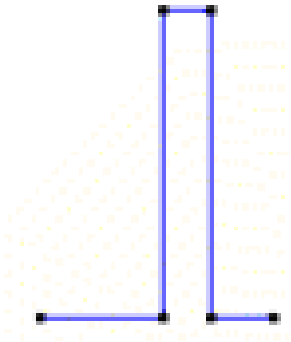


Figure 8. Profile of one of the disks around the pipe

4 . From the **Copy** window, choose **Lines** and **Translation**. A translation defined by points (0, 11) and (15, 11) will be made. In the **Multiple copies** option, enter 8 (the number of copies to be added to the original). Select the lines that have just been drawn.

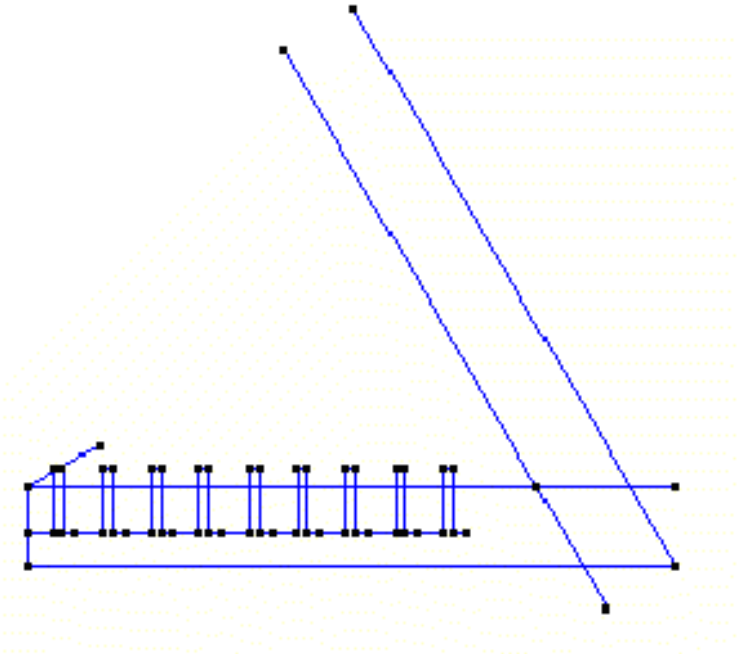


Figure 9. The profile of the disks using Multiple copies

5 . Choose **Line**, located in **Geometry->Create->Straight line.** Select the last point on the profile (at

the right part of the profile) using the option **Join Ctrl-a**, which is in the **Contextual** menu in the mouse menu. Now choose the option **No join Ctrl-a**. Enter point (200, 11). Press **ESC** to finish the process of creating lines.

6 . Again, choose the **Line** option and enter points (0, 9) and (200, 9). Press **ESC** to conclude the process of creating lines (Figure 10).

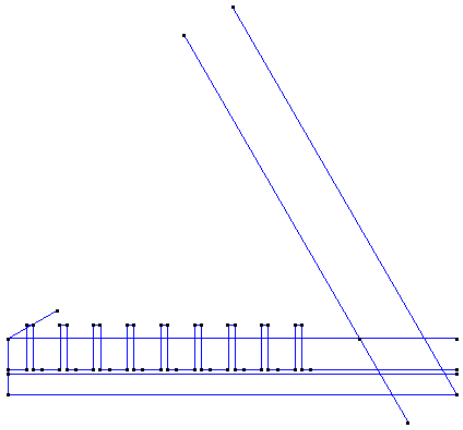


Figure 10. Creating the lines of the profile

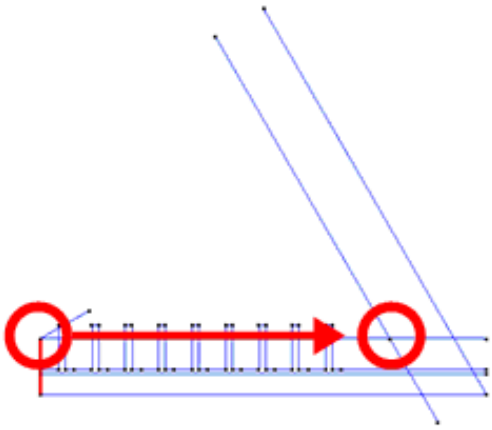


Figure 11. Copy of the vertical line segment starting at the origin of coordinates

7 . From the **Copy** window, choose **Lines** and **Translation**. As the first and second points of the translation, enter the points indicated in Figure 11. Click **Select** and select the vertical line segment starting at the origin of coordinates. Press **ESC**.

8 . Choose **Geometry->Edit->Intersection->Multiple lines.** Select the last two lines created and the vertical line segment coming down from the tangential center (see Figure 12). Press **ESC**.

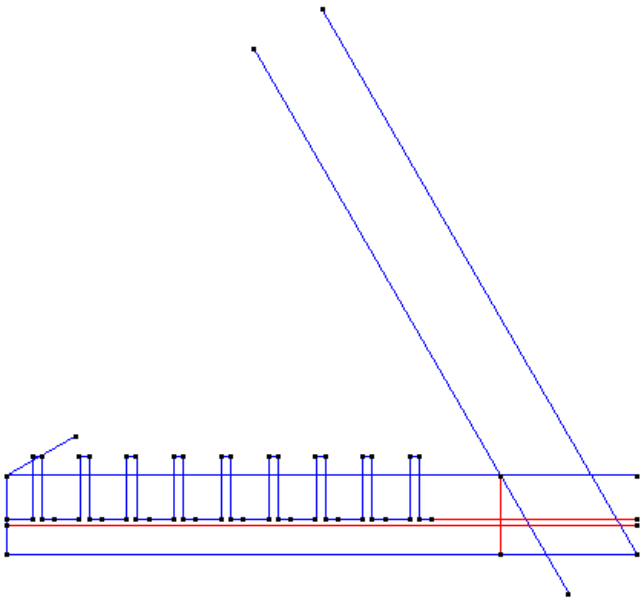


Figure 12. Selecting the lines to intersect

9 . Choose **Geometry->Delete->All Types**. (This tool may also be found in the GiD Toolbox) Select the lines and points beyond the vertical that passes through the tangential center. Press **ESC**. They will be deleted and the result should look like that shown in Figure 13.

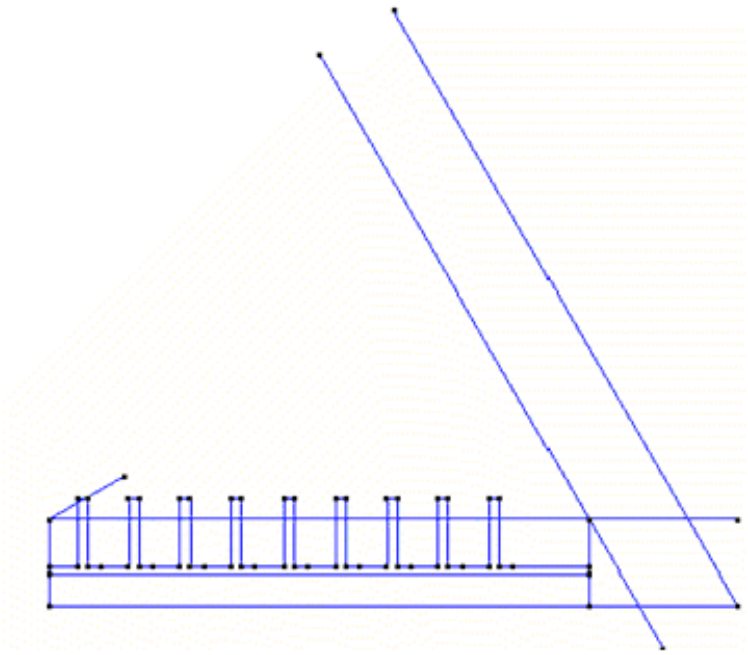


Figure 13. Profile of the pipe and the auxiliary lines

4.3.2 Creating the volume by revolution

1 . Rotation of the profile will be carried out in two rotations of 180 degrees each. This way, the figure will be defined by a greater number of points.

2 . From the **Copy** window, select **Lines** and **Rotation**. Enter an angle of 180 degrees and from the **Do extrude** menu, select **Surfaces**. The axis of rotation is that defined by the line that goes from point (0, 0) to point (200, 0). Enter these two points as the **First Point** and **Second Point**. Be sure to enter 1 in **Multiple Copies**.

3 . Click **Select.** For an improved view when selecting the profile, click **Off** the "aux" layer. Press **ESC** when the selection is finished. The result should be that illustrated in Figure 14.

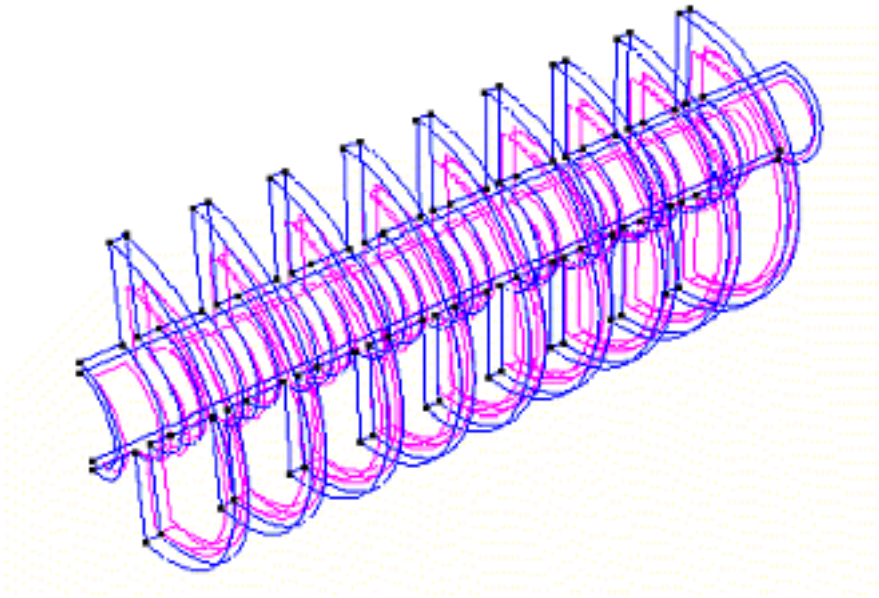


Figure 14. Result of the first step in the rotation (180 degrees)

4 . Repeat the process, this time entering an angle of –180 degrees.

5 . To return to the side view (elevation), choose **Rotate->Plane XY**.

6 . Choose **Render->Flat** from the mouse menu to visualize a more realistic version of the model. Return to the normal visualization with **Render->Normal**. This option is more comfortable to work with.

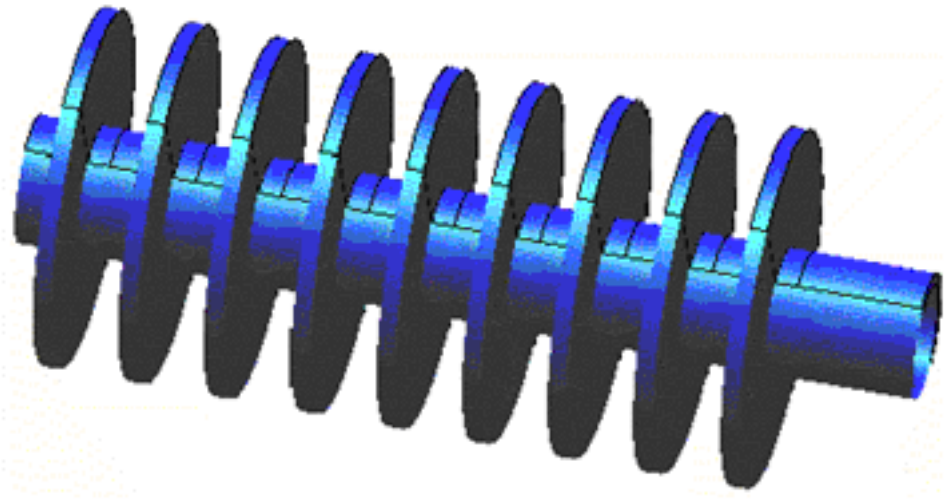


Figure 15. The pipe with disks, created by rotating the profile.

NOTE: To select the profile once the first rotation has been performed, first select all the lines and then delete those that do not form the profile. Use the option **Rotate->Trackball** from the mouse menu to rotate the model and make the process of selection easier.

4.3.3 Creating the union of the main pipes

1 . Choose the **Zoom In** option from the mouse menu. Magnify the right end of the model.

2 . Make sure the "aux" layer is visible.

3 . From the **Copy** window, select **Lines** and **Rotation**. Enter an angle of 120 degrees and from the **Do extrude** menu, select **Surfaces**. Since the rotation may be done in 2D, choose the option **Two Dimensions**. The center of the rotation is the tangential center.

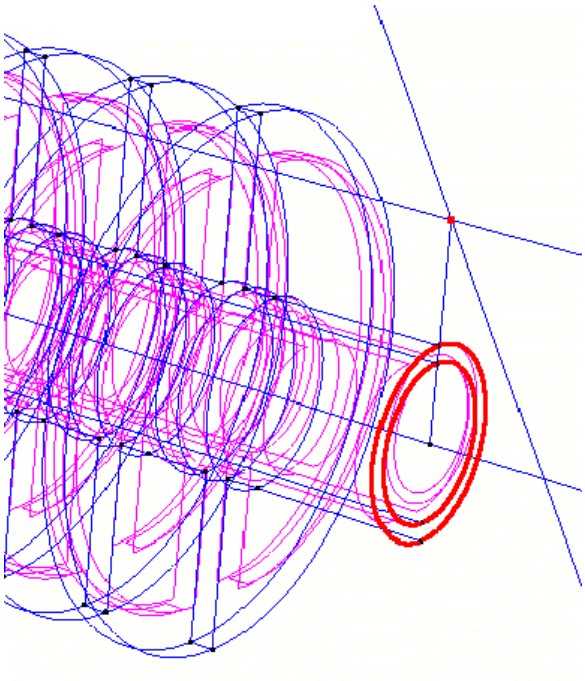


Figure 16. The magnified right end of the model, and the lines to be selected

4 . Click **Select** and select the four lines that define the right end of the pipe (see Figure 16). Press **ESC** when the selection is finished.

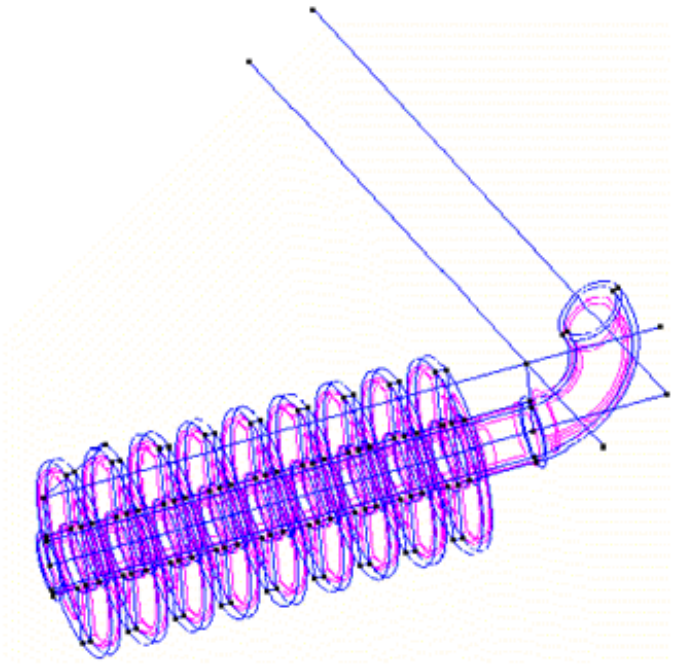


Figure 17. Result of the rotation

4.3.4 Rotating the main pipe

1 . From the **Copy** window, select **Surfaces** and **Rotation**. Enter an angle of -60 degrees. Since the rotation may be done in 2D, choose the **Two Dimensions** option. The center of the rotation is the intersection of the axes, namely point (200, 0). Ensure the **Do Extrude** menu is set to **No**.

2 . Click **Select** and select all the surfaces except those defining the elbow of the pipe. Press **ESC** when the selection is finished.

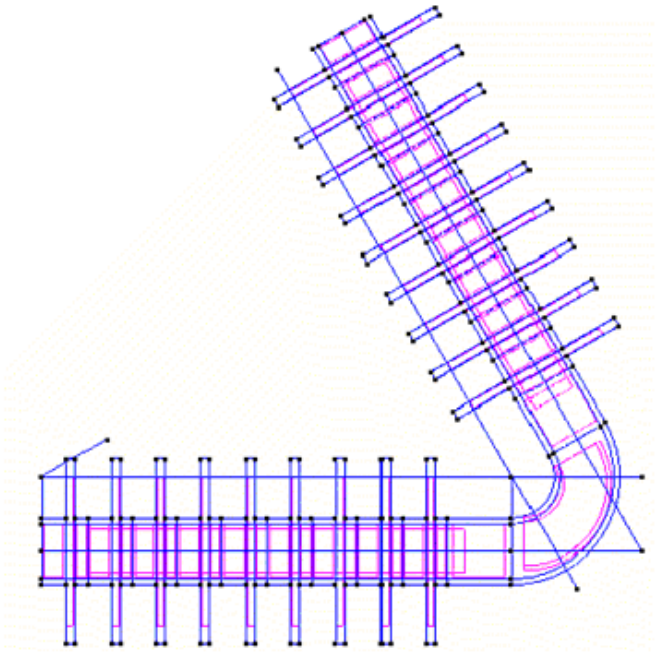


Figure 18. Geometry of the two pipes and the auxiliary lines

4.3.5 Creating the end of the pipe

1 . From the **Copy** window, select **Surfaces** and **Rotation**. Enter an angle of 180 degrees. Since the rotation may be done in 2D, choose the option **Two Dimensions**. The center of rotation is the upper right point of the pipe elbow. Make sure the **Do Extrude** menu is set to **No**.

2 . Click **Select** and select the surfaces that join the two pipe sections.

3 . In the **Move** window, select **Surfaces** and **Translation**. The points defining the translation vector are circled in Figure 19.

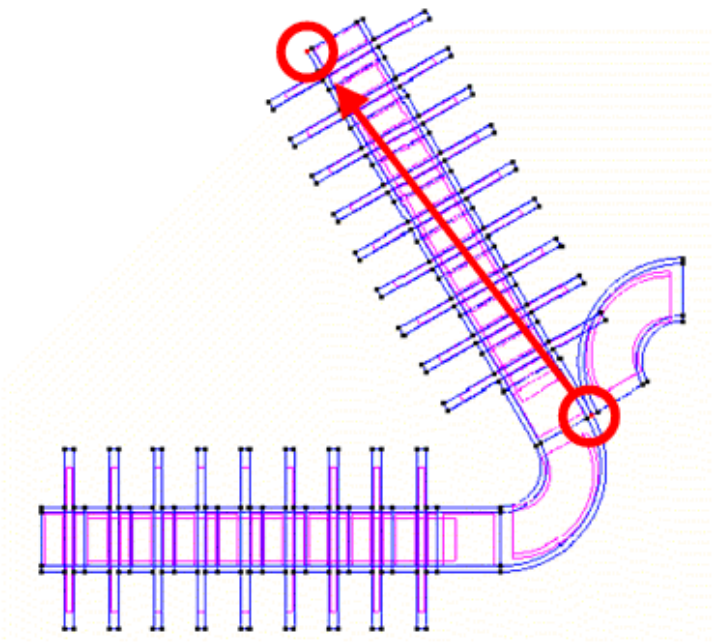


Figure 19. The circled points define the translation vector.

4 . Click **Select** and select the surfaces to be moved. Press **ESC.** The result should be as is shown in Figure 20.

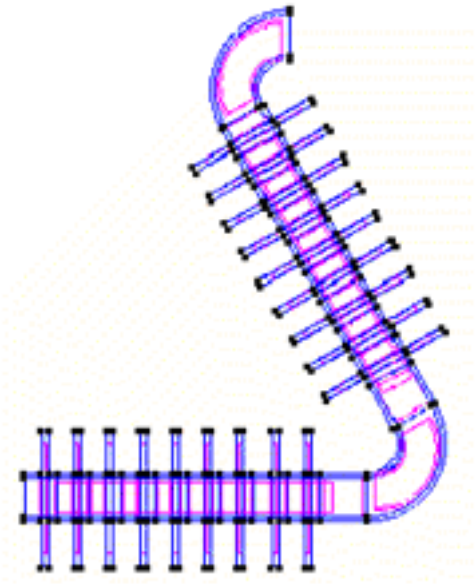


Figure 20. The final position of the translated elbow.

5 . Choose **Geometry->Create->NURBS Surface->By contour** and select the four lines that define the opening of the pipe (Figure 21). Press **ESC**.

6 . From the **Files** menu, choose **Save** in order to save the file. Enter a name for the file and click **Save**.

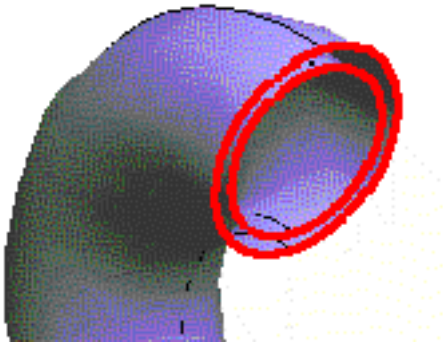


Figure 21. Opening at the end of the pipe

4.4 Creating the T junction

Now, an intersection composed of two pipe sections will be created in a separate file and the surfaces will be trimmed. Then this file will be imported to the original model to create the entire piece.

4.4.1 Creating one of the pipe sections

1 . Choose **Files->New**, thus starting work in a new file.

2 . Choose **Geometry->Create->Point** and enter points (0, 9) and (0, 11). Press **ESC** to conclude the creation of points.

3 . From the **Copy** window, select **Points** and **Rotation**. Enter an angle of 180 degrees and from the **Do extrude** menu, select **Lines**. The axis of rotation is the x axis. Enter two points defining the axis, one in **First Point** and the other in **Second Point**, for example, (0, 0, 0) and (100, 0, 0) (Figure 22).

4 . Click **Select** and select the two points just created.

5 . Repeat the process, this time entering an angle of –180 degrees, thus creating the profile of the pipe section with a second rotation of 180 degrees. The rotation could have been carried out in only one

rotation of 360 degrees. However, in the present example, each circumference must be defined between two points (Figure 23).

Figure 22. The result of the first180-degree rotation

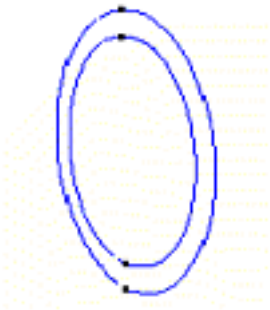


Figure 23. The combined result of the first rotation and the second rotation of 180 degrees, thus obtaining the profile of the pipe section.

6 . From the **Copy** window, choose **Lines** and **Translation**. In **First Point** and **Second Point**, enter the points defining the translation vector. Since the pipe section must measure 40 length units, the vector is defined by points (0, 0, 0) and (-40, 0, 0).

7 . From the **Do extrude** menu, choose the **Surfaces** option.

8 . Click **Select** to select the lines that define the cross-section of the pipe. Press **ESC** to conclude the selection process.

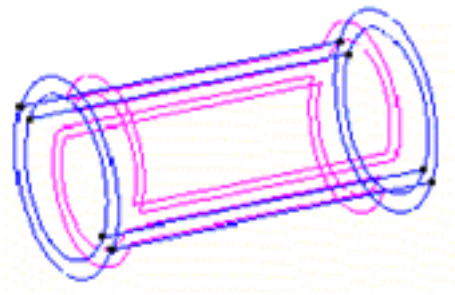


Figure 24. Creating a pipe by extruding circumferences

4.4.2 Creating the other pipe section

1 . Choose **Geometry->Create->Point** and enter points (-20, 9) and (-20, 11). Press **ESC** to conclude the creation of points.

2 . From the **Copy** window, select **Points** and **Rotation**. Enter an angle of 180 degrees and from the **Do extrude** menu, select **Lines**. Since the rotation can be done on the **xy** plane, choose **Two Dimensions**. The center of rotation is the coordinates (-20, 0, 0).

3 . Click **Select** and select the two points just created. Repeat the process, this time entering an angle of -180 degrees.

4 . From the **Copy** window, select **Lines** and **Translation**. In **First Point** and **Second Point** enter the points defining the translation vector. Since this pipe section must also measure 40 length units, the vector is defined by points (0, 0, 0) and (0, 0, 40).

5 . From the **Do extrude** menu, select the **Surfaces** option.

6 . Click **Select** to select the lines that define the cross-section of the second pipe. Press **ESC** to

conclude the selection.

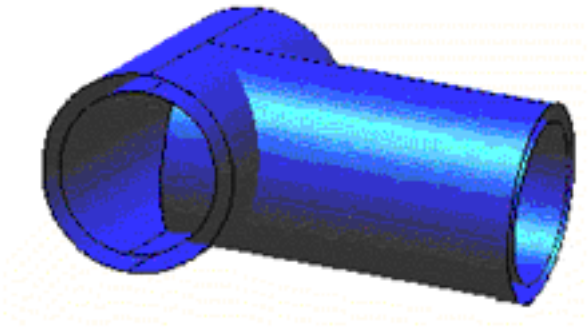


Figure 25. A rendering of the two intersecting pipes

4.4.3 Creating the lines of intersection

1 . Choose **Geometry->Edit->Intersection->Surface-surface**.

2 . Select the outer surfaces of each pipe, thus forming the intersection of the two surfaces selected.

3 . Repeat the process to obtain the four lines of intersection.

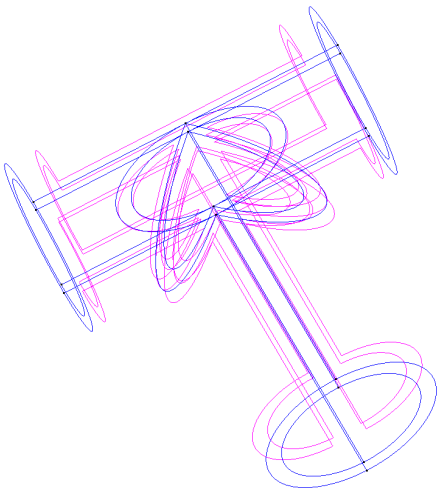


Figure 26. Creating lines of intersection between the surfaces

4.4.4 Deleting surfaces and lines

1 . Choose **Geometry->Delete->Surfaces** and select the small surfaces inside the first pipe. Press **ESC** to conclude the process of selection.

2 . Choose **Geometry->Delete->Lines**. Select the lines defining the end of the second pipe (foreground) that are still inside the first pipe (background). The result is shown in Figure 27.

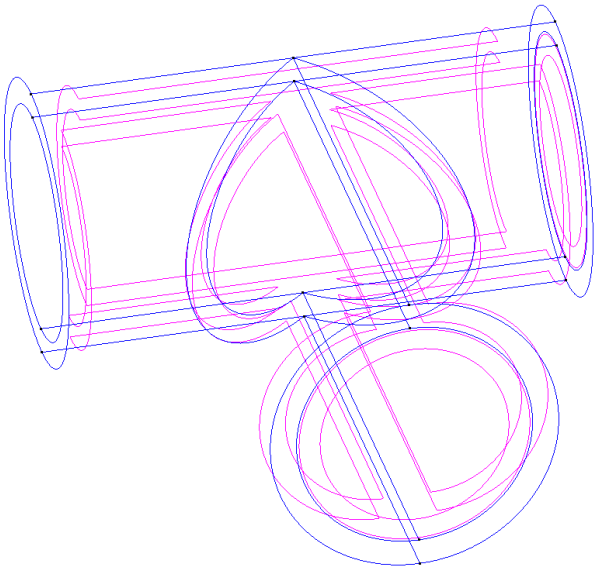


Figure 27. Final configuration after deleting surfaces and lines

4.4.5 Closing the volume

1 . The model now has three outlets. The two farthest from the origin of coordinates must be closed. The third will be connected to the rest of the piece when the T junction is imported.

2 . Choose **Geometry->Create->NURBS Surface->By contour** and then select the lines defining the outlet in the foreground of Figure 28. Press **ESC** (see Figure 28).

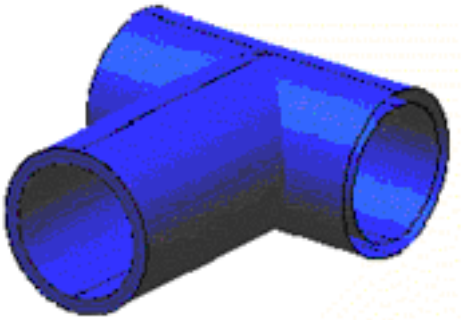


Figure 28. Creating a NURBS surface to close the outlet in the foreground

3 . Repeat the process for the other outlet to be closed.

4 . From the **Files** menu, select **Save** to save the file. Enter a name for the file and click **Save**.

4.5 Importing the T Junction to the main file

The two parts of the model have been drawn. Now they must be joined so that the final volume may be created and a mesh of the volume may be generated.

4.5.1 Importing a GiD file

1 . Choose **Open** from the **Files** menu. Select the file where the first part, created in section 3, was saved. Click **Open**.

2 . Choose **Files->Import->Insert GiD geometry** from the menu. Select the file where the second part, created in section 4, was saved. Click **Open**.

3 . The T junction appears. Bear in mind that the lines which define the end of the first pipe (background) of the T junction, and which have been imported, were already present in the first file.

Notice that the lines overlap. This overlapping will be remedied by collapsing the lines.

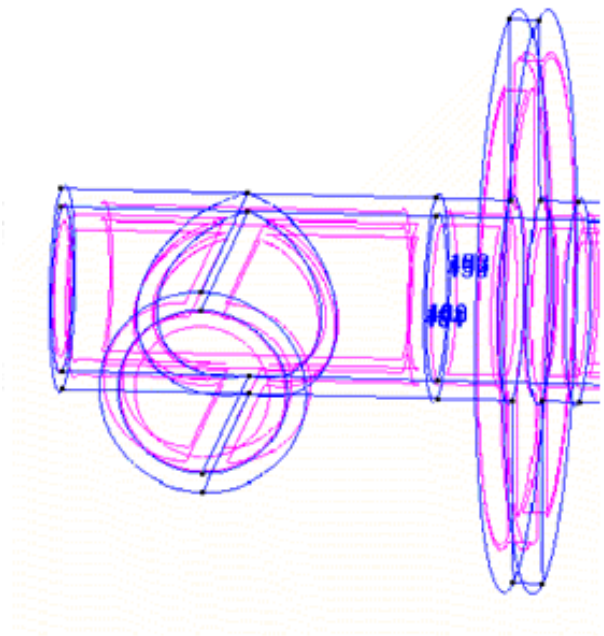


Figure 29. Importing the T junction file to the main file. Some points are duplicated and must be collapsed.

4 . Choose the option **Geometry->Edit->Collapse->Lines**. Select the overlapping lines and press **ESC**.

4.5.2 Creating the final volume

1 . Choose **Geometry->Create->Volume->By contour** and select all the surfaces that define the volume. Press **ESC** to conclude the selection process.

2 . Choose **Render->Smooth** to visualize a more realistic version of the model.

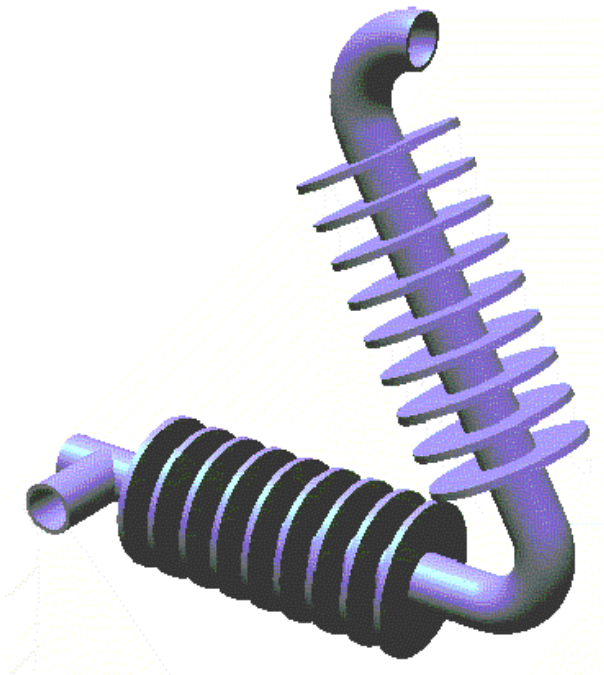


Figure 30. A rendering of the finished piece of equipment

4.6 Generating the mesh

Now that the model is finished, it is ready to be meshed. The mesh will be generated using **Chordal Error** in order to achieve greater accuracy in the discretization of the geometry. The chordal error is the distance between the element generated by the meshing process and the real profile of the model. By selecting a sufficiently small chordal error, the elements will be smaller in the zones with greater curvature.

4.6.1 Generating the mesh using Chordal Error

1 . Choose the option **Mesh->Unstructured->Sizes by Chordal error**.
2 . Enter 1 for the minimum element size.
3 . Enter 15 for the maximum element size.
4 . Enter 0.2 for the chordal error.
5 . Choose **Mesh->Generate mesh**.
6 . A window opens in which to enter the maximum element size of the mesh to be generated. Leave the default value provided by GiD unaltered and click **OK**.
7 . When the meshing process is finished, a window appears with information about the mesh that has been generated. Click **OK** to visualize the mesh.
8 . Choose **Mesh->View Mesh Boundary** to see only the contour of the volumes meshed but not the interiors.
9 . The visualization may be rendered using the various options in the **Render** menu, located in the mouse menu.

NOTE: By default GiD corrects element size depending on the form of the entity to mesh. This correction option may be deactivated or reactivated in the **Meshing** card in the **Preferences** window under the option **Automatic correct sizes**.

Figure 31. The mesh generated for the piece

4.6.2 Generating the mesh by assignment of sizes on surfaces

1 . Choose **Mesh->Unstructured->Assign sizes on surfaces**. A window opens in which to enter the element size for the surfaces to be selected. Enter size 1.

2 . Select the surfaces of the elbow.

3 . Choose **Mesh->Generate mesh**.

4 . A window appears asking whether the previous mesh should be eliminated. Click **Ok**.

5 . Another window appears in which the maximum element size should be entered. Leave the default value provided by GiD unaltered and click **OK**.

6 . Choose **Mesh->View Mesh Boundary** to see only the contour of the meshed volumes but not the interiors (Figure 32).

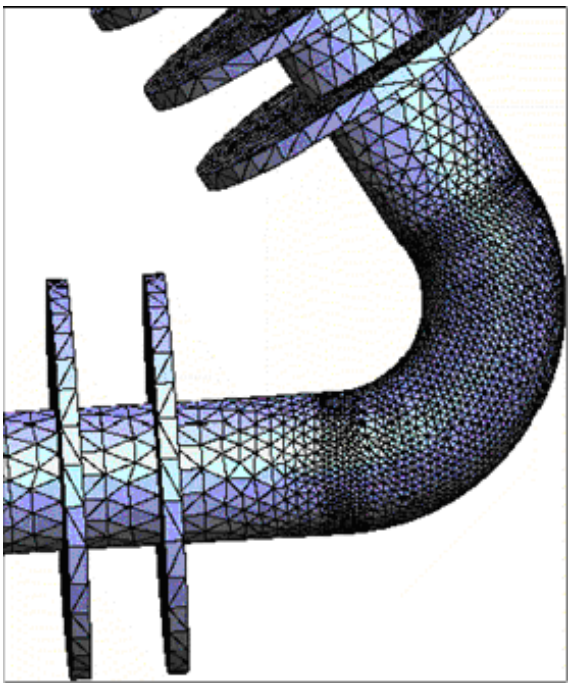


Figure 32. The mesh with a concentration of elements on the surfaces of the elbow

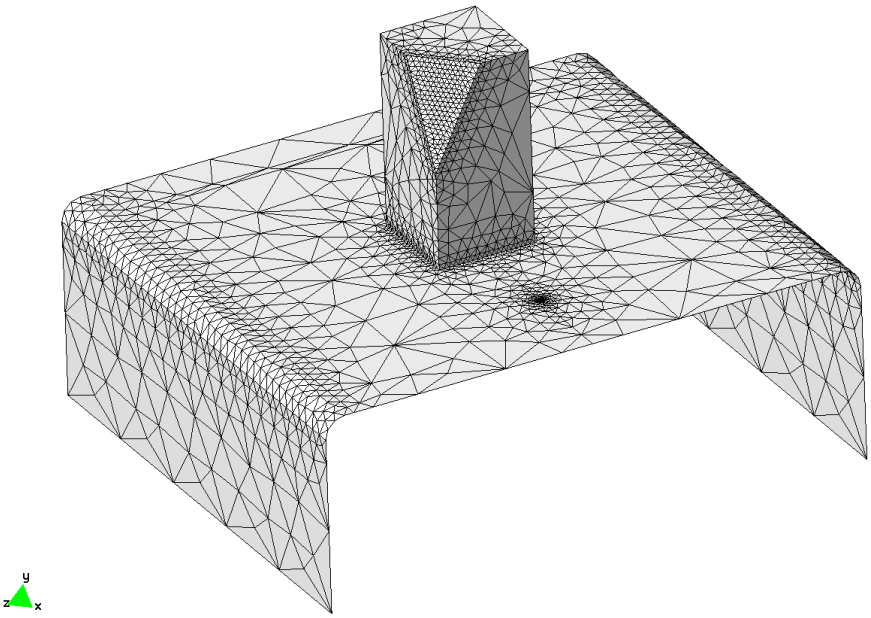
7 . A greater concentration of elements has been achieved on the selected surfaces.

5 ASSIGNING ELEMENT SIZES

ASSIGNING SIZES TO THE ELEMENTS OF A MESH

The objective of this example is to mesh a mechanical piece using the various options in GiD for assigning sizes to elements, and the different surface meshers available. In this example a mesh is generated for each of the following methods for assigning sizes, using different surface meshers:

- Assigning sizes around points
  - Assigning sizes around lines
    - Assigning sizes on surfaces
    - Assigning sizes with Chordal Error



5.1 Introduction

In order to carry out this example, start by opening the project "ToMesh4.gid". This project contains a geometry that will be meshed using four different methods, each one resulting in a different density of elements in certain zones.

5.1.1 Reading the initial project

- In the **Files** menu, select **Read** . Select the project "ToMesh4.gid" and click **Open**.
- The geometry appears on the screen. It is a set of surfaces.
- Select **Render->Flat** from the mouse menu[1].
- Select **Rotate->Trackball** from the mouse menu. (This tool is also available within the GiD Toolbox.) Make several changes in the perspective so as to get a good idea of the geometry of the object.
- Finally, return to the normal visualization, selecting **Render->Normal**. This mode is more user-friendly.

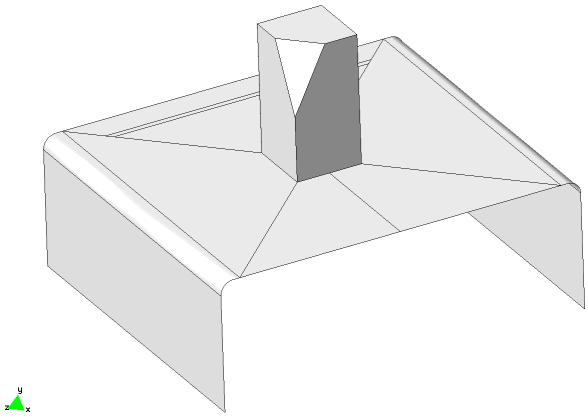


Figure 1. Contents of the project ToMesh.gid

---

[1] The mouse menu appears when the right mouse button is clicked.

5.2 Element-size assignment methods

GiD automatically corrects element sizes according to the shape of the entity to be meshed and its surrounding entities. This default option may be deactivated and reactivated by going to the **Mesh** menu, selecting **Preferences**[2], and then **Automatic correct sizes**[3].

Sometimes, however, this type of correction is not sufficient and it is necessary to indicate where on the mesh greater accuracy is needed. In these cases, GiD offers various options and methods allowing sizes to be assigned to elements.

Five examples are shown to illustrate the default method and the four other methods.

---

[2] The **Preferences** option can also be found in the Utilities menu.

[3] **Automatic correct sizes** automatically executes the options **Assign sizes->By geometry** and **Assign sizesCorrect sizes**.

5.2.1 Assignment using default options

1 . Select **Mesh->Generate Mesh**.
2 . window appears showing the maximum element size. Leave this default size unaltered and click **OK**.
3 . A meshing process window opens. Then another window appears with information about the mesh generated. Click **OK** to visualize the mesh (see Figure 2).

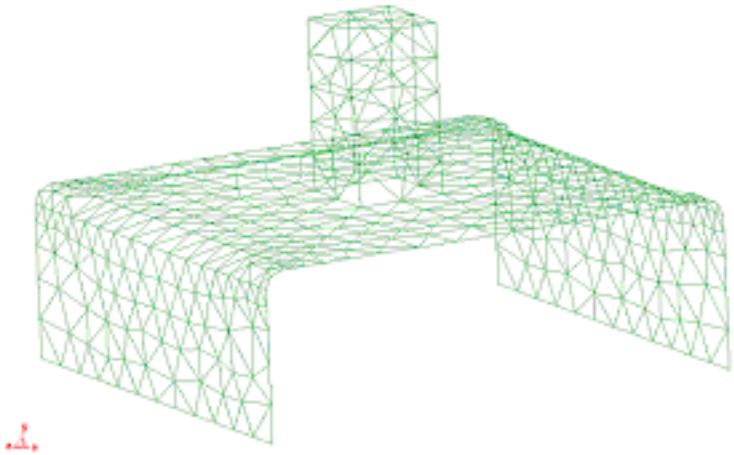


Figure 2. Meshing by default.

Note that in the zone highlighted in Figure 3, elements are smaller than in the rest of the model. This is because of the shape of the surface placed there. When all meshing preferences are set to their default levels, as for this example, the RFAST surface mesher is used. In this way, geometrical entities are meshed hierarchically: first of all lines are meshed, then the surfaces, and finally the volumes. The line elements size depends on the shape of surfaces (as can be seen in this example). Later on we will see an example using RJUMP mesher, where element sizes are distributed differently.

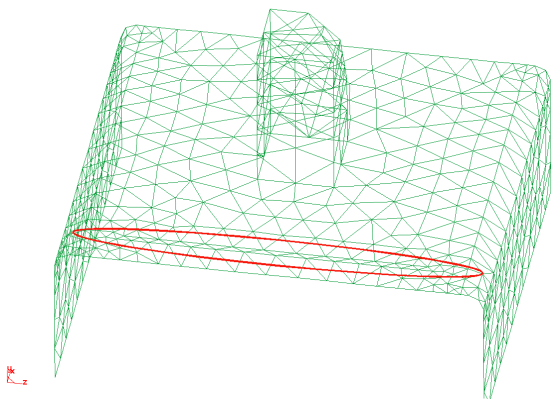


Figure 3. Meshing by default. Zone where elements are smaller because of the surface shape.

5.2.2 Assignment around points

1 . Select **Mesh->Unstructured->Assign size on points**. A window appears in which to enter the element size around the points to be chosen. Enter 0.1 and click **OK**.
2 . Select the point indicated in Figure 4. Press **ESC** [4] to indicate that the selection of points is finished.
3 . Select **Mesh->Generate Mesh**.
4 . A window opens asking whether the previous mesh should be eliminated. Click **Ok**.
5 . GiD then asks you to enter the maximum element size. Leave the default value unaltered and click **OK**.

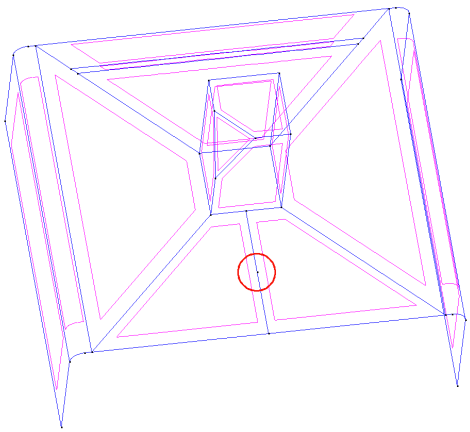


Figure 4. Geometry of the model. The point around which the mesh will be concentrated.

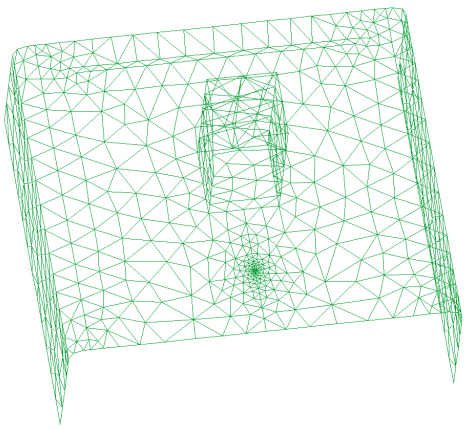


Figure 5. The mesh with a concentration of elements around the point.

6 . A concentration of elements appears around the chosen point, given the selected size (0.1) of these elements (see Figure 5).

7 . Go to **Utilities** and open **Preferences.** Click **Meshing**. In the window that appears there is the option **Unstructured Size Transitions**. This option defines the transition gradient of element sizes (size gradient), whose values are between 0 and 1. The greater the size gradient, the greater the change in space. To test this, enter the value 0.4 and click **Accept**.

8 . Again, select **Mesh->Generate Mesh**.

9 . A window opens asking whether the previous mesh should be eliminated. Click **Ok**.

10 . GiD then asks you to enter the maximum element size. Leave the default value unaltered and click **OK**.

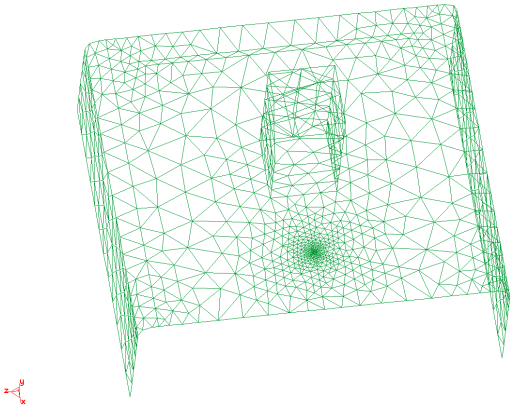


Figure 6. Mesh with the elements concentrated around a point, with a size gradient of 0.8.

11 . The size gradient (0.4) results in a higher density around the point (see Figure 6).

12 . Now go back and enter 0.6 in **Unstructured Size Transitions**. This will result in a mesh more suitable for our objectives. Click **Accept**.

---

[4] Instead of pressing the **ESC** key, the center mouse button or the mouse wheel can also be used.

5.2.3 Assignment around lines

1 . Select **Mesh->Unstructured->Assign size on lines**. In the window that appears, enter the size of the elements around the lines that will be chosen. Enter 0.5 and click **OK**.
2 . Select the lines defining the base of the prism (i.e. lines 1, 2, 3, 4 and 40). To see entity numbers select **Label** from the mouse menu or from the **View** menu. If you wish geometrical entity labels to be displayed, the view mode has to be set to Geometry using **View->Mode->Geometry** (this option may also be found in the GiD Toolbox), and the render mode must be set to **Normal**. Press **ESC**.
3 . Select **Mesh->Generate Mesh**.
4 . A window opens asking if the previous mesh should be eliminated. Click **Yes**.
5 . Another window appears in which you may enter a maximum element size. Leave the default value unaltered and click **OK**. This results in a high concentration of elements around the chosen lines, given that the selected element size (0.5) is much smaller than that of the rest of the elements in the model (see Figure 7).

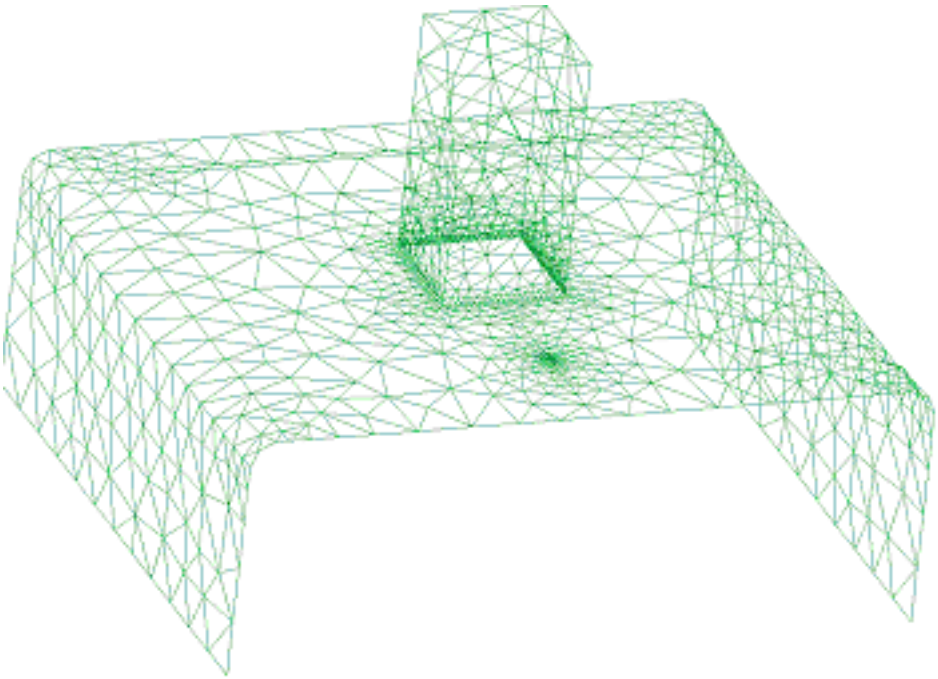


Figure 7. Mesh with a concentration of elements around lines.

5.2.4 Assignment on surfaces

1 . Select **Mesh->Unstructured->Assign size on surfaces**. In the window that appears, enter the size of the elements to be assigned on the surfaces that will be chosen. Enter 0.5 and click **OK**.
2 . Select the triangular surface resulting from the section of one of the vertexes of the prism (surface number 1). Press **ESC**.
3 . Select **Mesh->Generate Mesh**.
4 . A window opens asking if the previous mesh should be eliminated. Click **Yes**.
5 . Another window appears in which you can enter the maximum element size. Leave the default value unaltered and click **OK**. This results in a high concentration of elements on the chosen surface; due to the value selected (0.5) (see Figure 8).

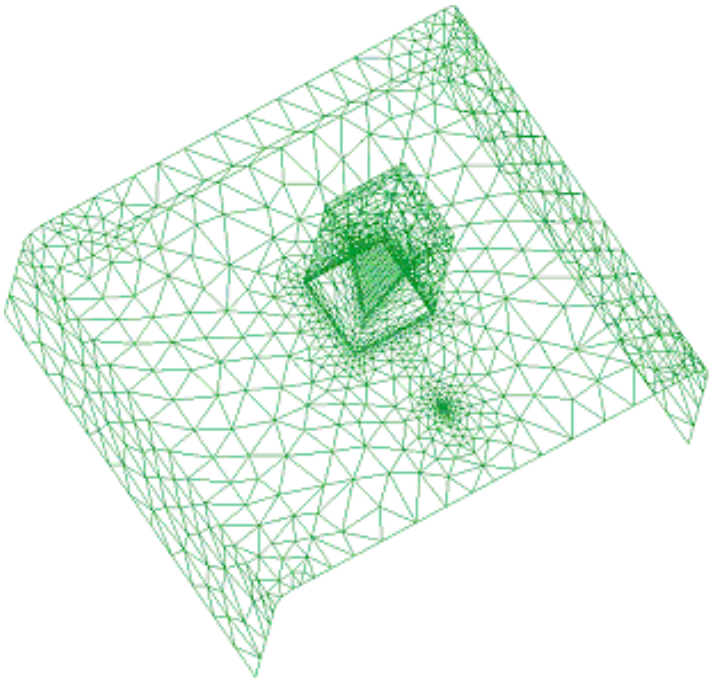


Figure 8. Mesh with a concentration of elements on a surface.

5.2.5 Assignment with Maximum Chordal Error

1 . Select **Mesh->Unstructured->Sizes by chordal error**….
2 . GiD asks for the minimum meshing size. Enter 0.1.
3 . GiD asks for the maximum meshing size. Enter 10.
4 . Enter the chordal error. This error is the maximum distance between the element generated and the real object (geometry). Enter 0.05 and press **OK**.
5 . Select **Mesh->Generate Mesh**.
6 . A window opens asking if the previous mesh should be eliminated. Click **Yes**.
7 . Another window appears in which you can enter a maximum element size. Leave the default value unaltered and click **OK**. This results in a high concentration of elements in curved areas. Now our approximation is significantly improved (see Figure 9).

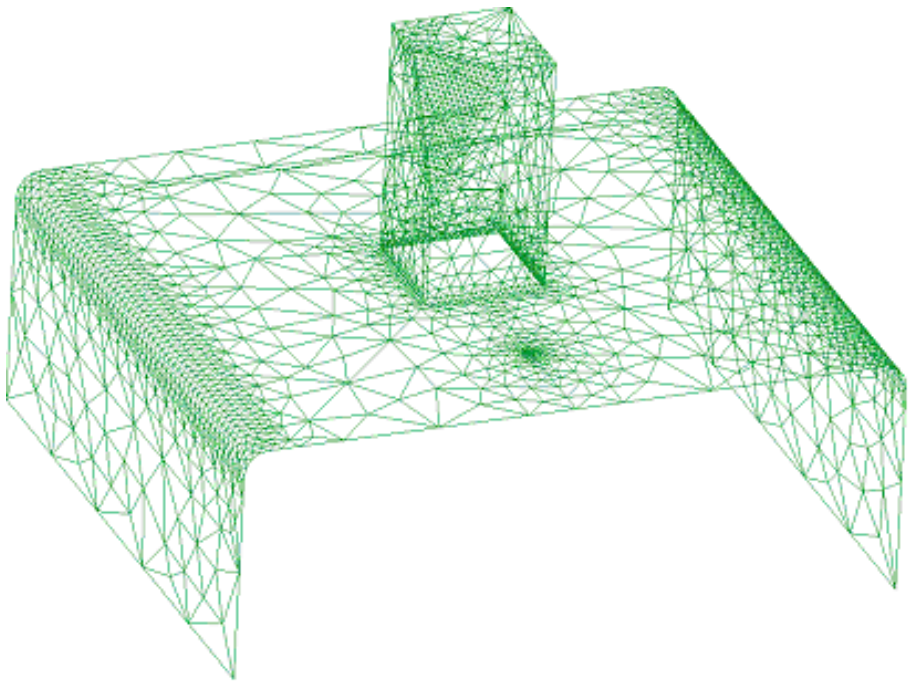


Figure 9. Mesh using sizes assignment by chordal error. Here, there is a greater concentration of elements in the curved zones.

5.3 Rjump mesher

The RJump mesher is a surface mesher that meshes patches of surfaces (in 3D space) and is able to skip the inner lines of these patches when meshing. By default, the RJump mesher skips the contact lines between surfaces that are tangent enough, and points between lines that are tangent enough. By selecting **Mesh->Draw->Skip entities (Rjump)**, the entities that RJump is going to skip and the ones that it is not going to skip are displayed in different colors. In this chapter we will see the properties of this mesher.

5.3.1 RJump default options

1 . Select **Mesh->Reset mesh data** to reset all mesh sizes introduced previously.
2 . A window appears advising that all the mesh information is going to be erased. Press **OK**.
3 . Go to **Utilities** and open **Preferences.** Click **Meshing**. In the window that appears you can choose between the three surface meshers available in GiD (RFast, RSurf and RJump). Select **RJump** mesher. Click **Accept**.
4 . Select **Mesh->Generate Mesh**.
5 . A window opens asking if the previous mesh should be eliminated. Click **Yes**.
6 . Another window appears in which you can enter a maximum element size. Leave the default value unaltered and click **OK**. This results in a mesh where contact lines between surfaces that are tangent enough do not have nodes; contact points between lines tangent enough are also skipped when meshing (see Figure 10).

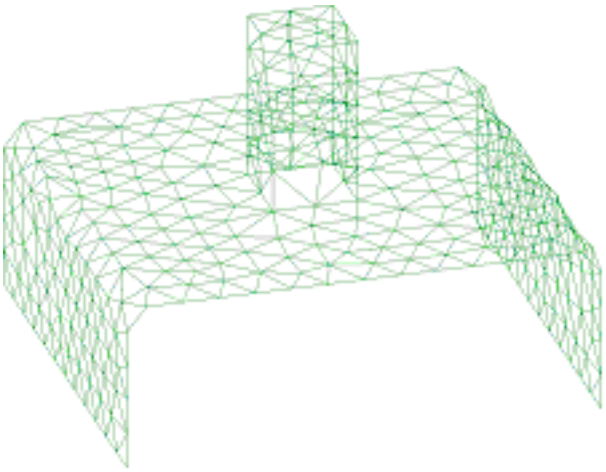


Figure 10. Mesh using the RJump mesher.

Note that the smaller elements shown in Figure 3 do not appear in this mesh, because of the properties this mesher.Using the RJump mesher it is possible to assign sizes to different entities. As an example, select **Mesh->Unstructured->Size by chordal error**….

7 . GiD asks for the minimum element size. Enter 0.1.
8 . GiD asks for the maximum element size. Enter 10.
9 . Enter the chordal error. This error is the maximum distance between the element generated and the real object. Enter 0.05 and press **OK**.
10 . Again, select **Mesh->Generate Mesh**.
11 . A window opens asking if the previous mesh should be eliminated. Click **Ok**.
12 . Another window appears in which you can enter a maximum element size. Leave the default value unaltered and click **OK**. This results in a high concentration of elements in curved areas, without the nodes in the lines and points that mesher skips. Now our approximation is significantly improved (see Figure 11).

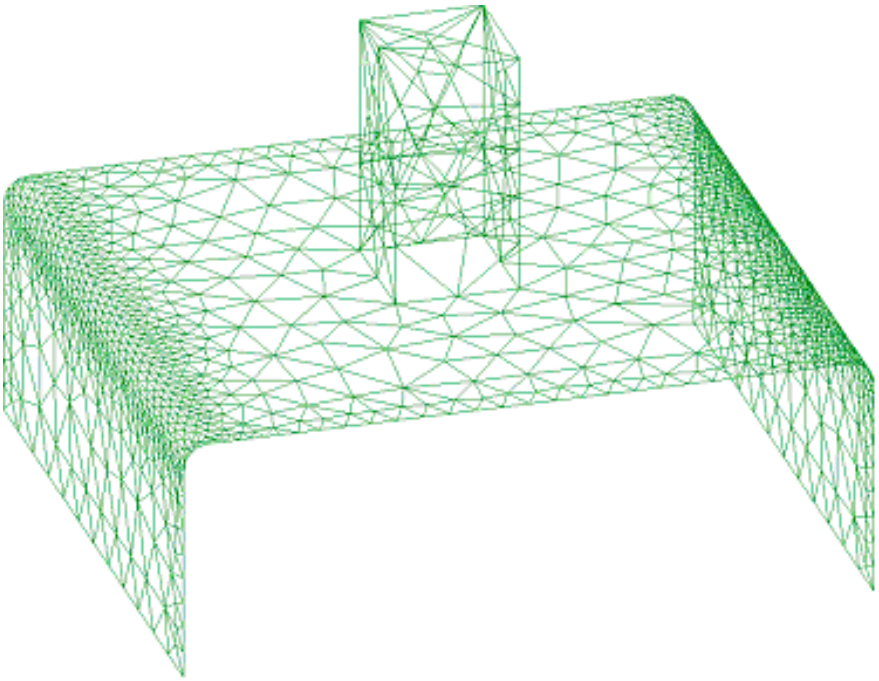


Figure 11. Mesh using the RJump mesher and assigning sizes by chordal error.

5.3.2 Force to mesh some entity

If there is a line or a point that the RJump mesher would usually skip, but that you wish to be meshed, you can specify the entity so that it is not skipped. As an example, we will force Rjump to mesh line number 43, in order to concentrate elements around point number 29, as it was done in chapter 2.2.

1 . Select **Mesh->Mesh criteria->No skip->lines**, and select line number 43. Press **ESC**.
2 . Select **Mesh->Draw->Skip entities(Rjump)** to display the entities that will and will not be skipped in different colors. As is shown in Figure 12, line 43 will now not be skipped; the rest of the lines are unaffected, and RJump will either skip or mesh them according to its criteria.

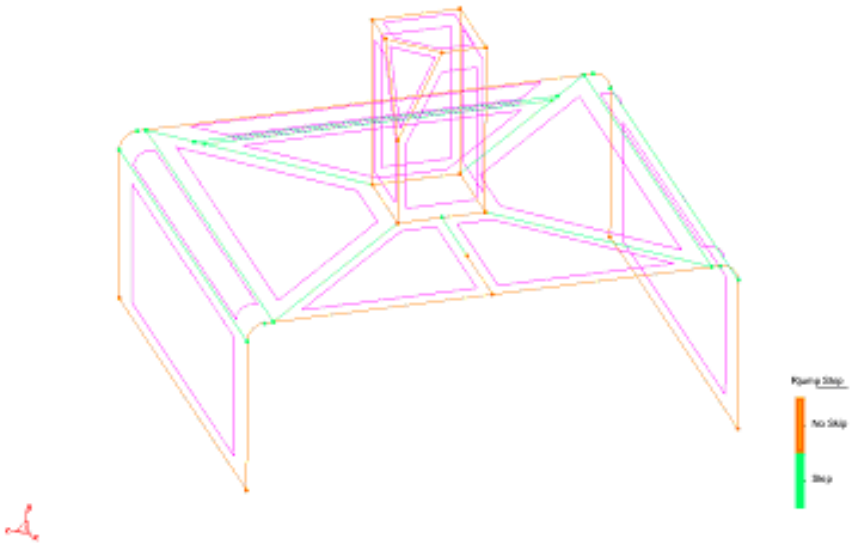


Figure 12. Entities that will be skipped and not skipped using the RJump mesher.

3 . Select **Mesh->Unstructured->Assign size on points**. A window appears in which to enter the element size around the points to be chosen. Enter 0.1 and click **Assign**.
4 . Select the point indicated in Figure 4 (point number 29). Press **ESC** to indicate that the selection of points is finished.
5 . Select **Mesh->Generate Mesh**.
6 . A window opens asking if the previous mesh should be eliminated. Click **Ok**.
7 . Another window appears in which you can enter a maximum element size. Leave the default value unaltered and click **OK**. This results in a mesh like the one obtained before (in Figure 10), but with high concentration of elements around point number 29. Note that there are nodes on line number 43 because we have forced RJump not to skip this line (see Figure 13).

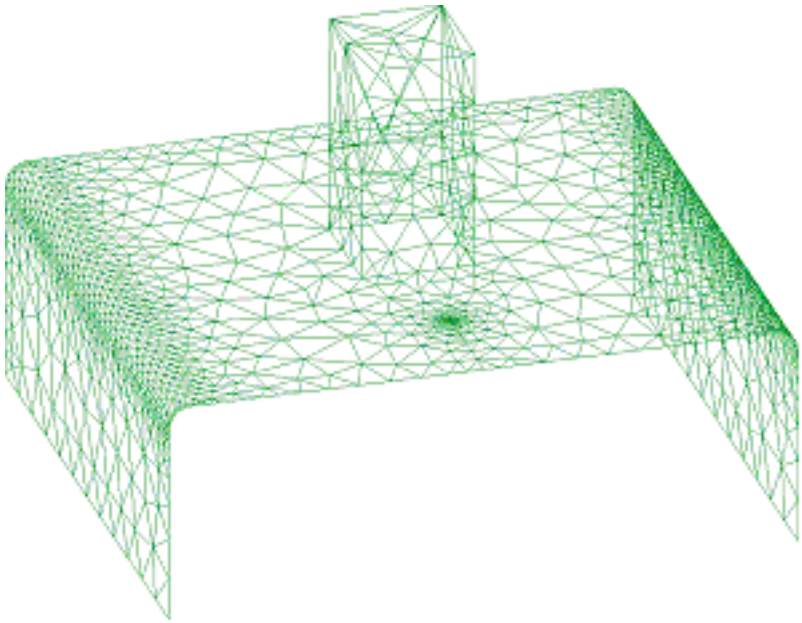


Figure 13. Mesh using the RJump mesher, assigning sizes by chordal error and forcing an entity to be meshed.

In this last example we have forced the mesher not to skip an entity, but it may be interesting in some models to allow the mesher only to skip a few entities, meshing almost all or them. In this case, a different surface mesher can be selected (in the **Preferences** window). One option is the RSurf mesher which meshes everything except the entities that you ask it to skip, using the **Mesh->Mesh criteria->Skip** command. Here, because RJump is not selected, no entity will be skipped automatically according to tangency with neighboring entities. The next example shows how to work with this mesher.

8 . Select **Mesh->Reset mesh data** to reset all mesh sizes introduced previously.

9 . A window opens advising that all the mesh information is going to be erased. Press **OK**.

10 . Go to **Utilities** and open **Preferences.** Click **Meshing**. In the window that appears you can choose between the three surface meshers available in GiD (RFast, RSurf and RJump). Select the **RSurf** mesher. Click **Accept**.

11 . Select **Mesh->Mesh criteria->Skip->lines**, and select lines 48 and 53. Press **ESC**.

12 . Select **Mesh->Generate Mesh**.

13 . A window opens asking if the previous mesh should be eliminated. Click **Yes**.

14 . Another window appears in which you can enter a maximum element size. Leave the default value unaltered and click **OK**. The result is a mesh similar to the first example obtained in chapter 2 (see Figure 2), but the smaller elements highlighted in Figure 3 do not appear because lines 48 and 53 (which were meshed before) are now skipped when meshing (see Figure 14).

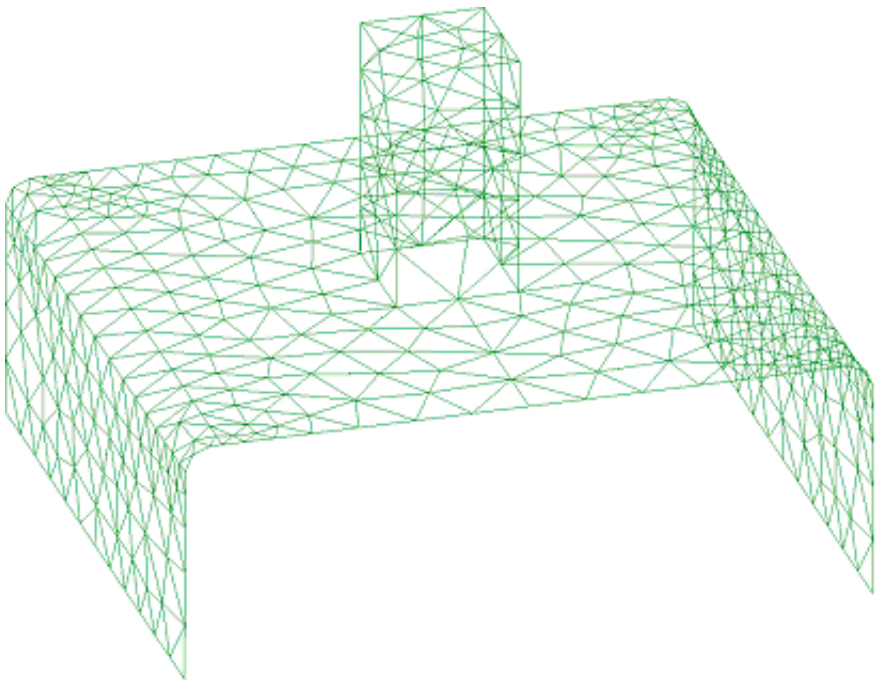


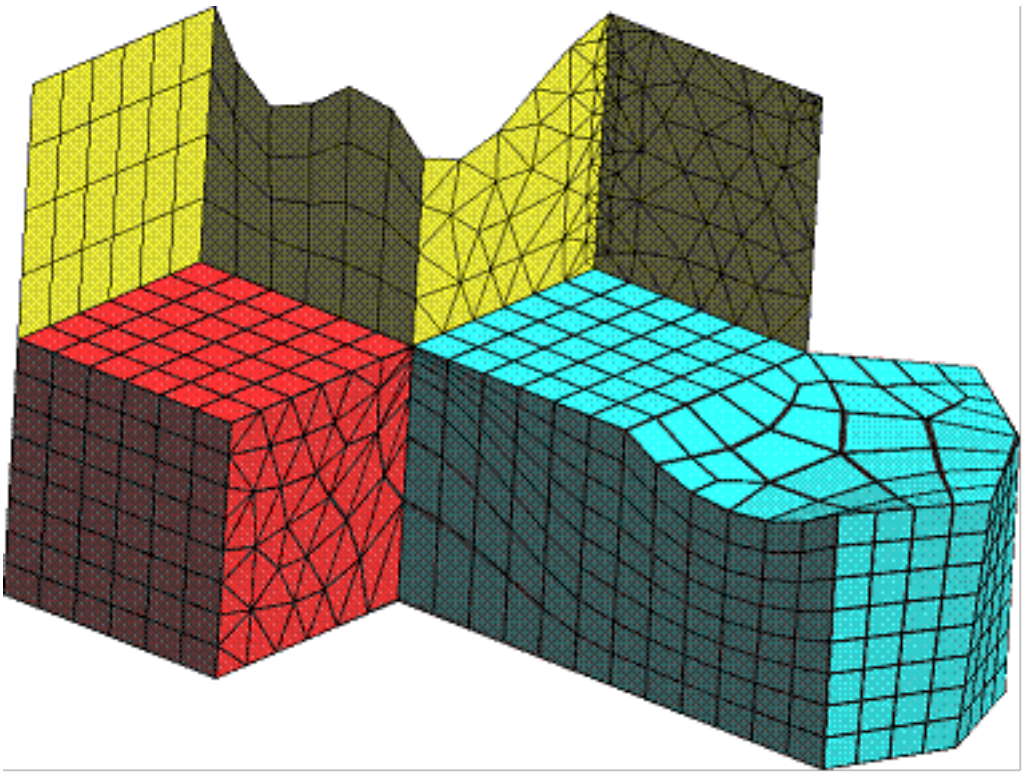
Figure 14. Mesh using the RSurf mesher, with some lines skipped.

6 METHODS FOR GENERATION THE MESH

The objective of this example is to mesh a model using the various options available in GiD for controlling the element type in structured, semi-structured and unstructured meshes. It also presents how to concentrate elements and control the distribution of mesh sizes.

The six methods covered are:

- Generating a mesh using tetrahedral
- Generating a volume mesh using spheres
- Generate a mesh using circles
- Generating a volume mesh using points
- Generating a mesh using quadrilaterals
- Generating a structured mesh on surfaces and volumes
- Generating a semi-structured volume mesh
- Generating a mesh using quadratic elements



6.1 Introduction

In order to carry out this example, start from the project "ToMesh3.gid". This project contains a geometry that will be meshed using different types of elements.

6.1.1 Reading the initial project

1 . In the **Files** menu, select **Read**. Select the project "ToMesh3.gid" and click **Open**.

2 . The geometry appears on the screen. It is a set of surfaces and three volumes. Select **Render->Flat** from the mouse menu[1] or from the **View** menu. In Figure 1 shows the geometrical model loaded.

3 . Select **Rotate->Trackball** from the mouse menu. (This option is also available in the GiD Toolbox.) Make several changes in the perspective so as to get a good idea of the geometry involved.

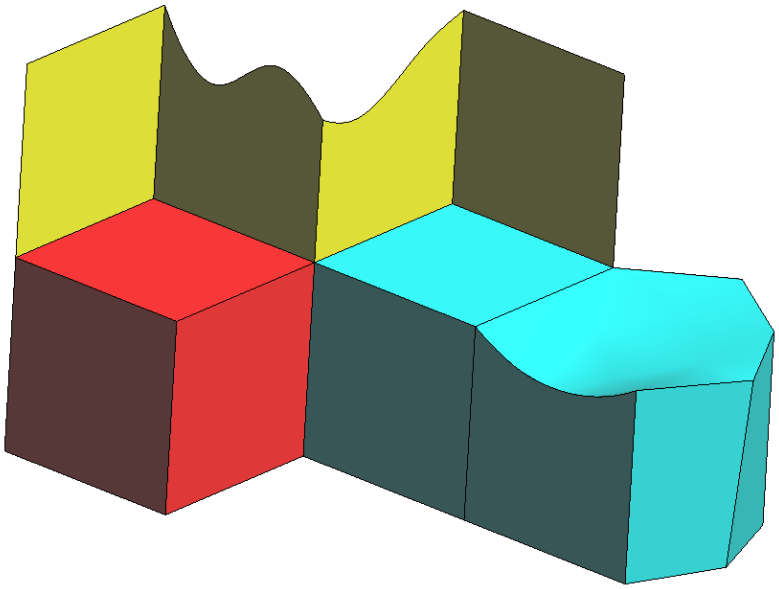


Figure 1. Contents of the project “ToMesh3.gid”

Finally, return to the normal visualization **Render->Normal**. This mode is more user-friendly.

---

[1] The mouse menu appears when the right mouse button is clicked.

6.2 Types Of Meshes

Using GiD the mesh may be generated in different ways, depending on the needs of each project. The two basic types of meshes are the structured[2] mesh and the unstructured mesh. For volumes only there is one additional type, the semi-structured[3] mesh.

For all these types of mesh a variety of elements may be used (linear ones, triangles, quadrilaterals, circles,tetrahedra, hexahedra, prisms, spheres or points). In this tutorial you will become familiarized with the mesh-generating combinations available in GiD.

---

[2] A structured mesh is one in which each node is connected to a constant number of elements.

[3] A semi-structured volume mesh is one in which you can distinguish a fixed structure in one direction, i.e. there is a fixed number of divisions. However, within each division the mesh need not be structured. This kind of mesh is only practical for topologically prismatic volumes.

6.2.1 Generating the mesh by default

1 . Select **Mesh->Generate mesh**.
2 . A window comes up in which to enter the maximum element size for the mesh to be generated. Leave the default value unaltered and click **OK**.
3 . A meshing process window comes up. Then another window appears with information about the mesh generated. Click **OK** to visualize the mesh.
4 . The result is the mesh in Figure 2. There are various surfaces and volumes. By default, mesh generation in GiD obtains unstructured meshes of triangles on surfaces and tetrahedra on volumes.
5 . Select **Render->Flat** to see the mesh in render mode. As is shown in Figure 3, volume meshes are

represented a little bit differentlyfrom surface meshes, although in both cases triangles are shown. If the triangles you see are the boundary of a volume mesh, they are shown with black edges that are thicker than surface meshes triangles. If the triangles form a boundary volume mesh and, at the same time, a triangle surface mesh (this can be obtained if surfaces are selected with the option **Mesh->Mesh criteria->MeshSurfaces**), the wider edges are colored with the color of the surface layer. Examples of these different kinds of render are shown in Figure 3.

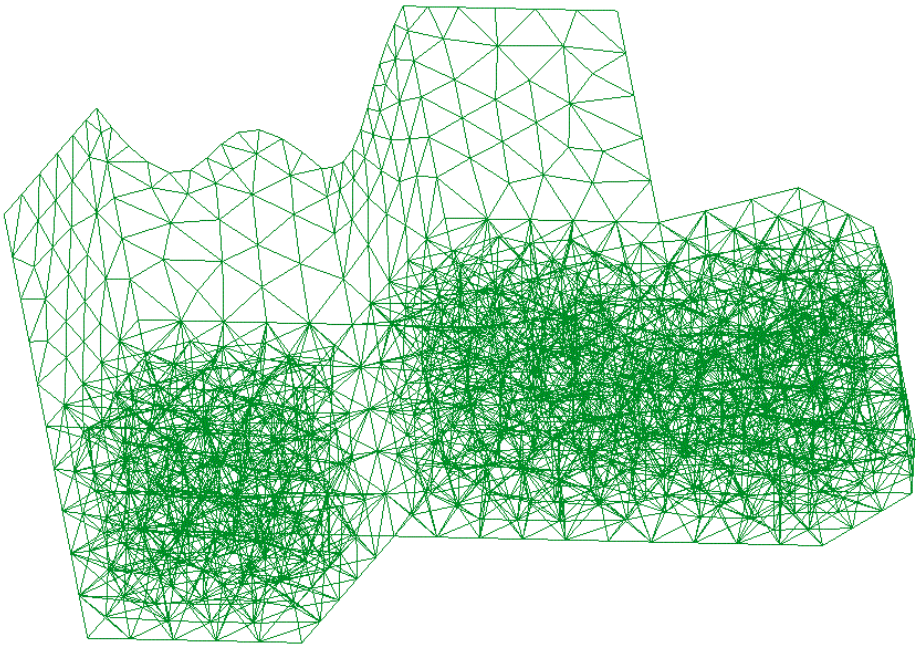


Figure 2. Generating the mesh by default.

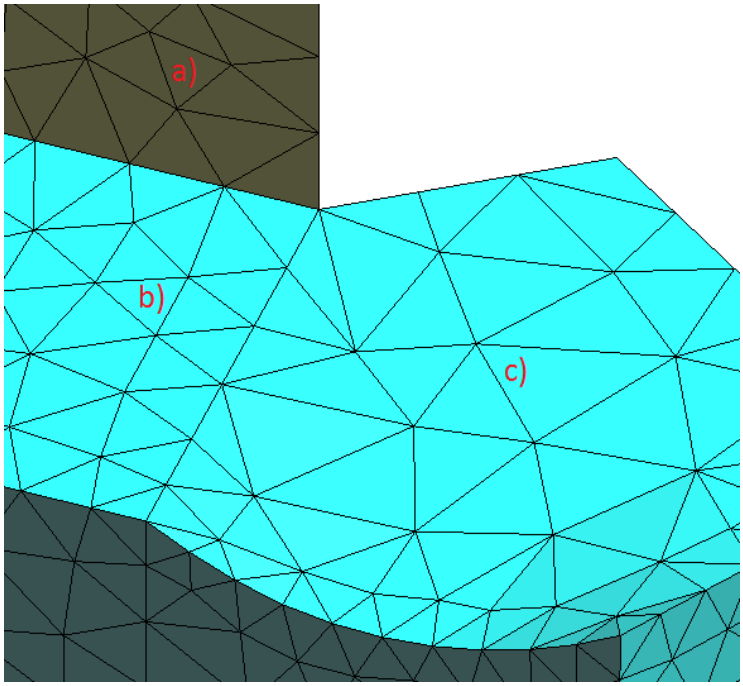


Figure 3. Different render styles: a) surface mesh, b) volume mesh, c) surface mesh and volume mesh together (surface layer is red and volume layer is blue)

6.2.2 Generating the mesh using circles and spheres

1 . Select **Mesh->Element type->Sphere**. Select volume number one and press **ESC.** To see entity numbers select **Label** from the mouse menu or from the **View** menu. If you wish the geometrical entity labels to be displayed, the view mode needs to be changed to Geometry using **View->Mode->Geometry** (this option may also be found in the GiD Toolbox). Select **RenderNormal** to see the labels.
2 . Select **Mesh->Element type->Circle**. Select surface number 24 and press **ESC**.
3 . Select **Mesh->Generate mesh**.
4 . A window comes up asking whether the previous mesh should be eliminated. Click **Ok**.
5 . Another window appears in which to enter the maximum element size. Leave the default value unaltered and click **OK**. The result is a mesh as illustrated in Figure 4.

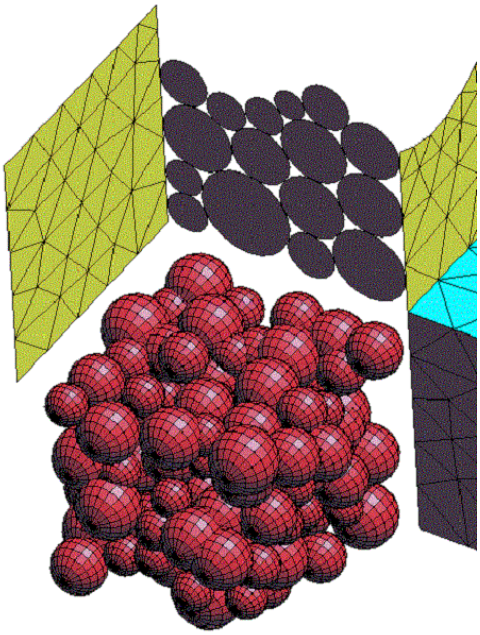


Figure 4.(Render Flat) Generating a mesh on a volume using points.

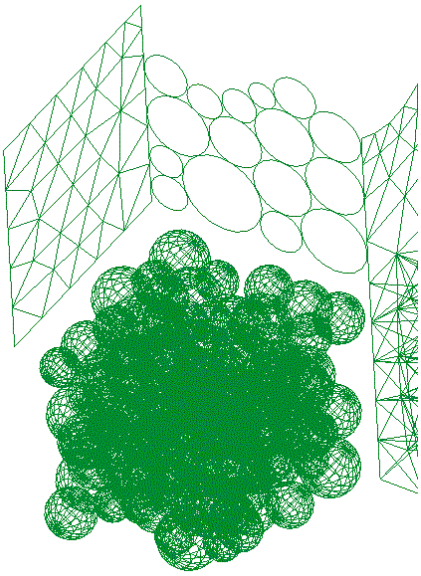


Figure 4.(Render Normal) Generating a mesh on a volume using points.

Instead of pressing the **ESC** key, the center mouse button or the mouse wheel can also be used.

6.2.3 Generating the mesh using points

1 . Select **Mesh->Element type->Only points**. Select volume number one and press **ESC.**
2 . Select **Mesh->Generate mesh**.
3 . A window comes up asking whether the previous mesh should be eliminated. Click **Ok**.
4 . Another window appears in which to enter the maximum element size. Leave the default value unaltered and click **OK**. The result is a mesh as illustrated in Figure 5.

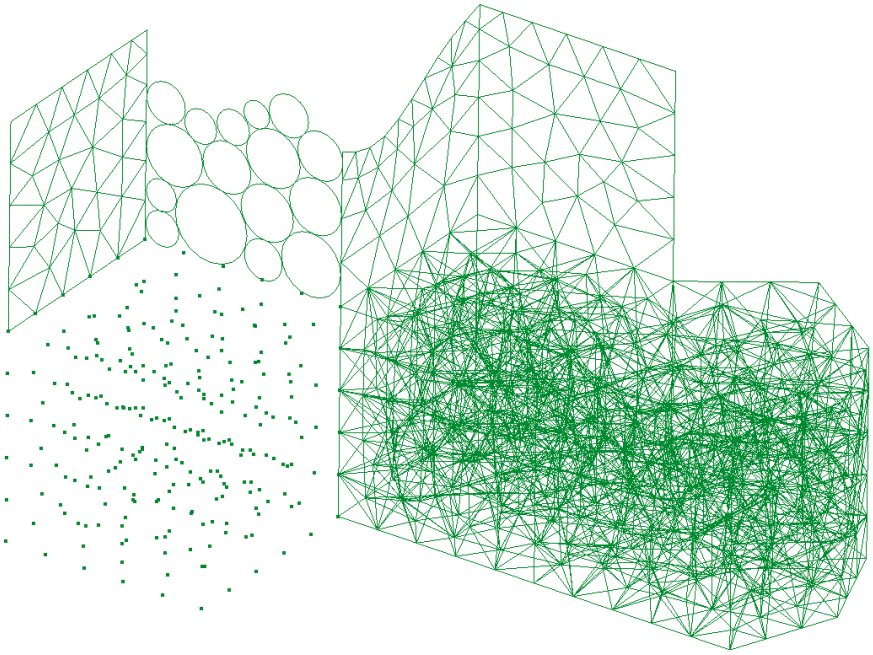


Figure 5. Generating a mesh on a volume using points.

5 . Now volume number one is meshed using only the generated nodes.

6.2.4 Generating the mesh using quadrilaterals

1 . Select **Mesh->Element type->Quadrilateral** . Select surfaces number 24 and 12.

2 . Select **Mesh->Generate mesh**.

3 . A window comes up asking whether the previous mesh should be eliminated. Click **Ok**.

4 . Another window appears in which the maximum element size can be entered. Leave the default value unaltered and click **OK**. The result will be the mesh illustrated in Figure 6.

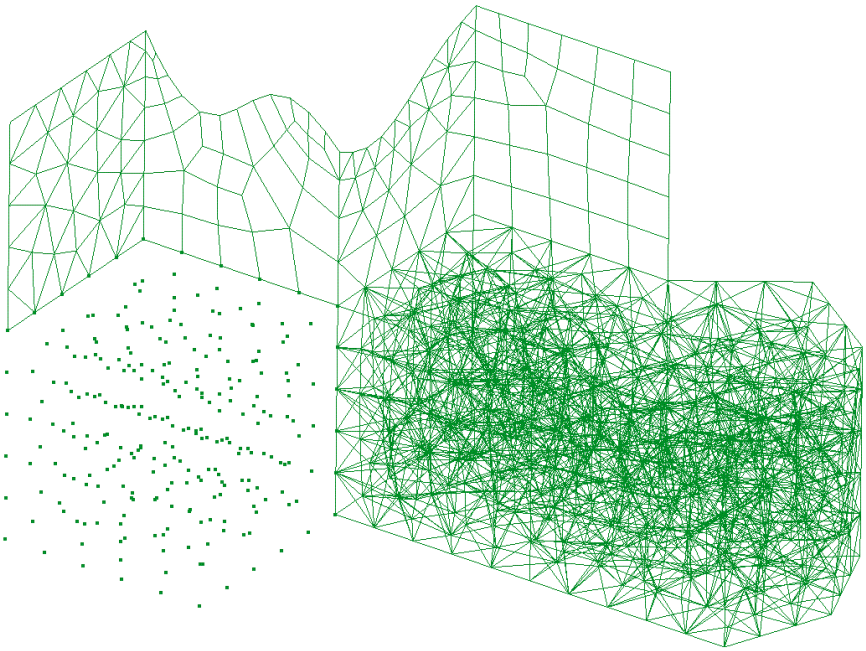


Figure 6. An unstructured mesh generated using quadrilaterals.

5 . The surface is meshed with quadrilaterals, forming an unstructured mesh, surface 12 seems structured but if you zoom in, you will see that it is not.

6.2.5 Generating a structured mesh (surfaces)

1 . To mesh surfaces with a structured mesh, select the option **Mesh->Structured->Surfaces->Assignnumber of cells**.

2 . Select all top surfaces 9, 24, 26 and 12 and press **ESC**.

3 . A window appears in which to enter the number of divisions that the lines to be selected will have. Enter 4.

4 . Press **Assign** and select one vertical line[5] (parallel to the **Y** -axe). Press **ESC**.

5 . Another window appears in which to enter the number of divisions on the lines. Enter 6.

6 . Press **Assign** and select the 4 bottom lines[5]. Press **ESC**.

7 . Another window appears in which to enter the number of divisions on the lines. In this case, all the boundary lines have already been defined. Therefore, click **Close**.

8 . Select **Mesh->Element type->Triangle**. Select surfaces 26 and 12.

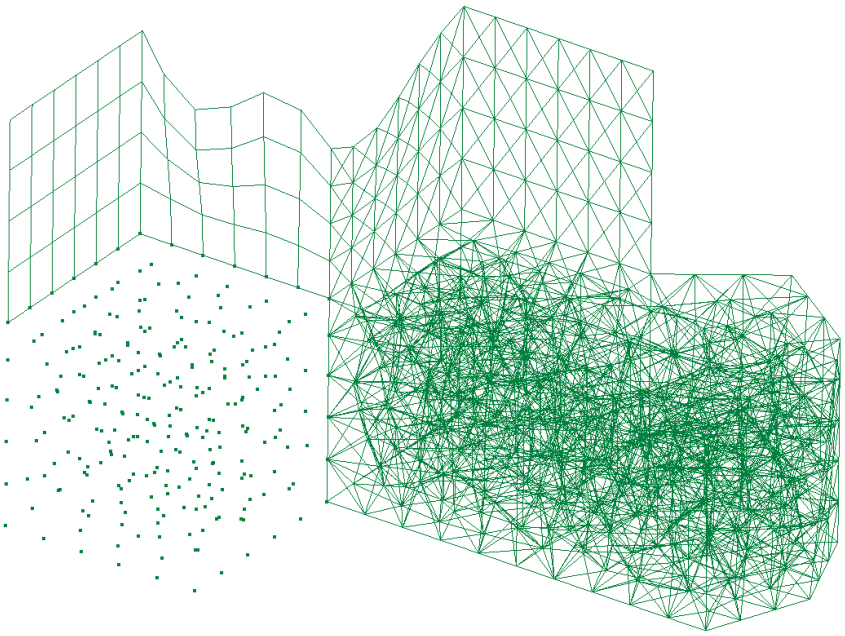


Figure 7. Structured mesh of quadrilateral and triangular elements on surfaces.

o

9 . Select MeshGenerate mesh.

10 . A window comes up asking whether the previous mesh sould be eliminated. Click Ok.

11 . Another window appears in wihich to enter the maximum element size.Leave the default value unaltered an click Ok. The result is the mesh shown in Figure 8.

12 . As seen in Figures 7 and 8, GID can obtain surface structured meshes made of quadrilaterals or triangles. There are tow kinds of structured mesh that use triangles:the one shown in Figure 7 is obtained when the **Utilities->Preferences->Meshing->Symmetrical structured->triangles** option i set. If this option is not set, the mesh presented in Figure 8 is produced (with fewer nodes than if using the previous option).

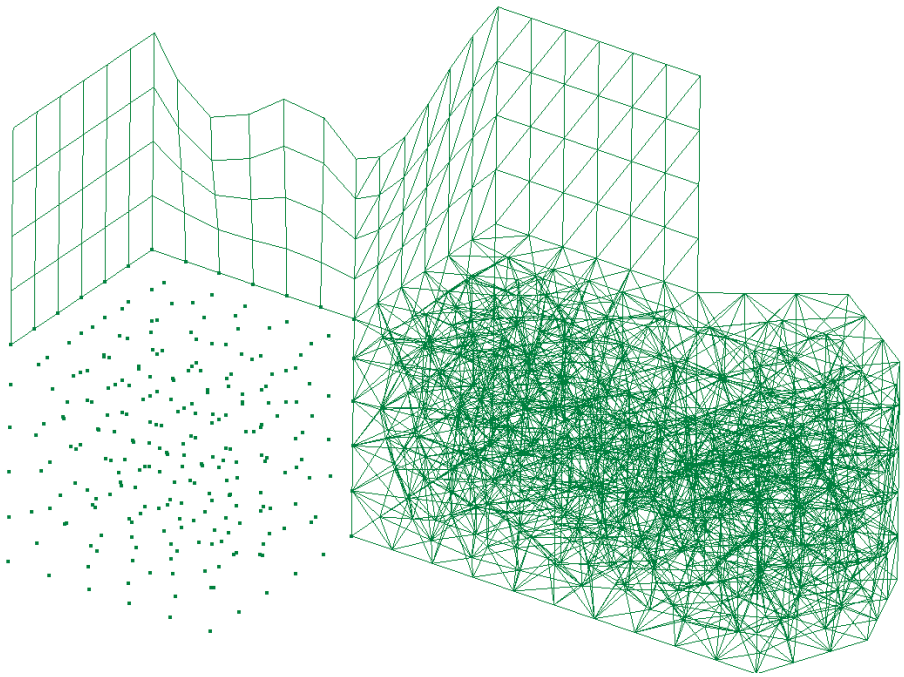


Figure 8. Structured mesh of quadrilateral and triangular elements on surfaces, with the option Symmetrical structured triangles not set.

---

[5] When selecting a line, GID automatically selects all lines parallel to it.

6.2.6 Generating structured meshes (volumes)

1 . To mesh volumes with a structured mesh, select the option **Mesh->Structured->Volumes**.

2 . Select volumes 1 and 2 and press **ESC**[6].

3 . A window appears in which to enter the number of divisions that the lines to be selected will have. Enter 6.

4 . Select lines of both volumes parallel to the **X** - and **Z** -axes. GiD automatically selects all the lines in each volume parallel to these in order to create the structured mesh. Press **ESC**.

5 . Another window appears in which to enter the number of divisions on the lines. Divide the lines parallel to the **Y** -axis into 8 segments. Enter 8 and click **OK**.

6 . Select an edge of volume 1 or 2 parallel to the **Y** -axis and press **ESC** . Again, the line-division window comes up. Since we have already finished the assignments, click **Close**.

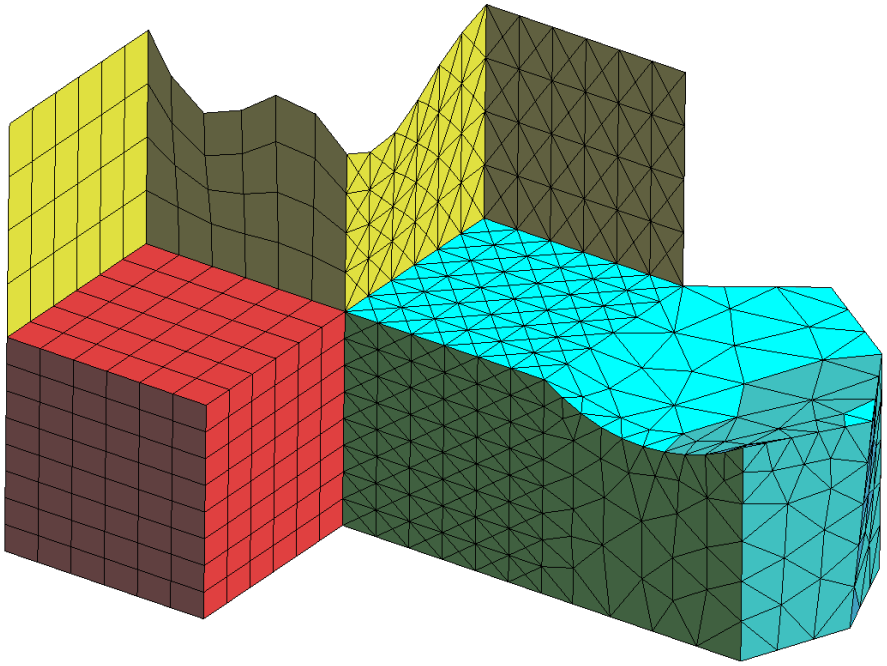


Figure 9. Structured volume mesh of hexahedra an tetrahedra

7 . For structured volumes, GiD generates hexahedron meshes by default, but tetrahedron structured meshes can also be assigned. Select **Mesh->Element type->Tetrahedra** and then select volume number 2.Select **Mesh->Generate mesh**.

8 . A window appears asking whether the previous mesh should be eliminated. Click **Yes**.

9 . Another window comes up in which to enter the maximum element size. Leave the default value unaltered and click **OK** . The result is the mesh shown in Figure 10.

10 . GiD can obtain volume structured meshes made of hexahedra, tetrahedra or prisms. As can be seen in Figures 9 and 10, there are two kinds of tetrahedron structured mesh: the one shown in Figure 9 is obtained when the option **Utilities->Preferences->Meshing->Symmetrical structured tetrahedra** is set. If this option is not set, the mesh presented in Figure 10 is produced (with fewer nodes than if using the previous option;also, it is not topologically symmetrical).

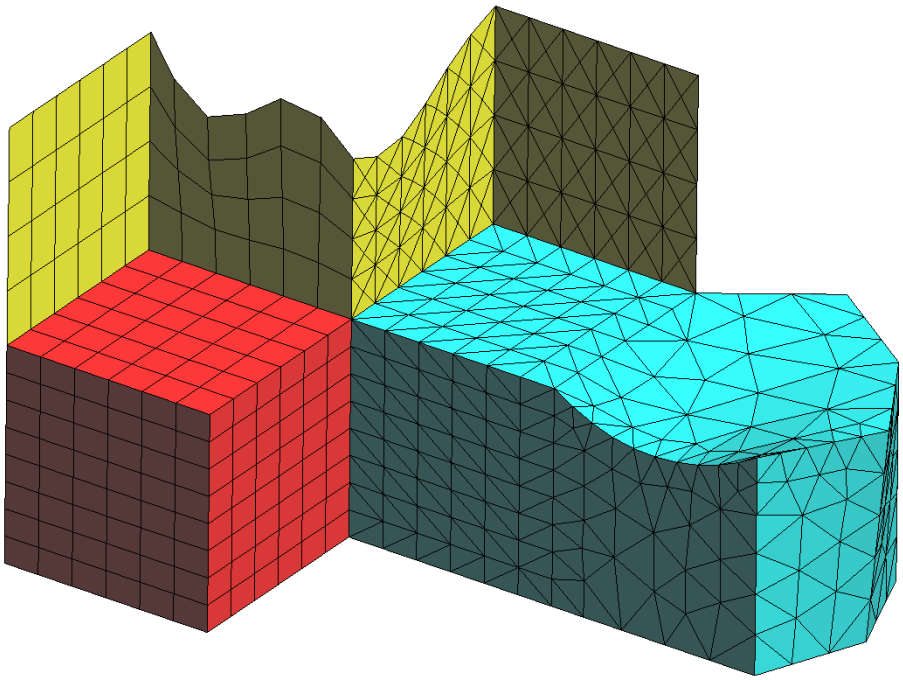


Figure 10. Structured volume mesh of tetrahedra with the option Symmetrical structured tetrahedra not set.

6.2.7 Generating semi-structured meshes (volumes)

1 To mesh volumes with a semi-structured mesh, select the option **Mesh->SemiStructured->Volumes**.
2 A window appears in which to enter the number of divisions for the direction in which it is structured (the prismatic one). Enter 8.
3 Select volume 3 and press **ESC**. As volume 3 is prismatic in one direction only (i.e. parallel to **Y** -axis) GiD will automatically detect this fact and will select it to be the direction in which the semi-structured volume mesh is structured.
4 Another window appears in which to enter the number of divisions in the direction of the structure. In this case we do not want to select any more volumes, so click **Close**.
5 Select **Mesh->Generate mesh**.
6 A window appears asking whether the previous mesh should be eliminated. Click **Yes**.
7 Another window appears in which to enter the maximum element size. Leave the default value unaltered and click **OK**. The result is the mesh shown in Figure 11.

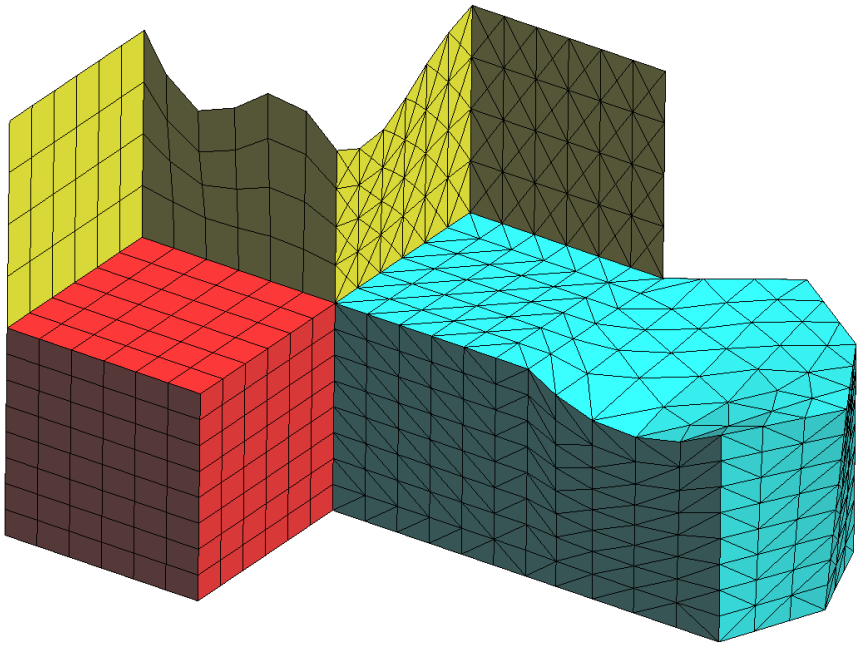


Figure 11. Semi-structured volume mesh of tetrahedra

As can be seen, volume 3 has been meshed with tetrahedra. Semi-structured volumes are meshed with prisms, by default. However, in this case it was not possible because of volume 2, which has tetrahedra assigned and shares one surface with volume 3. In the following steps a hexahedron mesh is produced.

8 Select **Mesh->Element type->Hexahedra.**

9 Select volumes 2 and 3 and press **ESC**.

10 Select **Mesh->Generate mesh**.

11 A window opens asking whether the previous mesh should be eliminated. Click **OK**.

12 Another window appears in which to enter the maximum element size. Leave the default value unaltered and click **OK**. The result is the mesh shown in Figure 12.

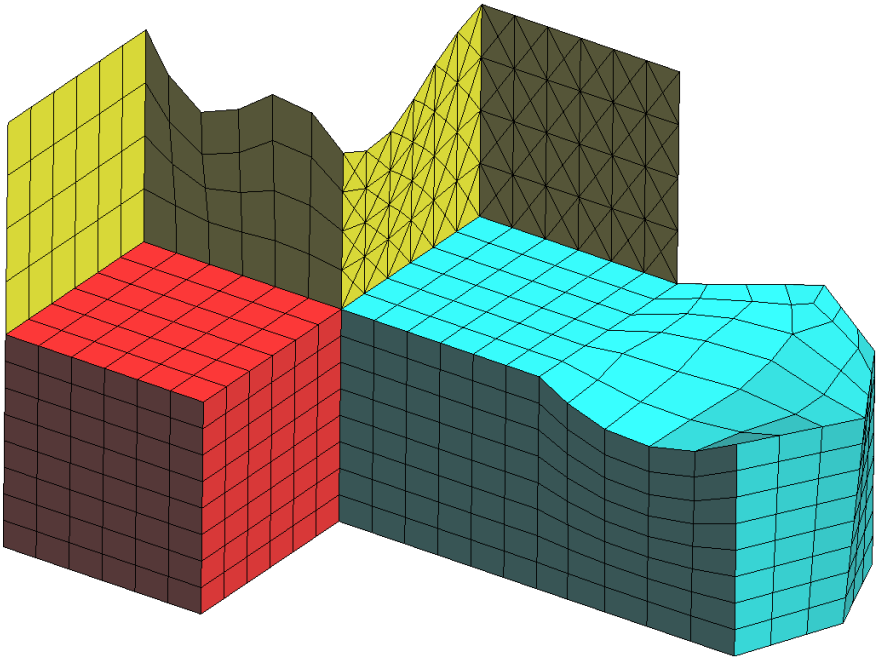


Figure 12. Semi-structured volume mesh of hexahedra

In case of volume number 3 there is only one direction in which it can possibly be structured (i.e. in the direction of the prism). If the volume is prismatic in more than one direction, there are two ways to choose between them: selecting one top surface (**Mesh->SemiStructured->Set->Master surface**) or the direction of the structure (**Mesh->SemiStructured->Set->Structured direction**). The following example explains this procedure.

13 Select the option **Mesh->SemiStructured->Volumes**.
14 A window opens in which to enter the number of divisions in the structured direction (prismatic). Enter 6.
15 Select volume 1 and press **ESC**.
16 Another window appears, click **Close**.
17 Select **Mesh->SemiStructured->Set->Structured direction**.
18 Select one line parallel to the**X** -axis of volume number 1 (for example line number 11) and press **ESC**.
19 Select **Mesh->Unstructured->Assign EntitiesSurfaces.**
20 Select surfaces 1 and 6 and press **ESC**.
21 Select **Mesh->Generate mesh**.
22 A window opens asking whether the previous mesh should be eliminated. Click **OK**.
23 Another window appears in which to enter the maximum element size. Leave the default value unaltered and click **OK**. The result is the mesh shown in Figure 13.

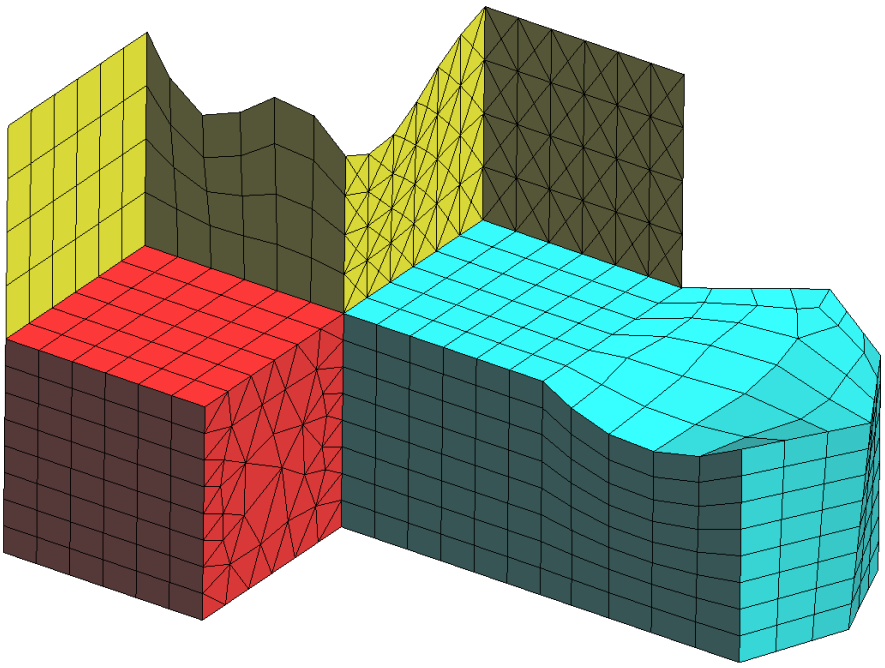


Figure 13. Semi-structured volume mesh of prisms and hexahedra.

NOTE : As can be seen, by selecting different element types for different geometrical entities, several kinds of meshes can be generated. Remember always to take care over the compatibility between element types in shared geometrical entities.

6.2.8 Concentrating elements and assigning sizes

1 Select **MeshStructuredLinesConcentrate elements**.
2 Select some structured lines, for example line 43. Press **ESC**.
3 A window comes up in which to enter two values for the concentration of elements. Positive values concentrate the elements and negative values spread them. Enter 1 as **Start Weight** and –0.5 as **End Weight** [10]. Press **OK**.
4 Select **Mesh->Generate mesh**.
5 A window opens asking whether the previous mesh should be eliminated. Click **OK**.
6 Another window appears in which to enter the maximum element size. Leave the default value unaltered. The result is the mesh shown in Figure 14.

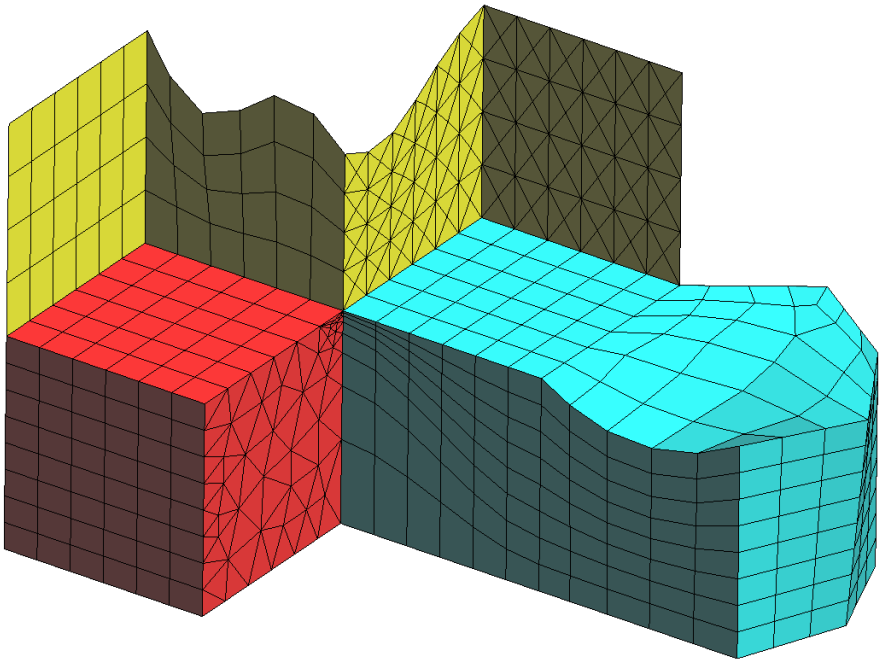


Figure 14. Concentration of elements on lines 3 and 23

It is also possible to assign sizes to geometrical entities, so that mesh elements can be concentrated in certain zones. In the following steps some examples are given.

7 Select **Mesh->Unstructured->Assign sizes on points.**
8 A window appears in which to enter the size to be assigned to points. Enter "0.1”.
9 Select point number 15 and press **ESC**.
10 Another window appears in which to enter the size to be assigned to points. In this case, we donot want to assign sizes to any other points,so click **Close**.
11 Select **Mesh->Unstructured->Assign sizes on lines.**
12 A window appears in which to enter the size to be assigned to lines. Enter "0.5”.
13 Select line number 25 and press **ESC**.
14 Another window appears in which to enter the size to be assigned to lines. In this case, we do not want to assign sizes to any more lines, so click **Cancel**.
15 Select **Mesh->Generate mesh**.
16 A window appears asking whether the previous mesh should be eliminated. Click **Ok**.
17 Another window appears in which the maximum element size should be entered. Leave the default value unaltered. The result is not the desired, we just get the previous mesh. This is because surrounding surfaces and lines are structured, so they do not have enought freedom to achive the

given sizes.

18 Select**Mesh->Unstructured->Assign entities->Surfaces.**

19 Select Surfaces 26 and 12. Press **ESC**.

20 Select**Mesh->Unstructured->Assign entities->Lines.**

21 Select lines 48, 26 and 27. Press **ESC**.

22 Select **Mesh->Generate mesh**.

23 A window appears asking whether the previous mesh should be eliminated. Click **OK**.

24 Another window appears in which the maximum element size should be entered. Leave the default value unaltered. The result is the mesh shown in Figure 15.

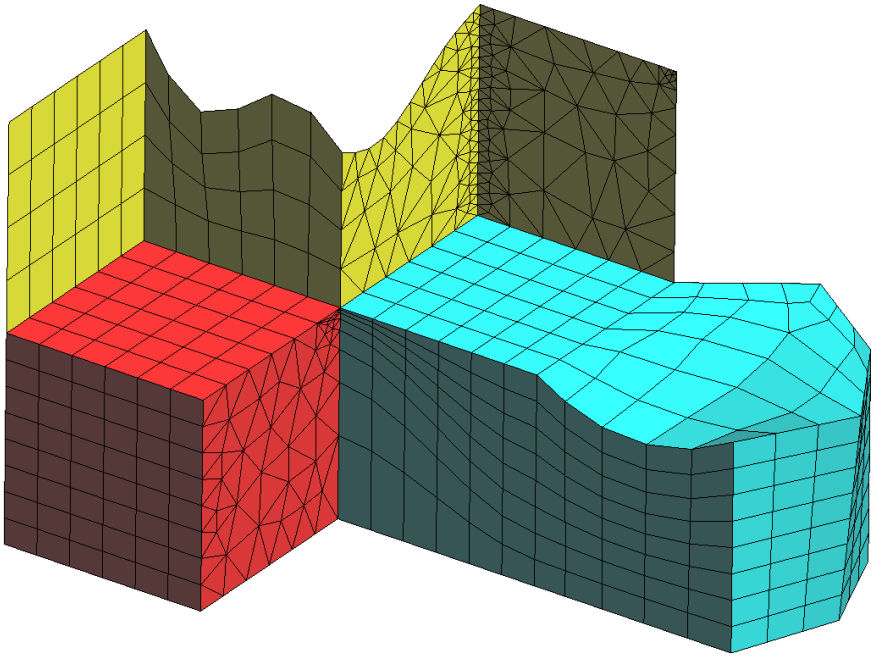


Figure 15. Unstructured size assigned in a point and a line.

---

[10] Start Weight and End Weight refer to the start point and end point of the line, oriented as it is drawn when you select it.

6.2.9 Generating the mesh using quadratic elements

Select **Zoom->In** from the mouse menu (this option may also be found in the GiD Toolbox or in the **View** menu). Enlarge one area of the mesh (e.g. the zone near point number 3).

1 Select **Label->All in->Points** . The result is shown in Figure 16.

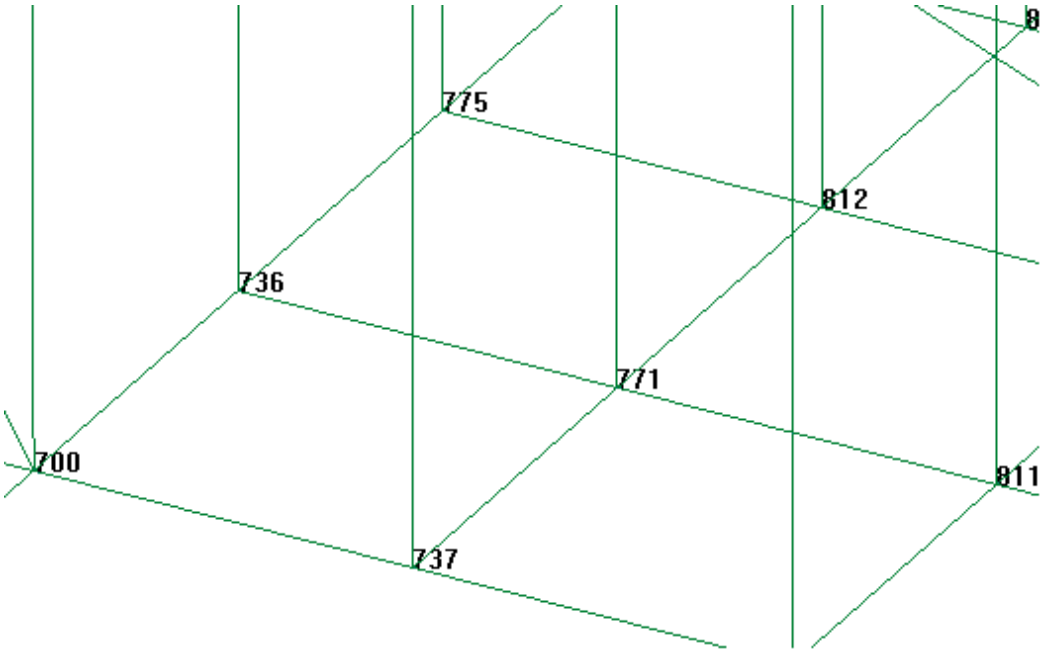


Figure 16. Each number identifies a node. There is a node for each element vertex.

2 The node identifiers created by generating the mesh appear on the screen. There is one identifier for each vertex of each element.

3 Select **Utilities->Preferences->Meshing->Quadratic type->Quadratic.**

NOTE : By default GiD meshes with first degree (linear) elements. To find out which mode GiD is working in, goes to **Utilities->Preferences->Meshing->Quadratic type**.

4 Select **Mesh->Generate mesh**.

5 A window opens asking whether the previous mesh should be eliminated. Click **OK**.

6 Another window appears in which the maximum element size should be entered. Leave the default value unaltered and click **OK**.

7 Once the mesh has been generated, select **Label->All in->Points** . The result is shown in Figure 17. Now, there are not only nodes at the vertices, but also at the midpoints of the edges of the elements.

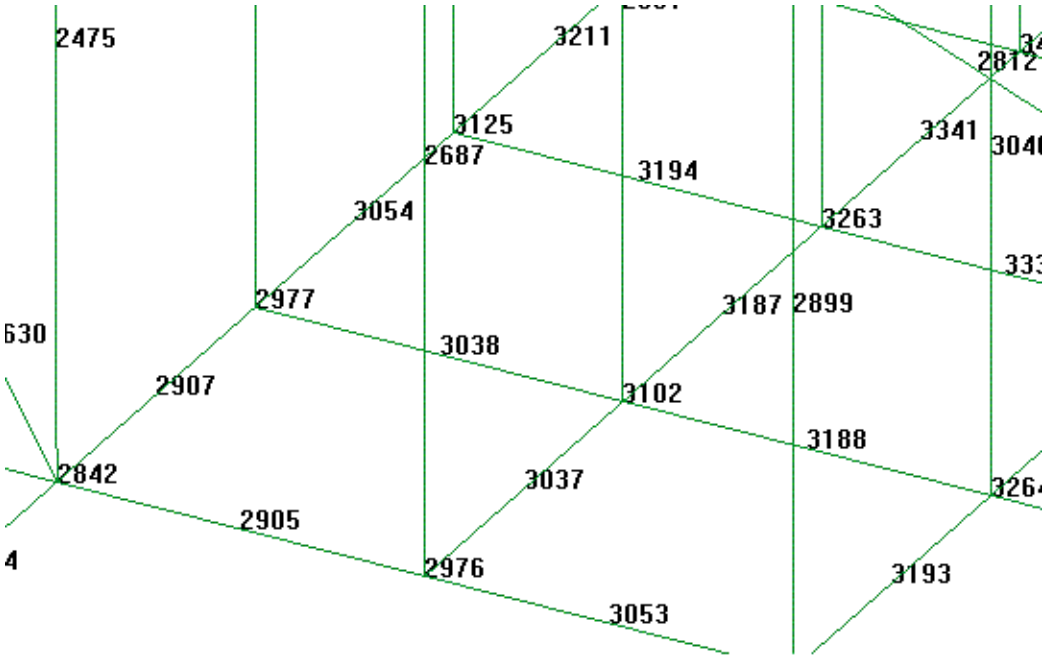


Figure 17. Each number identifies a node. There is a node at each element vertex and at the midpoint of each edge.

8 Select **Utilities->Preferences->Meshing->Quadratic type->Quadratic9**.

9 Select **Mesh->Generate mesh**.

10 A window opens asking whether the previous mesh should be eliminated. Click **OK**.

11 Another window appears in which the maximum element size should be entered. Leave the default value unaltered.

12 Select **Label->All in->Points** (see Figure 18).

13 Notice that the four-sided elements (quadrilaterals) also have a node in the center, in addition to the nodes at the vertices and midpoints of the edges. Similarly, hexahedra alsohave a node at their center point.

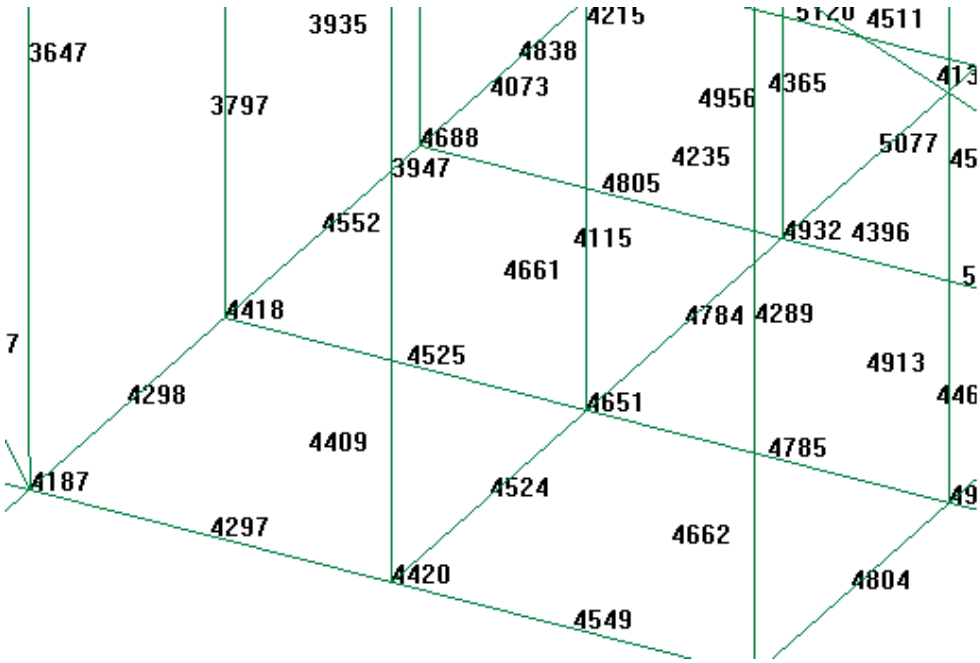


Figure 18. Each number identifies a node. There is a node at each vertex, at the midpoint of each edge and in the center of quadrilaterals and hexahedra.

7 POSTPROCESSING

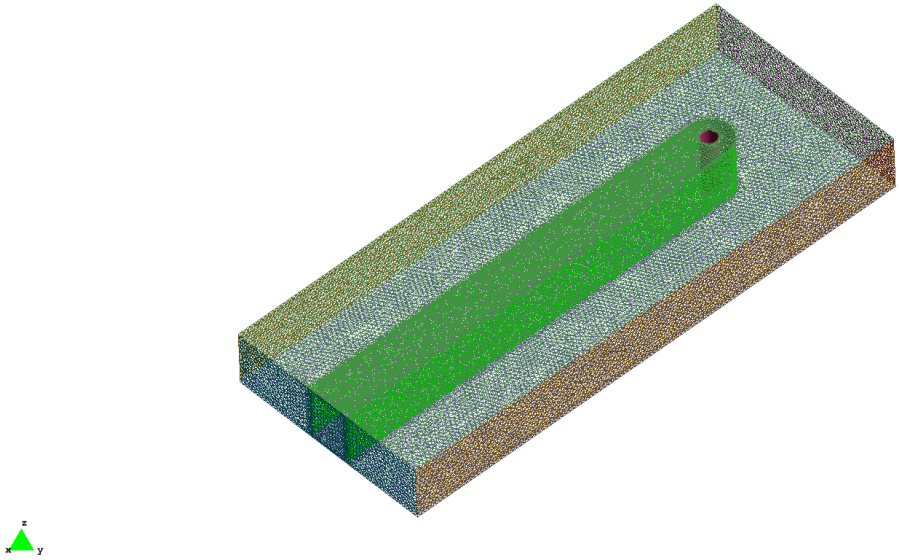
The objective of this tutorial is to do a postprocess analysis of an already calculated fluid simulations, no preprocess option is used.

Not only the model is already meshed and the constraints are assigned, but also the results have been calculated. For more information about the preprocess part of GiD, please check the preprocess tutorials.

In this tutorial, the model *Cylinder.bin* has been used. The problem type used to do this simulations is Tdyn, particularly the Ransol model. Tdyn is a fluid dynamic (CFD) simulation environment based on the stabilized Finite Element Method.

Steps followed in this tutorial:

- Loading the model
- Changing mesh styles
- Visualization of results
- Creating images



7.1 Loading the model

Loading the model

There are two ways to load the results simulation information into GiD:

- If the model has been calculated inside GiD, and so the results are inside a GiD project, then just loading the GiD project and the changing to postprocess mode is enough. This can be achieved clicking on this icon:

, or selecting the **Files->Postprocess** menu entry.

- If only a mesh and results file(s) is present then GiD should be started, and switched to postprocess

mode) before loading the file(s).

In this tutorial, only *Cylinder.bin* is present, with the postprocess information, so the steps to follow are:

1 . Start GiD
2 . Switch to postprocess mode: or **Files->Postprocess**
3 . Open the model with: **Files->Open**, **Ctrl-o** or clicking on

7.2 Changing mesh styles

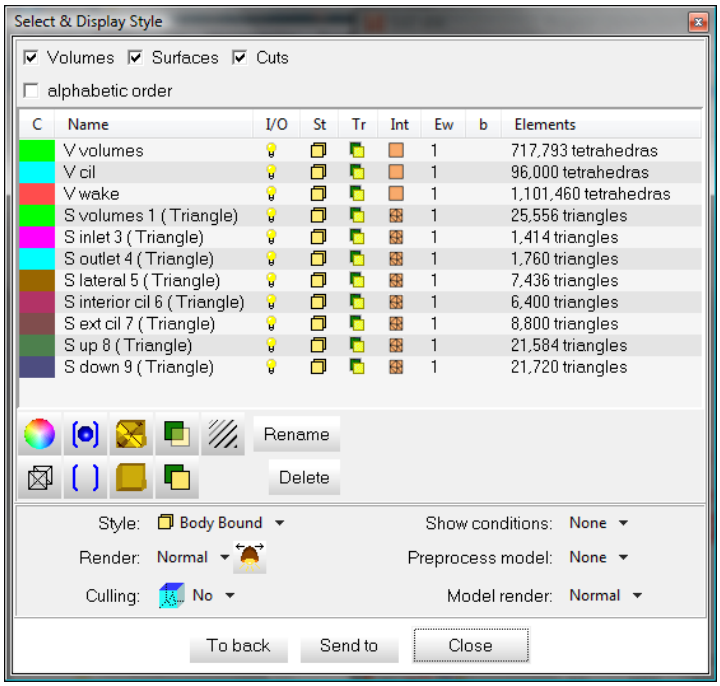
Through the '**Select & Display style**' window several options can be specified for volumes, surfaces and cuts. Among these options volumes, surfaces and cuts can be switched on and off, their colour properties can be changed, and their transparency too.

Other interesting options which can be changed are the style of the set and the width of the edges.

From this window, volumes, surfaces or cuts can be deleted or their names can be modified.

To access this windows select **Windows->View Style** or **Utilities->View style** o click the icon:

Mesh styles can also be changed clicking on the icon , placed in the left icon bar. This style affects all sets of the model.



1 . Select **Utilities->View style...**using the menu bar or clicking on

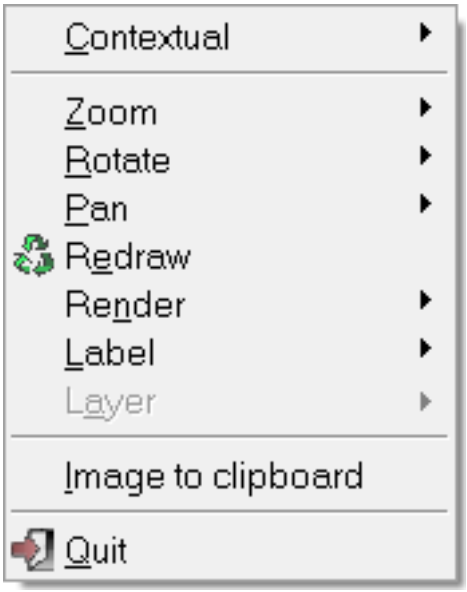
Our model is composed by 11 layers. Now we will change all the layers to boundaries style.

2 . Select all the layers.
3 . Select **Boundaries** in the Style option, at the bottom of the window. You can also do it clicking on in the St column or the same icon in the main window.
4 . By selecting one set and right-clicking on one corresponding style icon at the 'St' column, the style for the selected mesh is changed. Bear in mind that a boundary visualization of the surface of a

sphere nothing will be visualized, as a sphere surface has no borders.

5 . Play a little with the options of these windows, but to continue the tutorial, let a **Boundaries** style selected for all meshes

6 . Click on **Close** button



Mouse menu

Clicking the right mouse button on the main graphical window opens an on-screen menu with some visualization options. To select one of them, use the left or right mouse button; to quit, left-click anywhere outside the menu.

The first option in this menu is called **Contextual**. You can select from different options relevant to the function currently being used.

Zoom, Rotate, Pan and **Redraw** are the same options as in preprocess mode.

Label node, element or result shows a label with the current node number, element number or current visualizaed result's value for the selected nodes or gauss points.

Menu: View->Render

In order to improve the visualization and to get a more realistic view we can change the render mode. There are three main options:

- Normal: With this visualization mode no illumination is applied to the model, and so no 3D perception can be achieved. Ideal for 2D models and quick 3D temporal visualization.
- Flat lighting: Solid model with flat illumination: the triangle faces can be distinguished as no smoothing is done between triangles.
- Smooth lighting: Solid model with smooth illumination (better quality), i.e., illumination is smoothed between triangles if their angle is bigger than the one specified in **Options->Geometry->Border angle**.

In the Render menu we can also set more options:

- Change light direction: With this option you can change the vector of the light direction interactively;

this can also be done by entering the vector components in the command line.

- Reflection: with this option an environment or a pattern will be reflected on the surfaces.

For the moment we will select the **Normal** render.

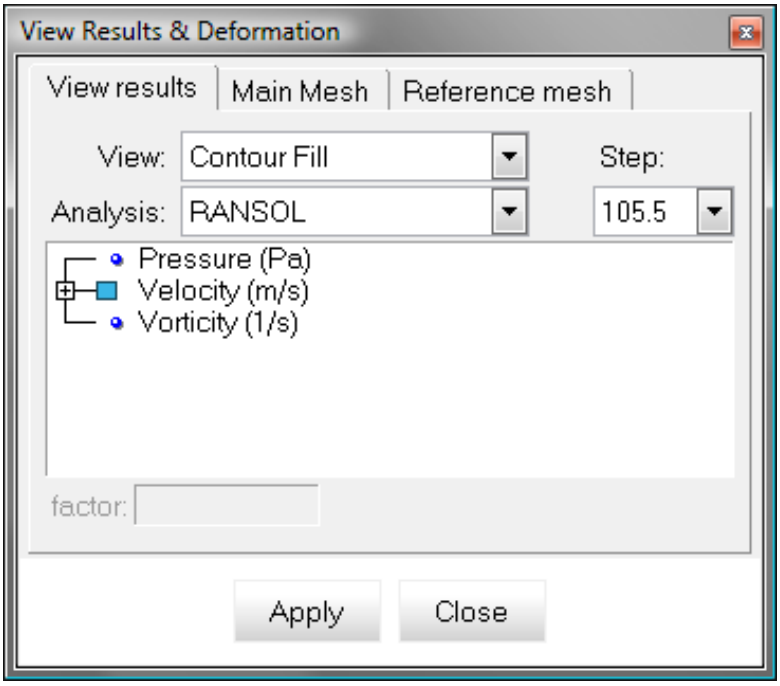
1 . Select **View->Render->Normal** throught the menu bar or in the **mouse** menu select **Contextual->Render->Normal**.

7.3 Viewing the results

Several results had been calculated for several time steps. You can check these results through the **Results** menu or opening the **View Results** window.

Menu:View Results

Window->View Results...



7.3.1 Iso surfaces

Menu: View results->Iso Surfaces

With this result visualization a surface, or line, is drawn passing through all the points which have the same result's value inside a volume mesh, or surface mesh. To create isosurfaces there are several options.

1 . Select **View results->Iso Surfaces->Automatic Width->Velocity(m/s)->|V|**throught the menu bar or clicking on 

After choosing the result, you are asked for a width. This width is used to create as many isosurfaces as are needed between the Minimum and Maximum defined values (these are included).

2 . Select the default value.

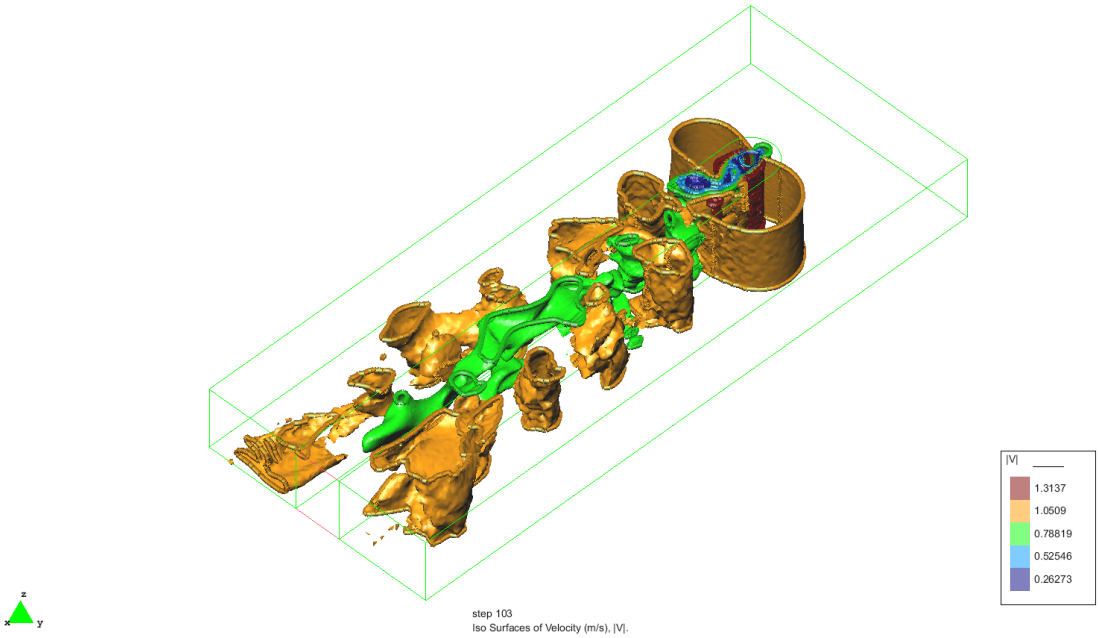
3 . Select **View->Render->Smooth** in order to get a better view.

Several configuration options can be set via the Options menu.

Menu: Options->Iso surfaces

In order to see the inner zones we will set the transpacency on the iso surfaces.

4 . Select**Options->Iso surfaces->Transparency->Transparent**
5 . Move the model to see the inner zones
6 . Select**Options->Iso surfaces->Transparency->Opaque**



Other interesting options are:

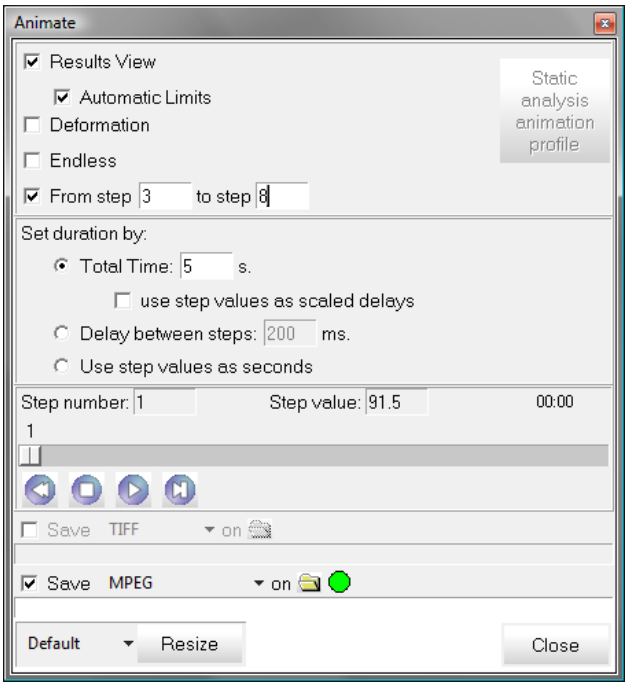
- **Options->Iso surfaces->convert to cut** which consolidates the isosurface as mesh which can be exported to a file.
- **Options->Iso surfaces->result contour fill** allows to draw the contour fill of any result over the isosurface. Select this option and then do a contour fill of any result.
- **Options->Iso surfaces->draw iso lines** this options allows the user to switch isolines of surafaces on or off.

7.3.2 Animate

Menu:Window->Animate...

This window allows the user to animate the current visualized results.

If only one step is present, then the **Static analysis animation profile** button is enabled so that a custom animation profile can be step to animate that one step.



If one result has several steps you can visualize them in an animation. In this case we will use the iso surfaces result.

1 . Select **View->Render->Normal**
2 . Select **Window->Animate...**to open the animation window

Please notice that we have from step 1 to 13. We will do the animation only of some of these steps.

3 . Check the **From step** option and set 3 **to step** 8
4 . Select the **Delay between steps** option and set it to 800 ms. The animation should take 4 seconds
5 . Try it clicking on the **play** icon

We will record a video during the animation.

6 . Once the animation is finished check the **Save** option

You can choose from several video formats.

7 . Select AVI/mjpeg

(to include this animation in a MS PowertPoint an apropiate codec is needed like the one supplied with Combined Community Codec Pack, CCCP)

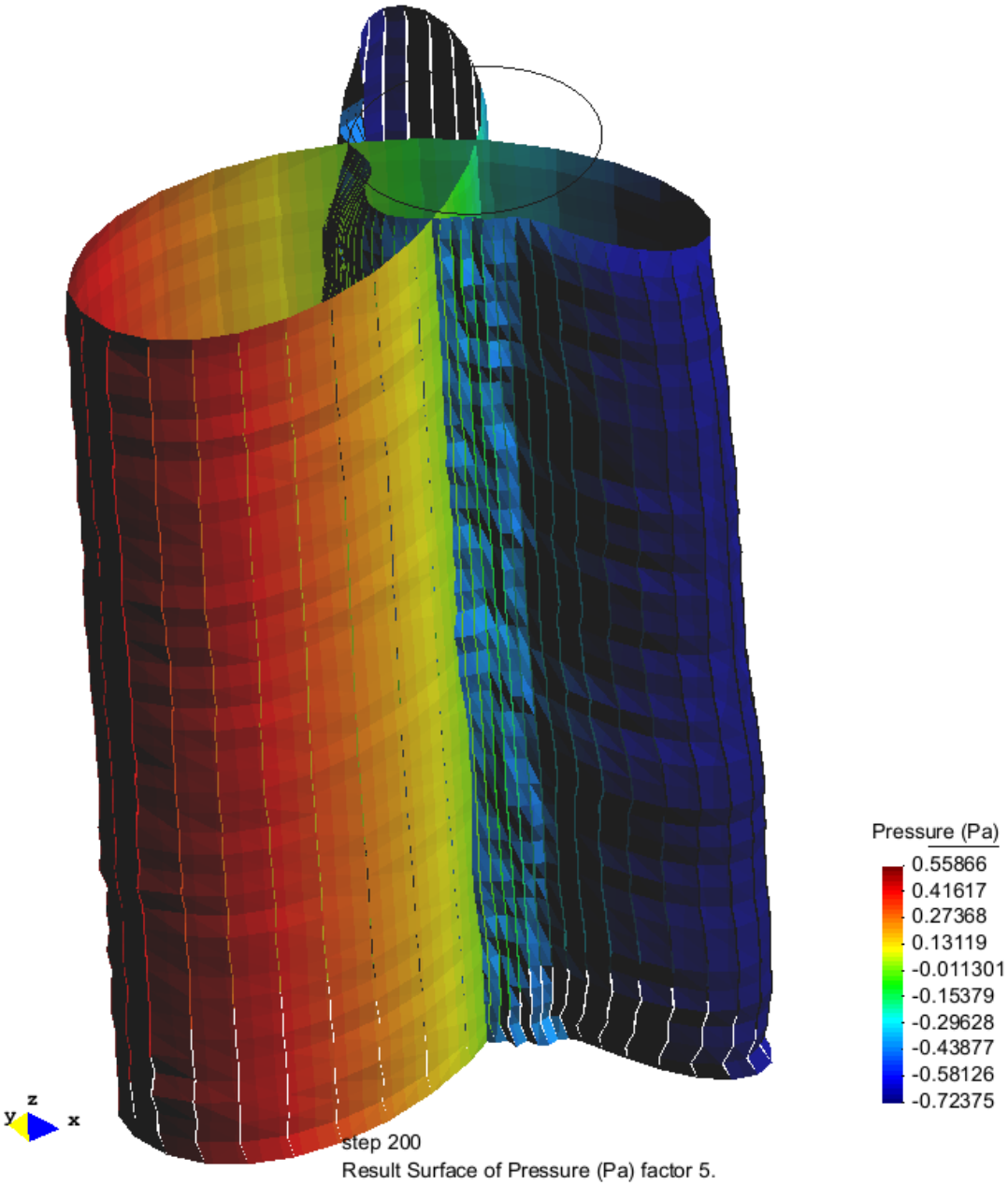
8 . Please select a folder where the video will be saved clicking on the **folder** icon or writing the path
9 . Please click on the **play** button and the recording will begin. This step could take a little bit long. Wait until the red circle turns to green
10 . **Close** the Animate window

Now we will visualize another result but before we will clear all the results.

11 . Select **View results->No results**through the menu bar or using the icon 

7.3.3 Result surface

Another result visualization of interest is this one:



To get this visualization follow these steps:

1 . Switch off all the sets except **S interior cil 6**

2 . Select **View results->Result surface-> Pressure**. A surface will be drawn which results from moving the nodes along its smoothed normal according to the results value for this node

3 . Enter 5 as factor in the command line

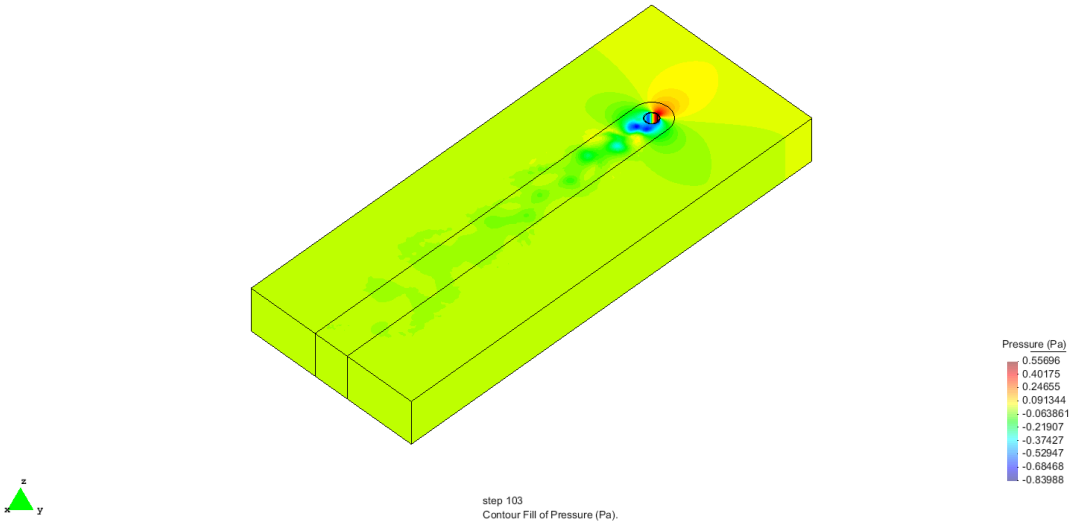
4 . Select **Options->Result surface->Show elevations->None**

5 . Select **Options->Result surface->Show elevations->Contour fill**. With this last option the surface is colored according to the pressure value

Play with the other options as you will.

7.3.4 Contour fill, cuts and limits

Contour fill



Menu: View results->Contour Fill

1 . Please select **View results->Contour Fill->Pressure (Pa)** through the menu bar, or clicking on  or using the **Window->View results...** window

This option allows the visualization of coloured zones, in which a scalar variable or a component of a vector varies between two defined values. GiD can use as many colours as permitted by the graphical capabilities of the computer The number of colours can be set through **Options->Contour->Number of colours**. A menu of the variables to be represented will be shown, and the one that is chosen will be displayed using the default analysis and step selected.

In the model the pressure has been calculated. We can visualize the result for each step in a contour fill.

You can choose the step that you want to view through the **View results** window or clicking on 

2 . Select the step 103

Several configuration options can be set via the Options menu.

Menu: Options->Contour

You can change the color scale in orther to get a more comfortable view. You can select several predefined color scales. The default scale is *standard,* starting from blue (minimum) through yellow and green, to red (maximum).

1 . Select **Options->Contour->Color Scale->Inverse Standard**

You can also set your own scale.

2 . Select **Options->Contour->Color scale->User defined...**

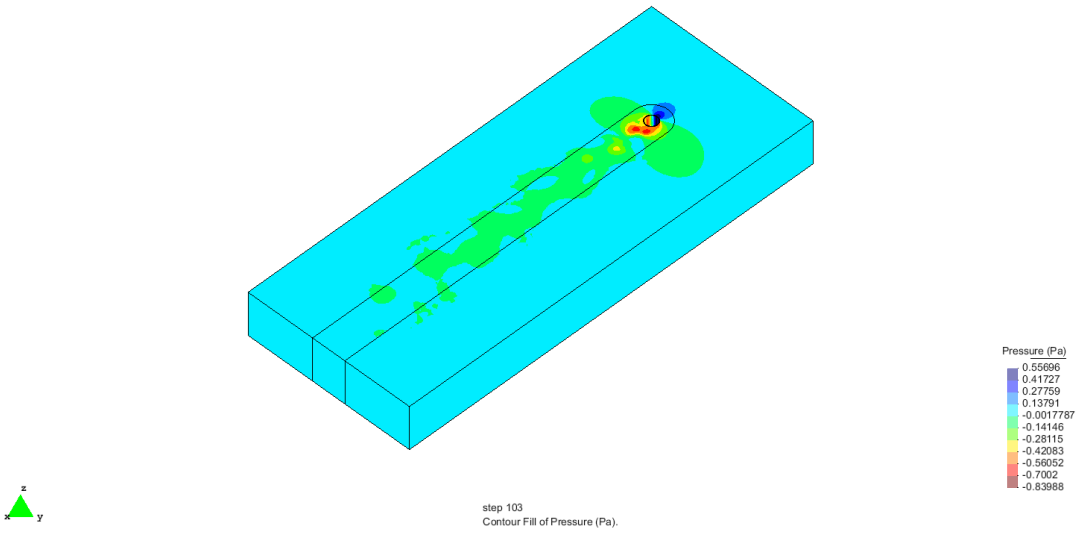
In this window you can change the number of diferent colors used in the scale. If you need more

accuracy you can increase this number, or decrease it for a higher contrast.

3 . Change the number of colors to 10

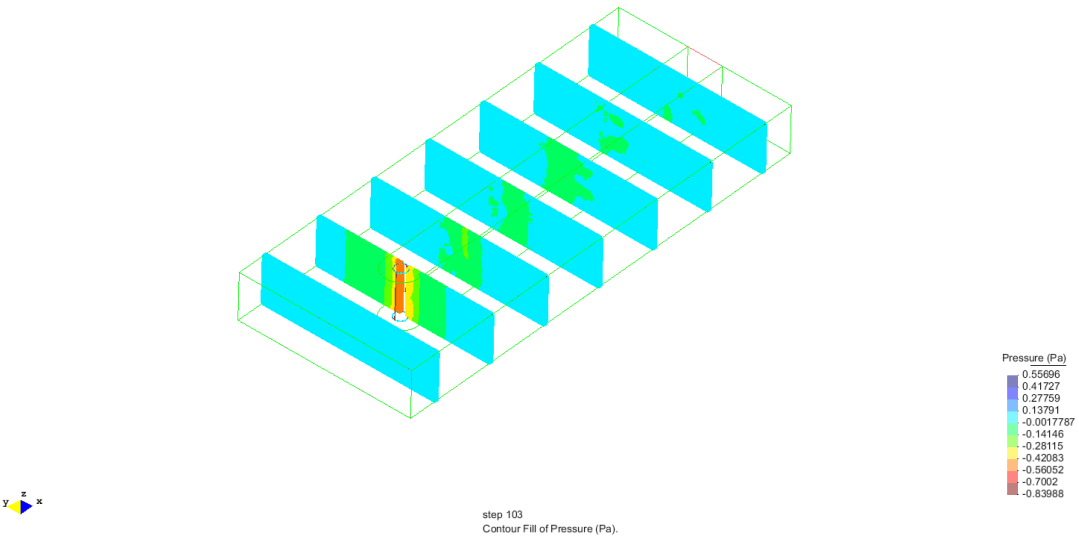
4 . Click on **Apply** button

5 . Click on **Close** button



Cuts

In order to view the inner zone we will do several cuts along the model.



Menu: *Do cuts*

In order to make it easier first we will change the plane visualization.

1 . Please select **View->Rotate->Plane XY(Original)** through the menu bar, **Rotate->Plane XY(Original)** throught the contextual menu or clicking on  or  Now you have a top view of the model.

2 . Select **Do cuts->Cut plane->Succession** through the menu bar or clicking on  and then 

With the **succession** option you specify an axis that will be used to create cut planes orthogonal to this axis. The number of planes is also asked for.

3 . Please specify the line through the X axis in the middle of the model and ask for 7 cuts. You should obtain 7 parallel planes to Y axis.

4 . Now change the display style (**Utillities->View style**) in order to see only the cuts. You can see that several layers had appeared a prefix like **CCutSetX** indicating which mesh or set has been cut. These names can always be changed through the **Window->View Style**. Select all the layers except the cuts and change their style to **Boundaries**. You can rotate the model in order to see the contour fill result on the cut planes.

5 . In the same window select all the **CCutSetX** and click on **Delete** button in order to delete all the cuts.

6 . Select **Options->Contour->Reset** all in order to set all the defaults options.

Define limits

You can set the limit values for the contour fill. In our case we only want to see the positive values. In order to do this we will set the minimum value to 0.

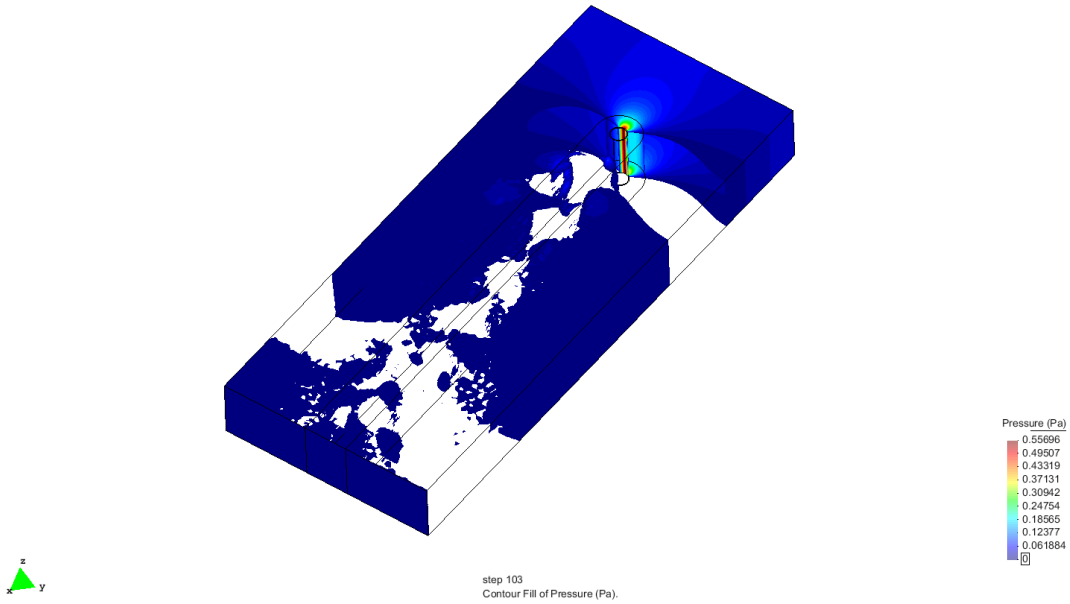
1 . Select **Options->Contour->Define Limits...** through the menu bar or clicking on 

Choosing the first option the Contour Limits window appears. With this window you can set the minimum/maximum value that Contour Fill should use.

2 . Check the Min checkbox
3 . Change the value to 0
4 . Click on the **Apply** button
5 . Click on the **Close** button

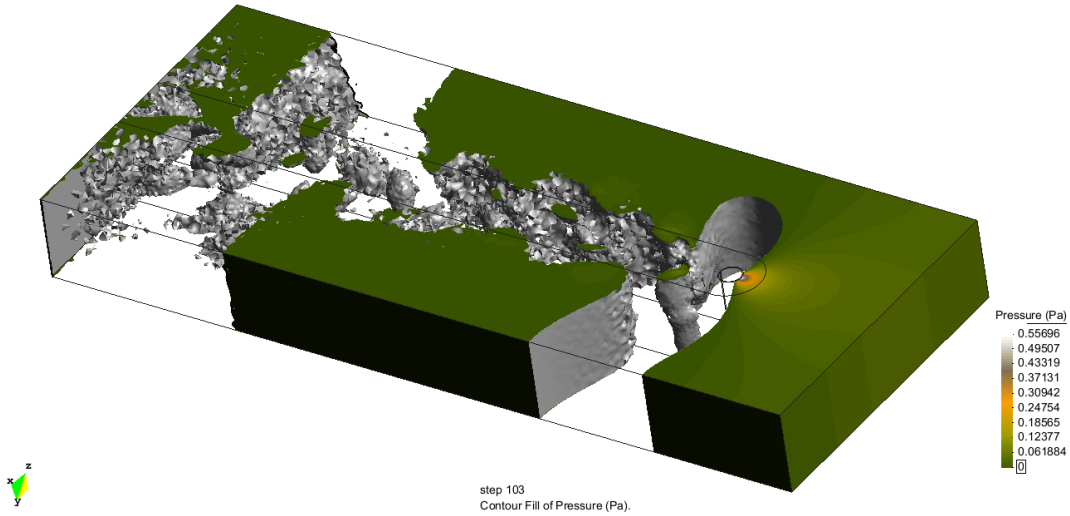
Outliers will be drawn in the colour defined in the Out Min Colour option. In order to view it better we will change this color to transparent.

6 . Select **Options->Contour->Min Options->Out Min Color->Transparent**



7.3.5 Combined results

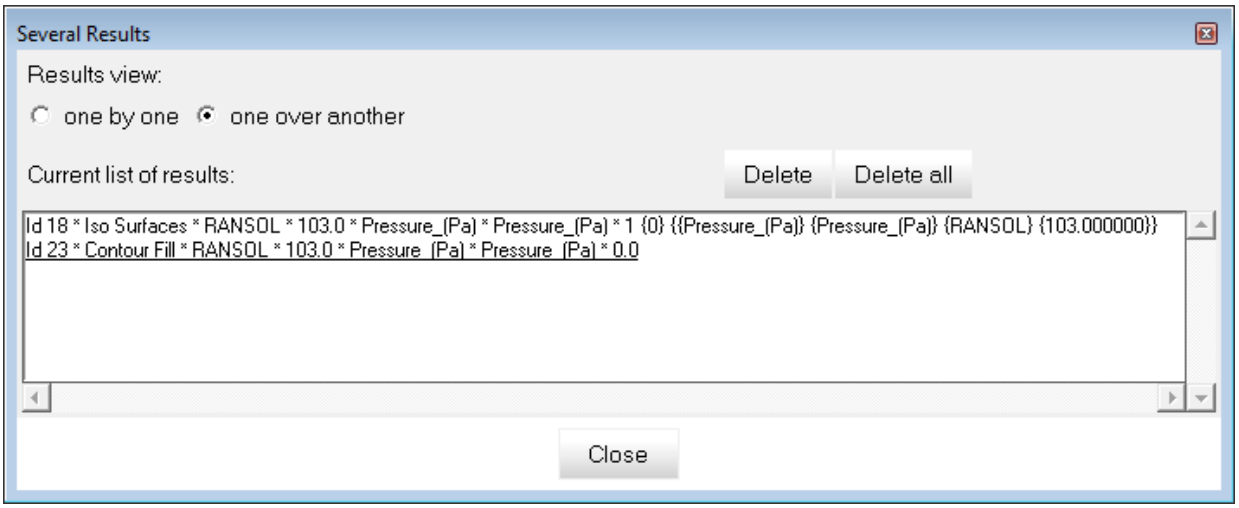
An interesting postprocess options is to combine several result visualizations, like this one:



To get this view follow these steps:

1 . Clear all results visualizations with **View Results->No results** or the icon 

2 . Select **Window->Several results**. Then following window appears:



3 . In this window select **one over another**. With this option GiD is told to visualize one result over another

4 . Select **View Results->Default Analysis/Step->Ransol->103**

5 . Select **View Results->Iso surfaces->Exact->Pressure** through the menu bar or clicking on the 

6 . In the following questions: How many **isosurfaces**? Enter 1

7 . Enter the 1 value ...? Enter 0

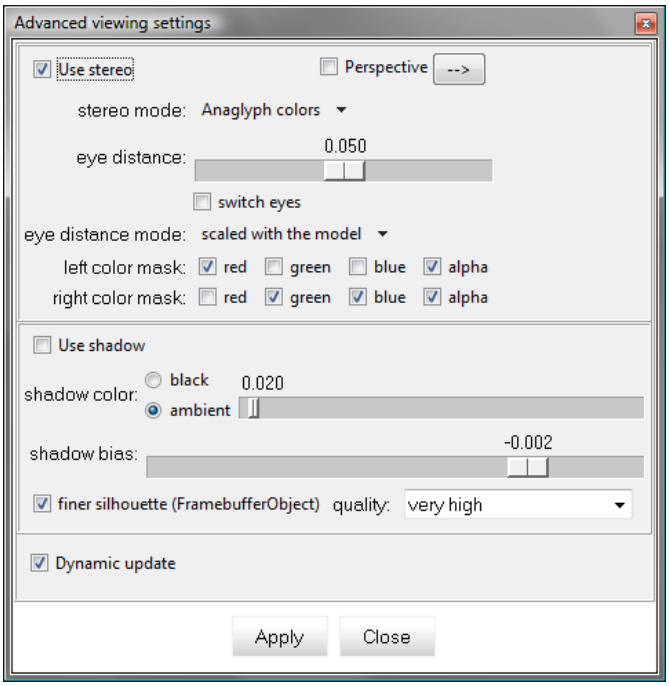
8 . Select **View Results->Contour Fill->Pressure**

9 . Set the **minimum** value to 0

10 . Select **Options->Contour->Min options->Out min color->Transparent**

11 . Select **Options->Contour->Color scale->Terrain Map**

7.3.6 Stereo mode (3D)



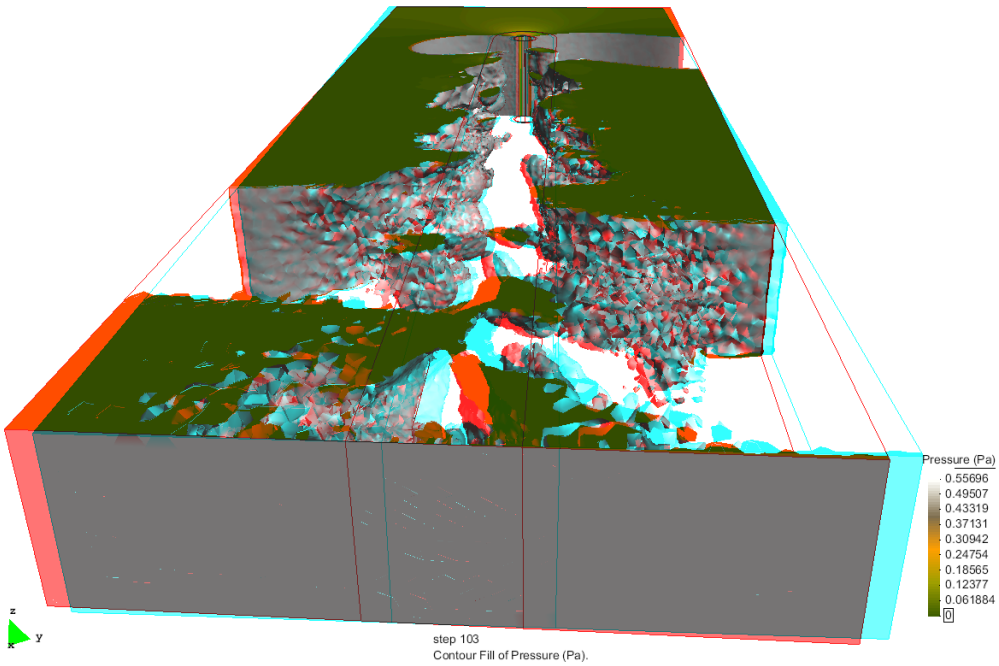
Menu: View->Advanced options...

If you have an **anaglyphic** glasses you can try this option. The model can be set as an anaglyphic image in order to provide a stereoscopic 3D effect, when viewed with 2 color glasses (each lens a

chromatically opposite color, usually red and cyan).

Anaglyphic images are made up of two color layers, superimposed. Since the glasses act as red and cyan filters we should be careful with the model's colors. To avoid problmes we will change the contour fill color scale.

1 . Select **Options->Contour->Color Scale->3D Anaglyphs**
2 . Select**View->Advanced options...**
3 . Check the **Use stereo** option
4 . Check the **Dynamic update** option in order to change the options without the need to click the Apply button
5 . Set the eye distance to the value where you can see the 3D effect
6 . Unheck the **Use stereo** option
7 . **Close** the window
8 . Select **View results->No Results**
9 . Change the **view style** to boundaries for all the layers, like in **Changing style** chapter

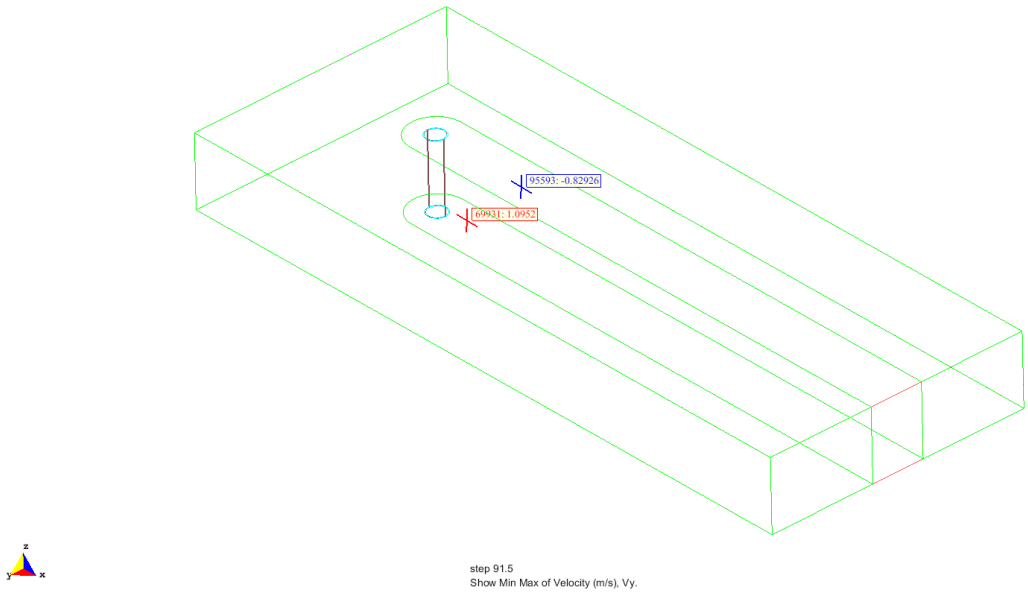


7.3.7 Show Min Max

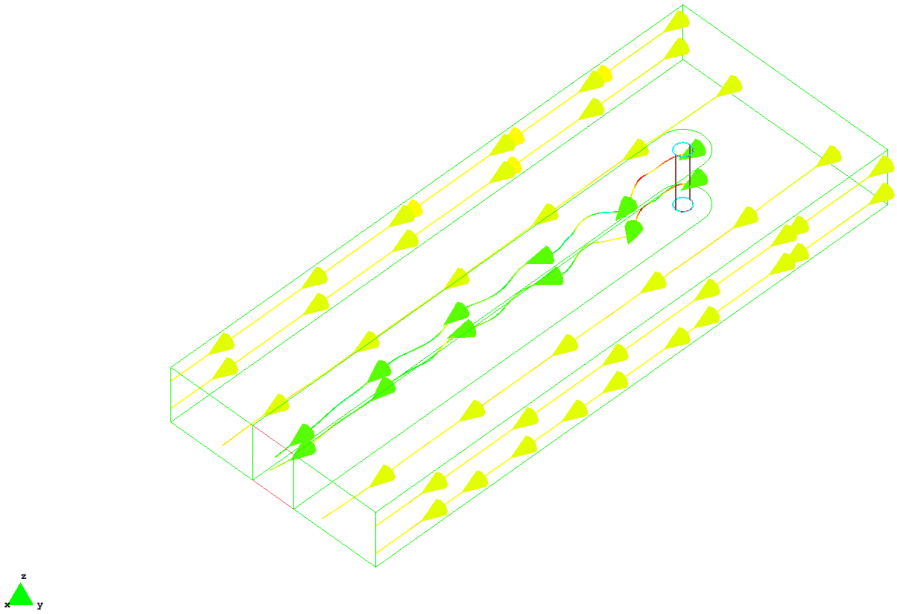
Menu: View results->Show Min Max

With this option you can see the minimum and maximum value of the chosen result in the chosen analysis step. In our case we will choose the Vy component of velocity result for the first analysis step.

1 . Select **View results->Default Analysis/Step->RANSOL->91.5** throught the menu bar or clicking on
2 . Select **View results->Show Min Max->Velocity (m/s)->Vy** throught the menu bar or clicking on . The label shows the node number and the value of the result.
3 . Select **View results->No Results**



7.3.8 Stream lines



Menu: View results->Stream Lines

With this option you can display a stream line, or in fluid dynamics, a particle tracing, in a vector field.

1 . Select **View results->Stream Lines->Velocity (m/s)** throught the menu bar or clicking on 

After this the program asks you for a point from which to start plotting the stream line. This point can be given in several ways. We will select the nodes using the *Along line option.*

2 . Open the mouse menu and select **Contextual->Along line**

With this option you can define a segment along which several start points will be chosen. The number of points will also be asked for, including the ends of the segment. In the case of just one start point, this will be the center of the segment.

We want to create several stream lines along the model doing 2 lines.

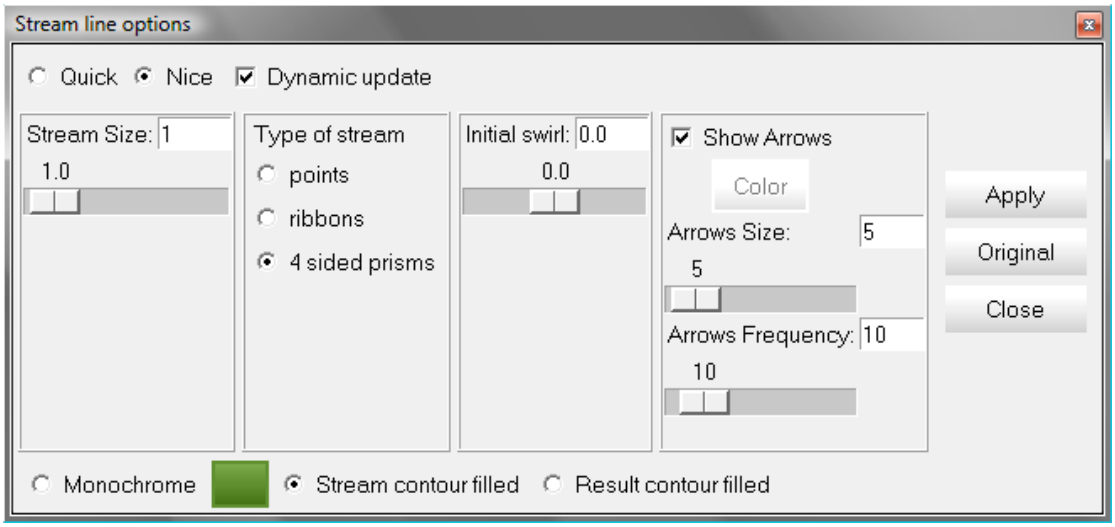
3 . Write the **initial** point in the command line 10 15 3
4 . Write the **final** point in the command line 10 -15 3
5 . You are asked for the **number of points** along the line. Choose 5.

The first line with 5 stream lines is created.

6 . Write the **initial** point in the command line 10 15 7
7 . Write the **final** point in the command line 10 -15 7
8 . You are asked for the **number of points** along the line. Choose 3.

The second line with 3 stream lines is created.

9 . Click the middle mouse button or press the **Esc** key in order to finish the operation.



Several configuration options can be set via the Options menu.

Menu: Options->Stream lines

The options can be also managed through the **Size & detail** window.

10 . Select **Options->Stream lines->Size & detail...**
11 . Check the **Dynamic update** option
12 . Select **Options->Contour->Color Scale->Standard**
13 . Select Stream contour filled

The stream lines will be drawn with the colors used in the velocity contour fill.

14 . Set 5 for the **Arrows Size** option
15 . Set 10 for the **Arrows Frequency** option
16 . Check the **Show Arrows** option
17 . Close the window
18 . Select **Options->Stream lines->Delete**
19 . Select **all** the stream lines with the mouse
20 . Click the middle mouse button or press the **Esc** key in order to finish the operation.
21 . You may play with the diferent stream types: points, ribbons or 4 sided prisms. If the ribbons type is selected you may adjust the initial swirl to rotate the ribbon.

NOTE: A way to achive the best results is to first create a cut of the volume mesh throught the region of interest and then use these nodal information as support to create *stream lines* and its options: *along line, in a quad,* etc.

7.3.9 Graphs

Menu: View results->Graphs

From this menu several graphs types can be created, we will try some of them. Graphs are supported for results defined over nodes.

The **Point evolution** graph displays a graph of the evolution of the selected result along all the steps, of the default analysis, for the selected nodes.

1 . Select **View results->Graphs->Point evolution->Velocity(m/s)->|V|**
2 . Write 20 0 4 in the command line in order to specify the point.

The **Line graph** displays a graph defined by the line conectig two selected nodes of surfaces or volumes, or any arbitrary points on any projectable surface and in any position.

3 . Select **View results->Graphs->Line graph->Velocity(m/s)->|V|**
4 . Write 3 0 4 in the command line in order to specify the initial point.
5 . Write 50 0 4 in the command line in order to specify the final point.

Once the graphs are created are showed in the main window, several graphs can be showed at the same time. You can swap this view with the model view.

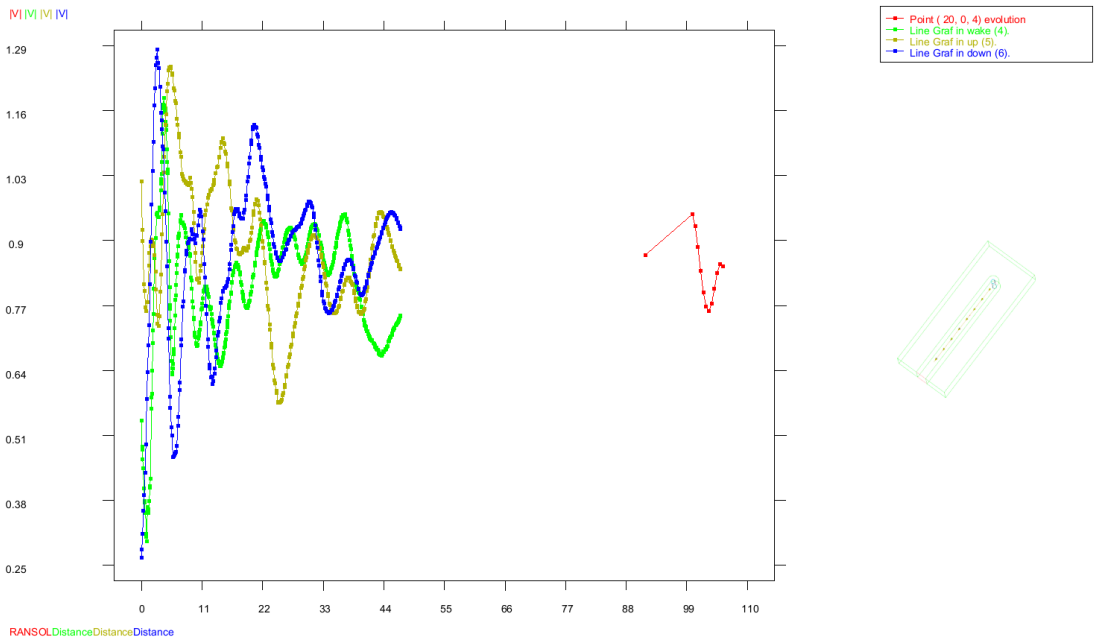
6 . Select *View results->No Graphs* throught the menu bar or clicking on:

You can also select the **Model view** option in order to get a view of the model while you are working with the graph.

7 . Select **Options->Graph->Model view->Show**
8 . Choose Yes

By default the model view's size it's a 50% of graph's size.

9 . Select **Options->Graph->Model view->Size**
10 . Choose 75



You can also create the graphs and change their options using the graph window.

1 . Select **Window->View graphs...**
2 . Select the **Graph management** tab
3 . Select the first graph of the list Point_(_20,_0,_4)_evolution
4 . Click the Delete button

The graph size is readapted. We will change several style options of a graph.

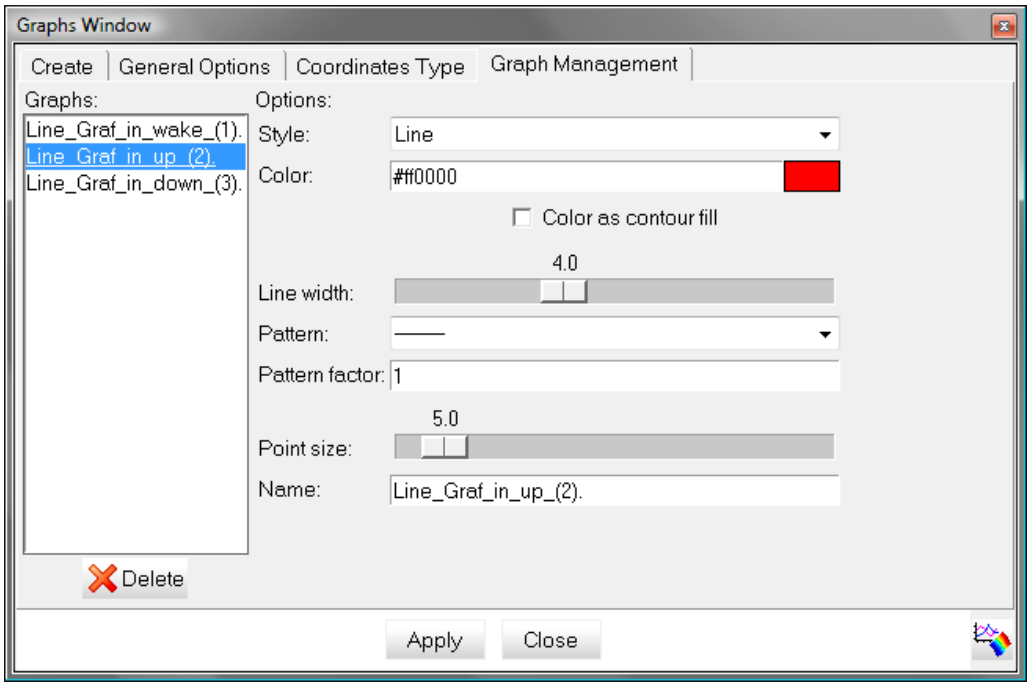
5 . Select the second grahp of the list Line_Graf_in_up(2).
6 . Select Line in the **Style** option
7 . Set to red the **Color** option. You can do it writing #ff0000 or selecting the red clicking on the right color window
8 . Set to 4.0 the **Line width**
9 . Click on Apply button
10 . Click on Close button

We can export the graph information in order to open it later with GiD.

11 . Select **Files->Export->Graph->All**. You are asked for the location where to save the .grf file.
12 . Choose the location

Now you can import the selecting **Files->Import->Graph**

13 . Select **Options->Graphs->Clear graphs**in order to delete all the graphs

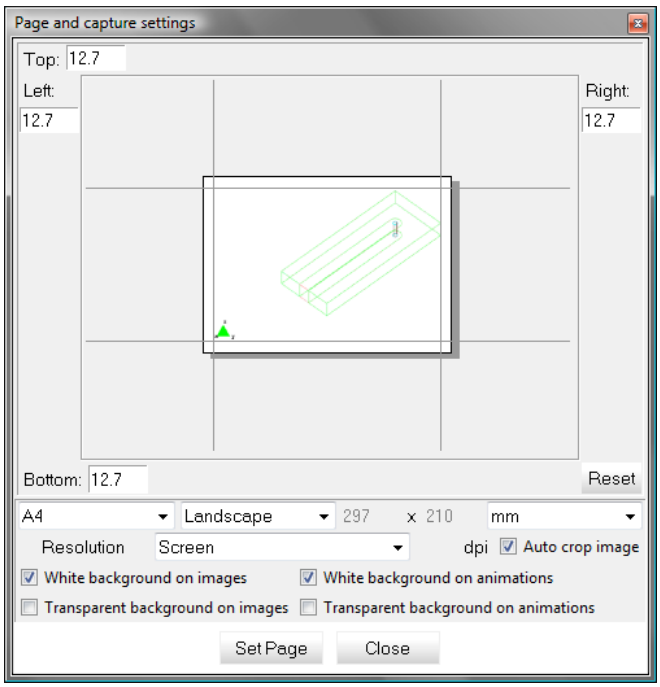


7.4 Creating images

Menu:Files->Page and capture settings...

Finally we will take some snapshots of our model. You can save images in several formats. The properties of the image (resolution, size, etc.) can be assigned in Page and capture settings option.

1 . Select **Files->Page and capture settings...**
2 . Check the **Auto crop image** option in order to cut the image in the model limits
3 . Click on **Set Page** button
4 . Click on **Close** button



Menu:Files->Print to file

This option asks you for a file name and saves an image in the required format with the defined properties in **Page and capture settings**.

1 . Select **Files->Print to file->PNG...** throught the menu bar or clicking on  and then 

2 . Choose the location where you want to save the image

3 . Choose a name for the file

4 . Click on Save button

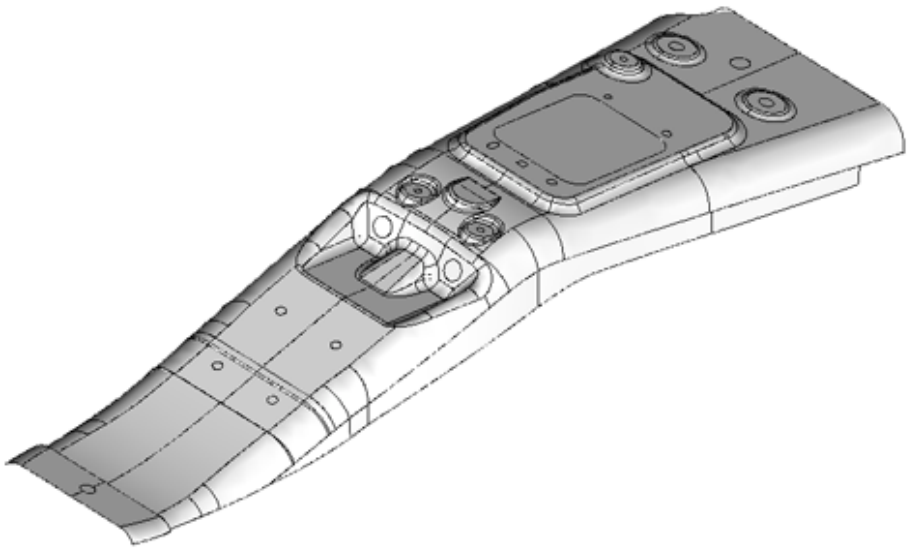
8 IMPORTING FILES

IMPORTING FILES

The objective of this case study is to see how GiD imports files created with other programs. The imported geometry may contain imperfections that must be corrected before generating the mesh.

For this study an IGES formatted geometry representing a stamping die is imported. These steps are followed:

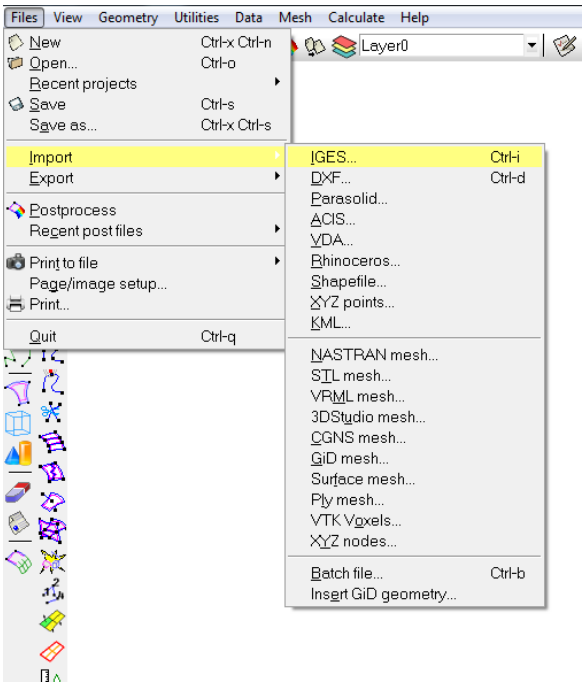
- Importing an IGES-formatted file to GiD
- Correcting errors in the imported geometry and generating the mesh
- Generating a conformal mesh and a non-conformal mesh



Pice provided by courtesy of PSA DEGAD-MAC AIE

8.1 Importing on GiD

GiD is designed to import a variety of file formats. Among them are standard formats such as IGES, DXF, or VDA, which are generated by most CAD programs. GiD can also import meshes generated by other programs, e.g. in NASTRAN or STL formats.



The file importing process is not always error-free. Sometimes the original file has incompatibilities with the format required by GiD. These incompatibilities must be overcome manually. This example deals with various solutions to the difficulties that may arise during the importing process.

8.1.1 Importing an IGES file

1 . Select **Files->Import->IGES** …

2 . Select the IGES-formatted file "base.igs" and click **Open**.

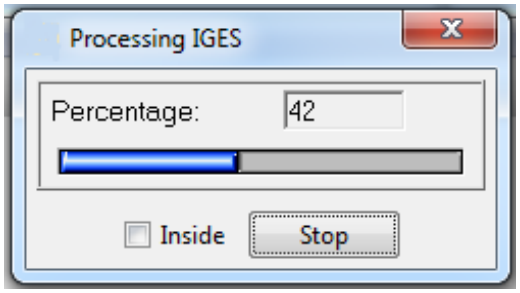


Figure 1. Reading the file.

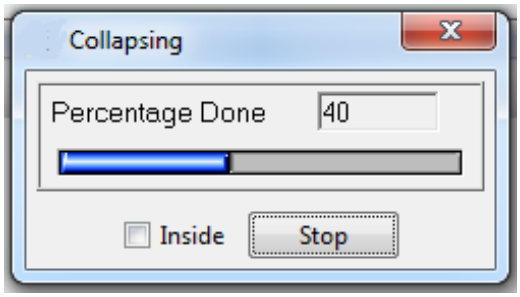


Figure 2. Collapsing the model.

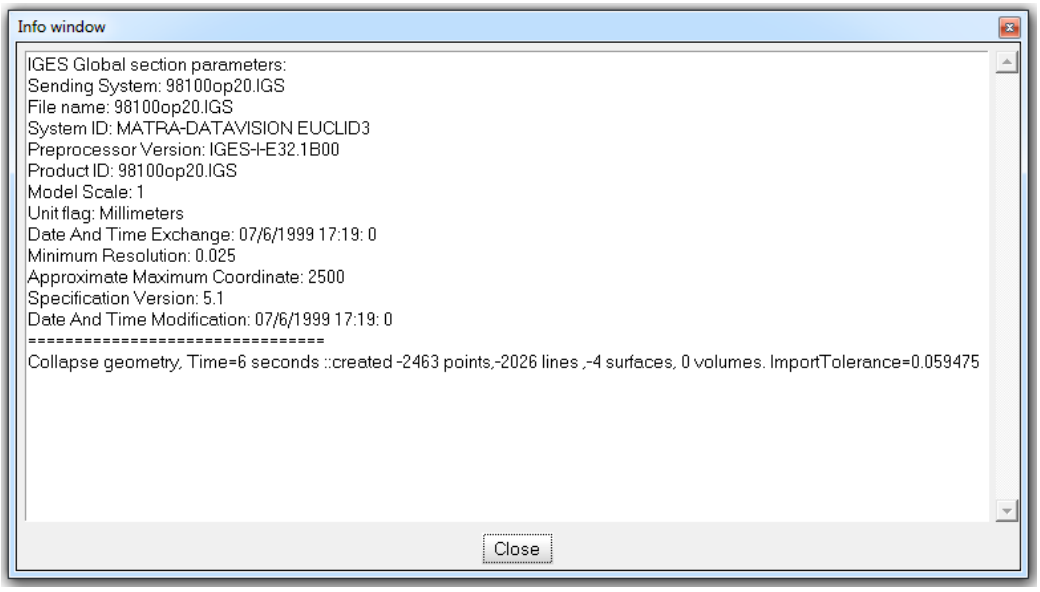


Figure 3. Importing process information.

After the importing process, the IGES file that GiD has imported appears on the screen.

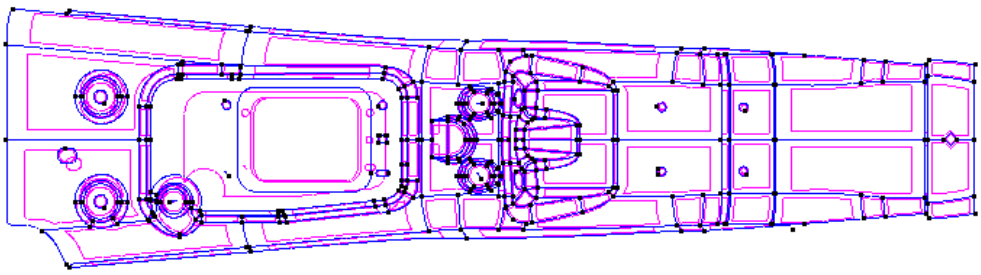


Figure 4. File "base.igs" imported by GiD.

NOTE: One of the operations in the importing process is collapsing the model (Figure 3). We say that two entities collapse when, the distance between them being less than the **Import Tolerance**, they become one.

The **Import Tolerance** value may be modified by going to the **Utilities** menu, opening **Preferences**, and bringing up the **Exchange** card. By default, the **Automatic import tolerance value** is selected. With this option selected, GiD computes an appropriate value for the **Import Tolerance** based on the size of the geometry.

Collapsing the model may also be done manually. This option is found in **Geometry->Edit->Collapse->Model**.

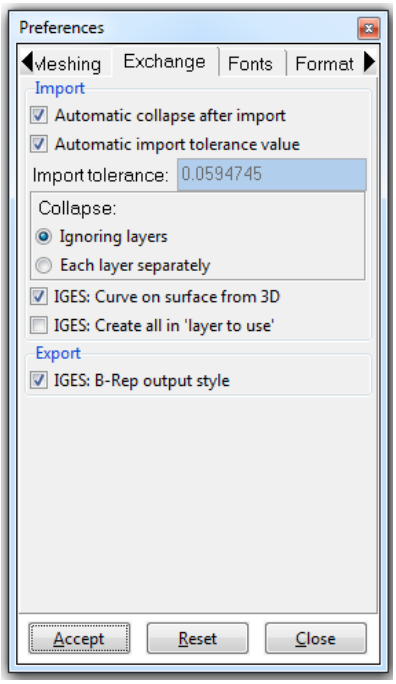


Figure 5. The Preferences window

8.2 Correcting errors in the imported geometry

The great diversity of versions, formats, and programs frequently results in differences (errors) between the original and the imported geometry. With GiD these differences might give rise to imperfect meshes or prevent meshing altogether. In this section we will see how to detect errors in the imported geometry and how to correct them.

Importing the same file with different versions of GiD might produce slight variations in the results. For this tutorial it's necessary load the project **"imported48.gid"**, which contains the original IGES file translated into GiD format.

8.2.1 Meshing by default

1 . Select **Mesh->Generate Mesh**.

A window comes up in which to enter the maximum element size for the mesh to be generated. Leave the default value provided by GiD unaltered and click **OK**.

When the **GiD** finishes the meshing process, an error message appears (see Figure 6). This error is due to a defect in the imported geometry. As the window shows, there have been errors meshing surface number 149.

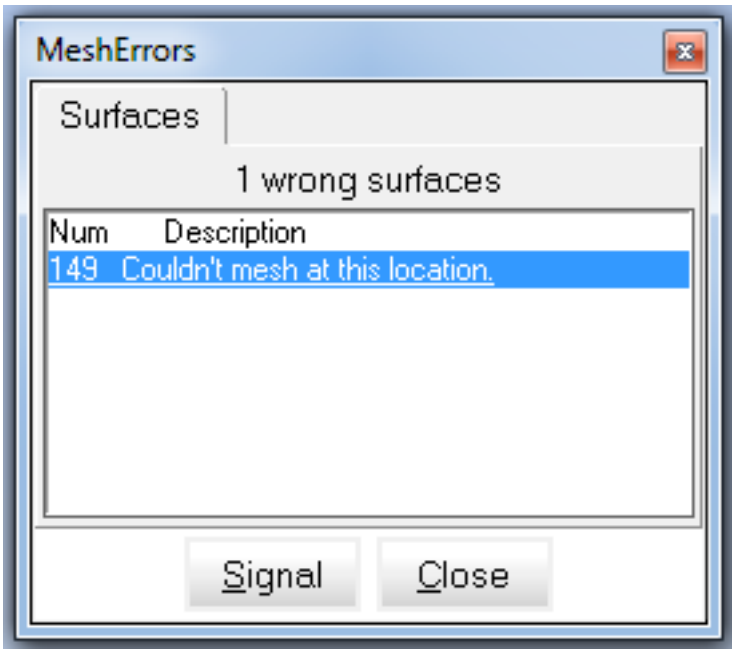


Figure 6. Dialog window warning of an error found when meshing surface 149.

In this part of the tutorial we focuson repairing surface number 149.

To locate surface 149, select the line "**149 couldn't map this point** " in the dialog box and press the **Signal** button (the same effect is obtained by double-clicking over the message with the left mouse button).

NOTE : If user clicks the right button over a message in the Mesh Errors window, three options are displayed: "Signal problematic point", "More help…" or "List…" The first option is the same as the Signal button, while the "List…" option presents a list of the problematic geometrical entities to make selection easier when performing some common procedures (like sending the entities to a separate layer, erasing the entities, etc…). The "More help…" option gives advice about to correct the geometrical model so the mesh can be generated.

NOTE : *The* **Mesh Errors** *window can be recovered while dealing with the model by selecting the "***Show errors***…" option in the* **Mesh** *menu*.

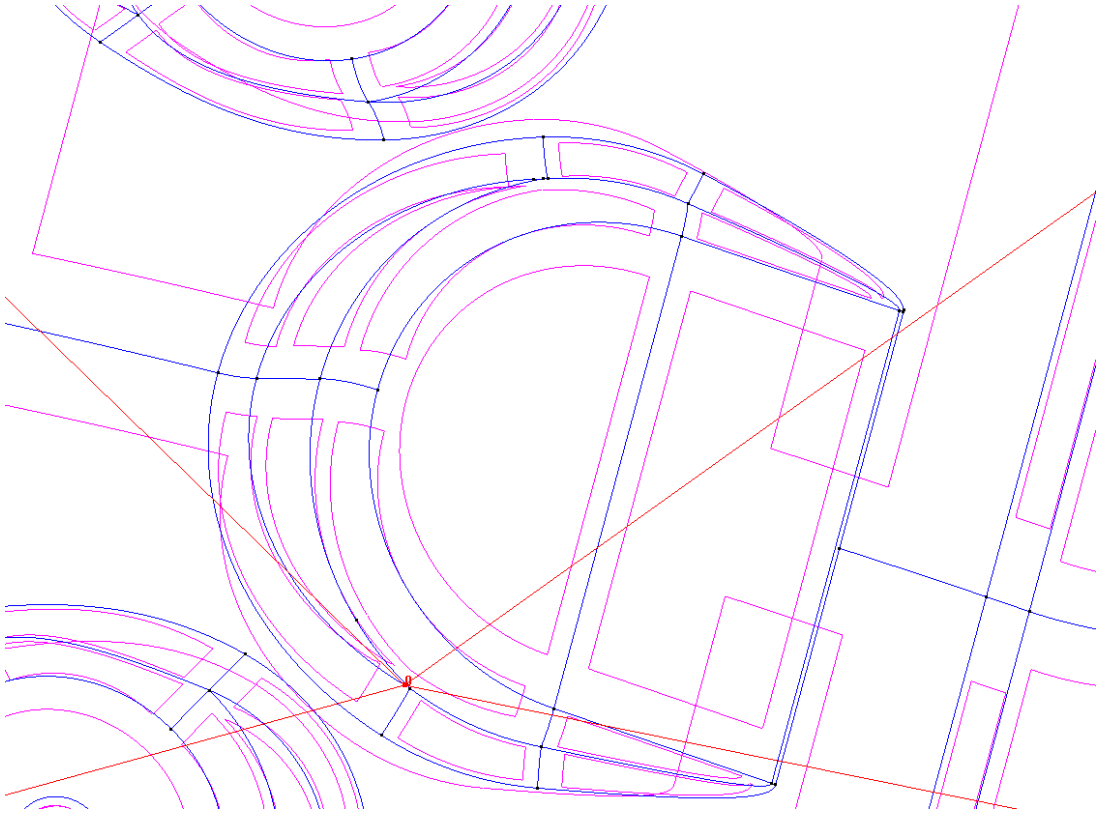


Figure 7. Signaling surface 149.

NOTE : The identifiers of the entities vary each time the instruction **Mesh->Generate Mesh** is executed.

8.2.2 Correcting surfaces

1 . With the **View->Zoom In**[1] option on the mouse menu, magnify the zone around surface 149.

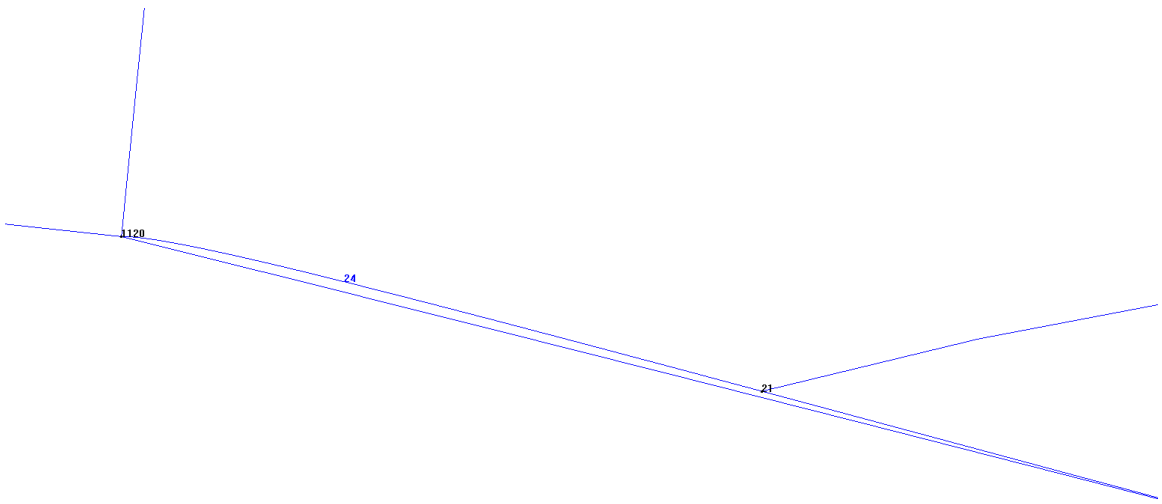


Figure 8. An enlargement of the zone arounnd surface 149.

2 . Several line segments are superimposed over each other, thus creating an incorrect surface boundary. Select **Geometry->Edit->Divide->Lines->Nearpoint** and then select point 21 (to select it, go to **Contextual** in the mouse menu, then select the option **Join C-a**). Point 21 is the point at which to make the cut.

3 . Then select line 3005. Press **ESC**. After the cut is made, the result will be as illustrated in Figure 10.

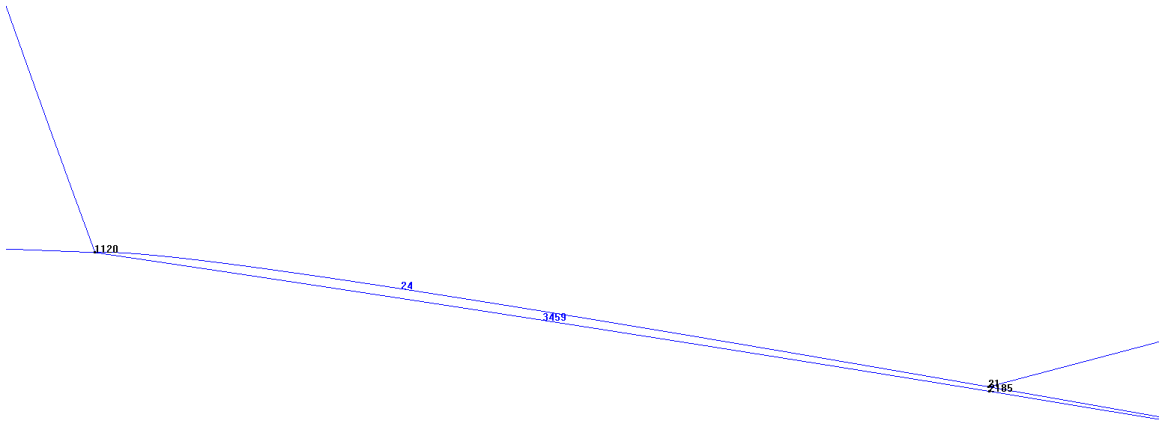


Figure 9. The zone after cutting line 3005 at point 21

4 . Now that the lines are precisely connected, a local collapse may be executed. Select **Geometry->Edit->Collapse->Lines**. Then select the lines that appear on the screen.

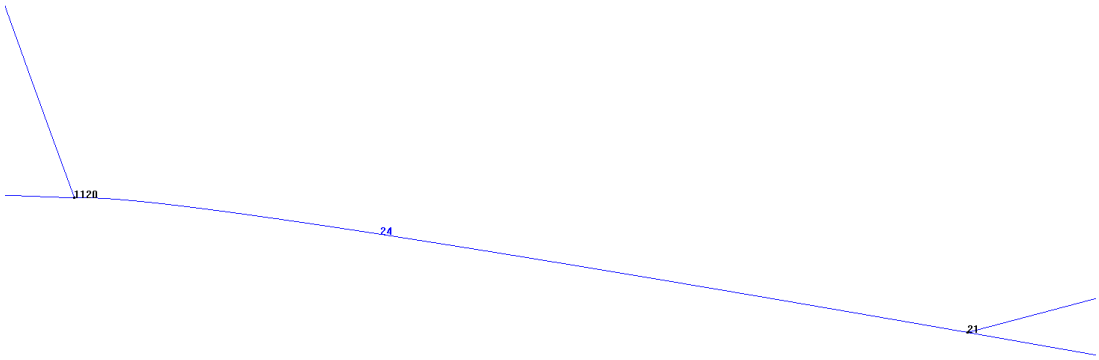


Figure 10. The situation after collapsing the lines

5 . After the collapse, the surface boundary is correct and the surface may be drawn with the new boundary. The labels are no longer needed, so click **Label->Off** in the contextual menu.

6 . Select **Geometry->Create->NURBS ->surfaceTrimmed**. Select surface 149. Then select the lines defining the recently repaired boundary. Press **ESC**.

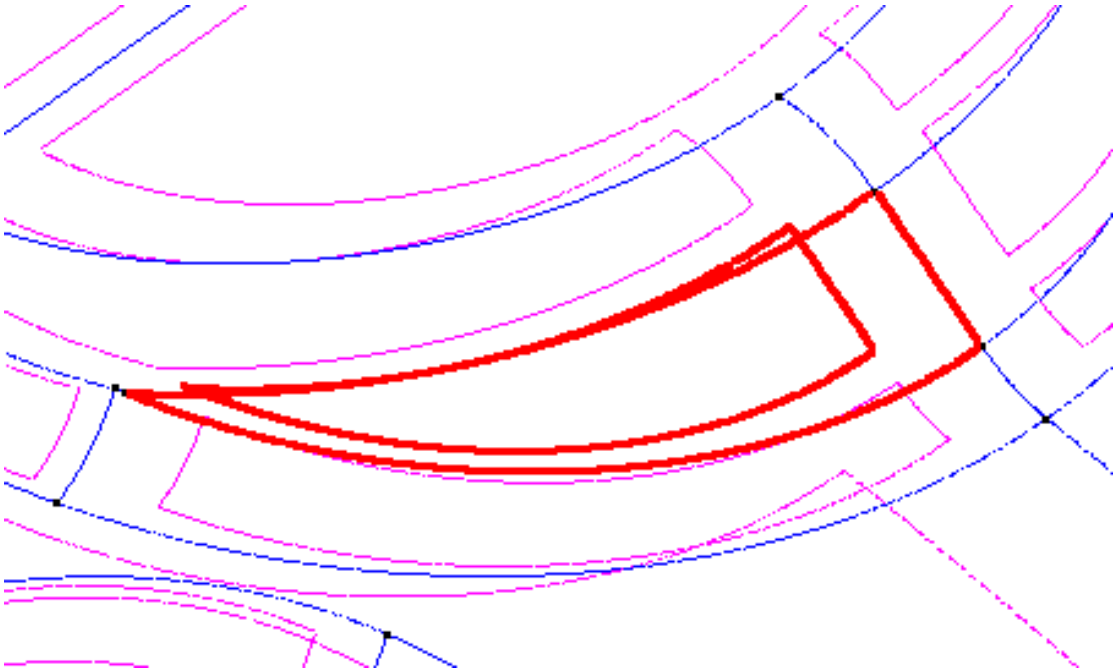


Figure 11. Surface 123 with its new boundary.

7 Select **Geometry->Delete->Surfaces**. Select surface 149 and press **ESC**.

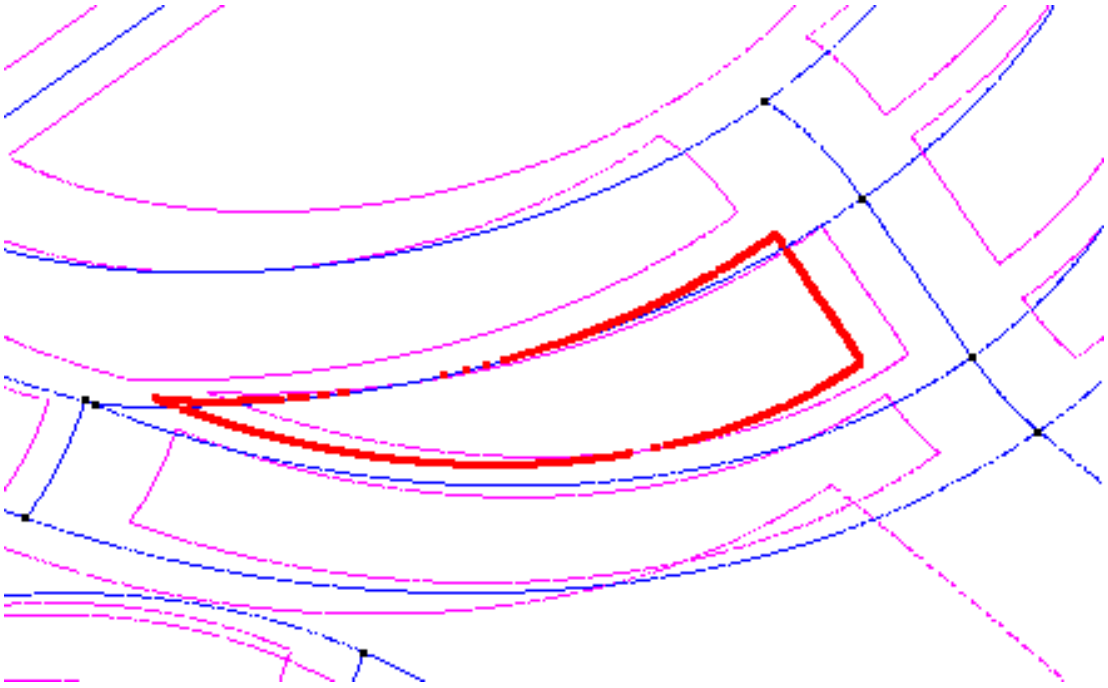


Figure 12. The surface to be eliminated.

8 To begin the second example in this section, mesh the geometry again with **Mesh->Generate Mesh**.

9 A window comes up in which to enter the maximum element size for the mesh to be generated. Leave the default value provided by GiD and click **OK**.

The mesh generating process may be carried out with no further errors found.

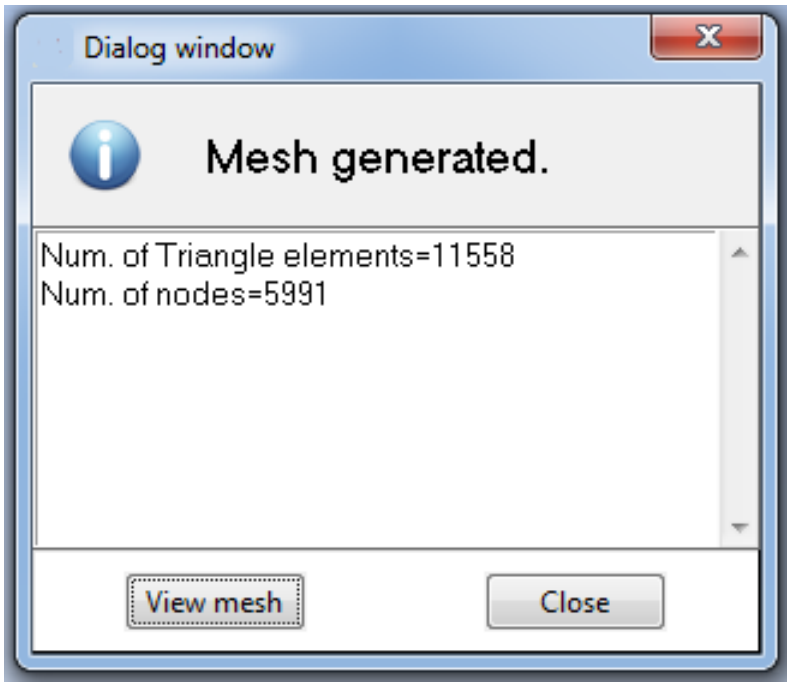


Figure 13. Window with information about the meshing process.

10 The imported piece is now meshed. (Figure 14)

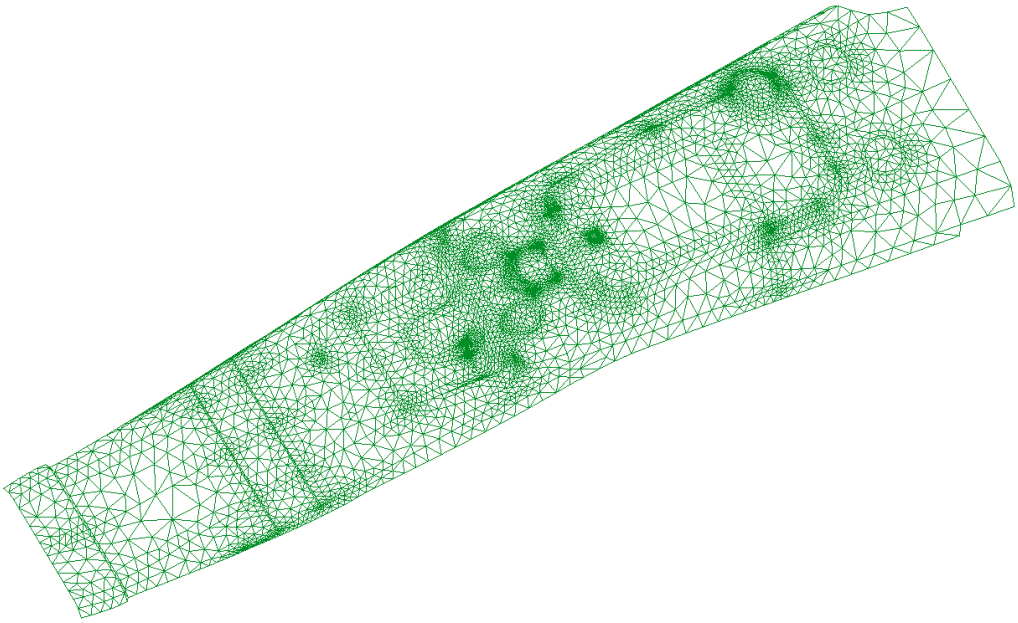


Figure 14. A mesh of the imported geometry.

---

[1] This option is also found in the GiD Toolbox.

8.3 The conformal mesh and the non-conformal mesh

In the previous section, after correcting some errors, we were able to mesh the imported geometry, thus obtaining a non-conformal mesh. A conformal mesh is one in which the elements share nodes and sides. To achieve this condition, contiguous surfaces (of the piece) must share lines and points of the

mesh. Most calculating modules require conformal meshes; however, some modules accept non-conformal meshes. A non-conformal mesh normally requires less computation time since it generates fewer elements.

8.3.1 Global collapse of the model

1 The option **Mesh->View mesh boundary** shows the boundary of all the surfaces of the conformal elements.
2 After generating the mesh, select **Mesh->View mesh boundary**. This will result in the image pictured in Figure 15.

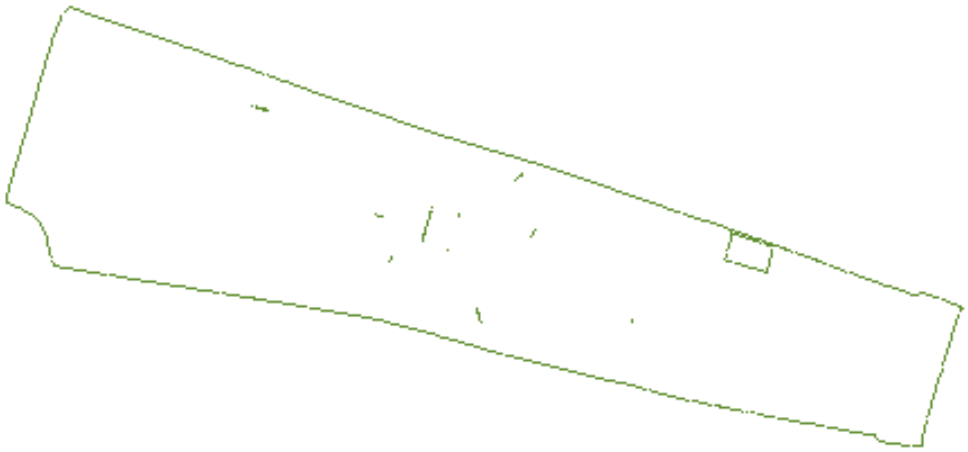


Figure 15. Visualization of the mesh generation using the option Mesh->View mesh boundary .

3 Visualization of the boundaries shows that in the interior of the piece some surfaces are isolated.
4 To generate a conformal mesh, first execute a global collapse of the model.
5 The GiD collapse depends upon the **Import tolerance**. Two entities are collapsed (converted into one) when they are separated by a distance less than the **Import tolerance** parameter. To test this, enter a new value for the **Import tolerance** parameter.
6 Go to **Utilities**, open **Preferences** , and bring up the **Exchange** card. Enter 0,15 for the **Import tolerance value**. Click **Accept**.

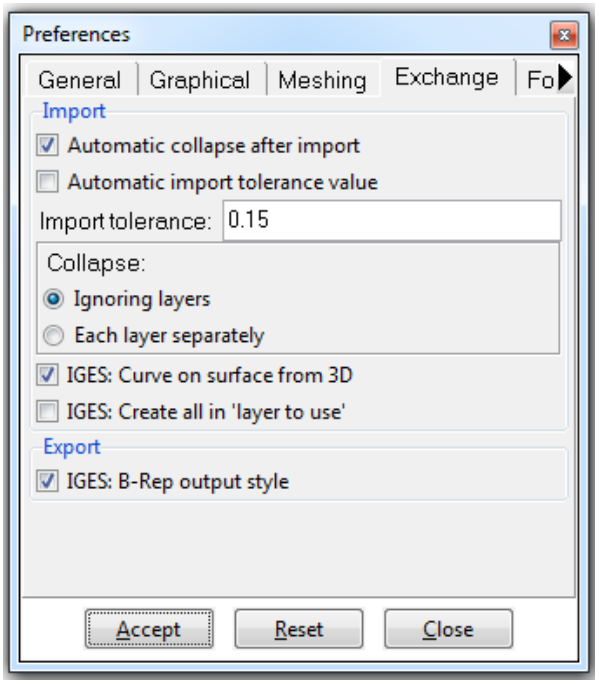


Figure 16. The Preferences window.

7 Select **Geometry->Edit->Collapse->Model**.

8 Select **Mesh->Generate**.

9 . When the **GiD** finishes the meshing process, an error message appears. As the window shows, there are errors meshing surface number 124.

10 Go to **Geometry->Delete->Surfaces** and delete surface 124.

11 . Now, it's necesary regenerate the surface. Go to **Geometry->Create->NURBS surface**.

12 . Select the lines of old surface 124 and press **ESC.**

13 . Select **Mesh->Generate**.

14 . Visualize the results with **Mesh->View mesh boundary**.

Some of the contiguous surfaces in the interior of the model have now being joined. However, there are still some surfaces that prevent the mesh from being completely conformal. These surfaces must be modified manually.

Figure 17. The mesh after the collapse.

8.3.2 Correcting surfaces and creating a conformal mesh

1 With the option **View->Zoom In**, magnify the zone illustrated in Figure 18.

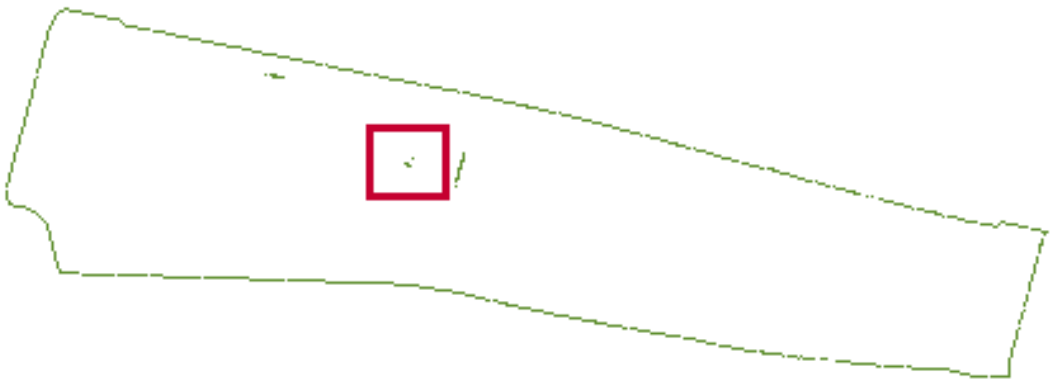


Figure 18. Zone in the mesh to zoom in.

Select **View->Mode->Geometry** to visualize the geometry of the piece.

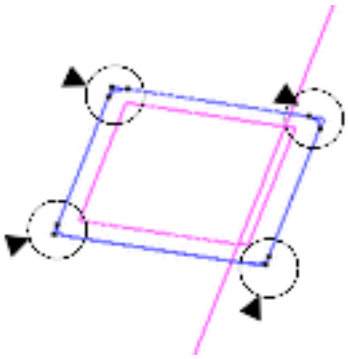


Figure 19

An image like that shown in Figure 19 appears. There is a rectangular surface that does not fit within the boundaries of a rounded-corner surface (a hole, in this case). We will suppose that the problematic surface is planar. This way, it can be erased and recreated in order to fit the rounded-corner boundary.

Select **Geometry->Delete->Surfaces**. Select the problematic surface and press **ESC**. Select **Geometry->Delete->Lines**. Select the lines forming the problematic surface and press **ESC**[2].

Use the option **Geometry->Delete->Points** to erase the points that do not belong to any surface.

With **Geometry->Create->NURBS surface->By contour**, create a new surface. The result is shown in Figure 20.

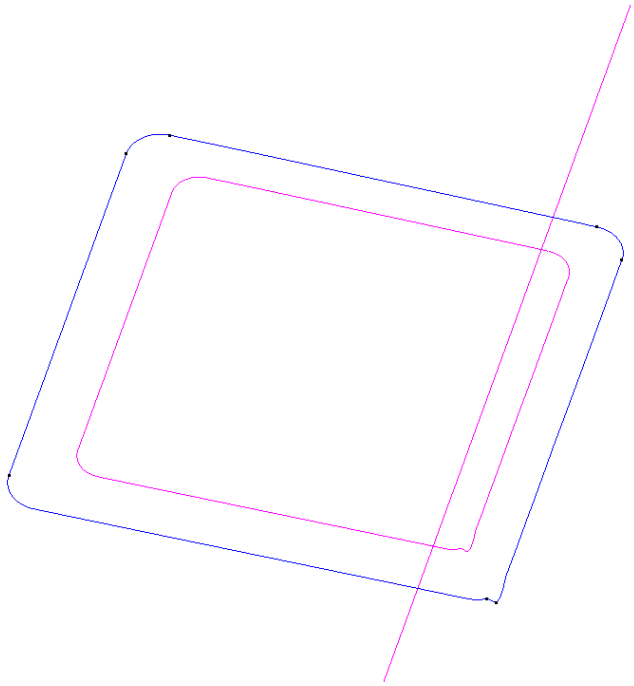


Figure 20

Visualize the mesh again using **Mesh->View mesh boundary** and magnify the zone indicated in Figure 21.

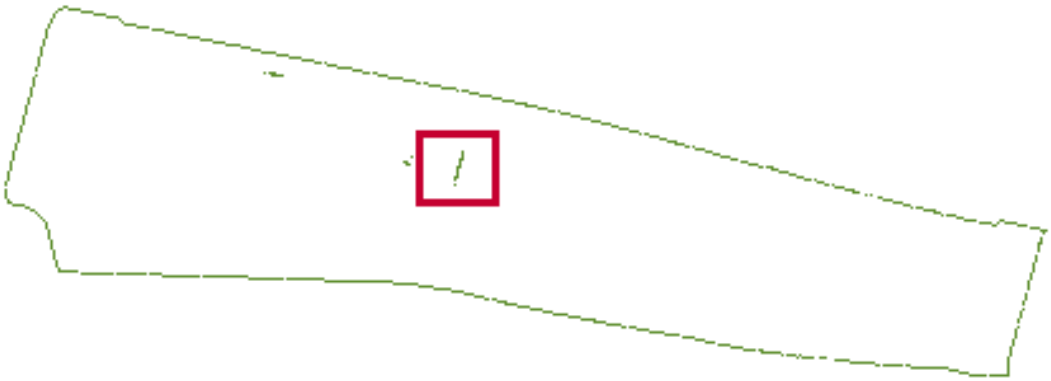


Figure 21

Select **View->Mode->Geometry**.

In this example, the situation involves a contour of four lines that does not correspond to any real surface (of the piece). These lines were too far apart to be collapsed (Figure 22).

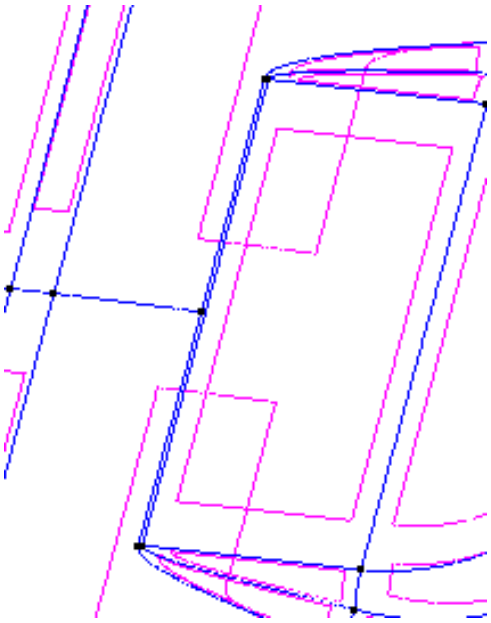


Figure 22

Select **Geometry->Create->NURBS surface->By contour**. Select the lines. Press **ESC**.

Magnify the zone indicated in Figure 23.

Figure 23

There are two surfaces that overlap each other at one end. (Figure 24)

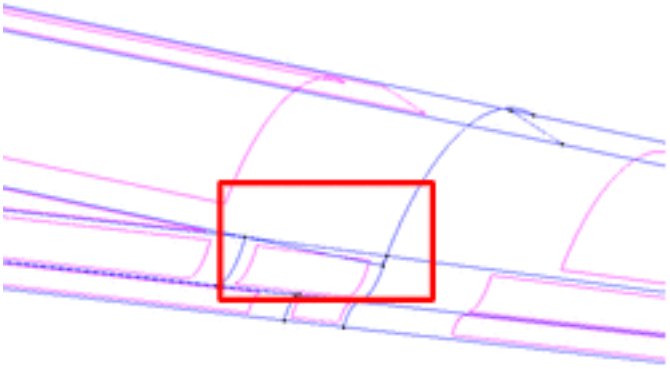


Figure 24. The magnified zone with two overlapping surfaces.

In this case the best solution for correcting the boundary is to trim the overlap. Select

Geometry->Create->NURBS Surface->Trimmed.

Select the surface to be trimmed. Then select the new boundary (Figure 25).

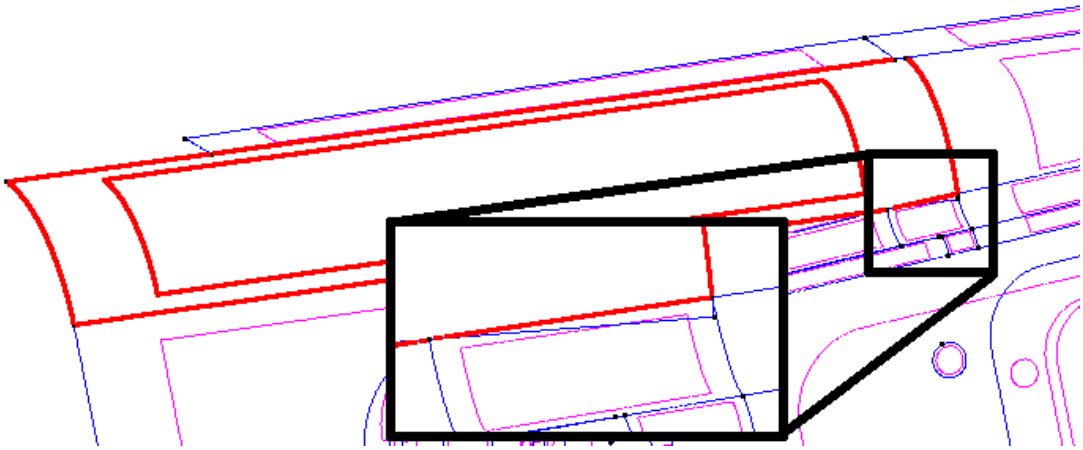


Figure 25. The surface to be trimmed and the new boundary.

Select **Geometry->Delete->Surfaces**. Select the original surface (Figure 26). Press **ESC**.

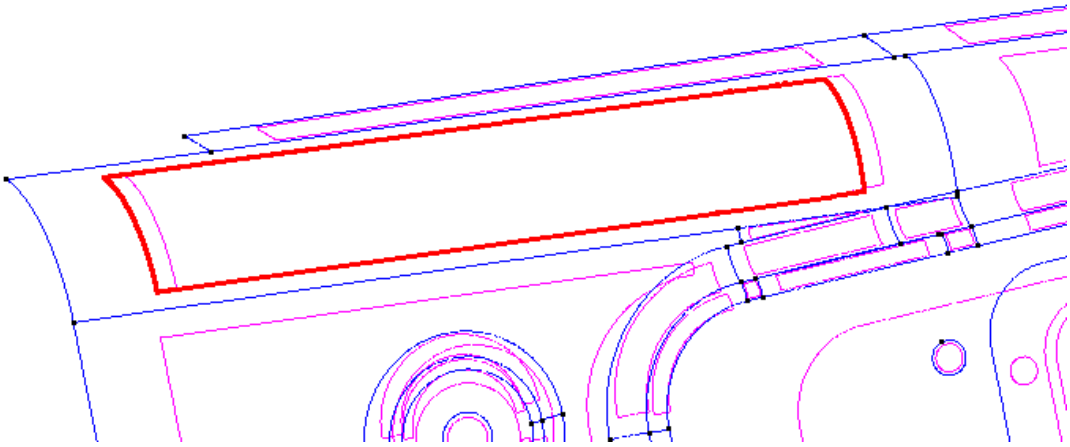


Figure 26. The original surface to be deleted.

Use **Geometry->Delete->Lines** and **Geometry->Delete->Points** to select the lines and points that belong to the surface that has been trimmed and which no longer belong to any surface (Figure 27).

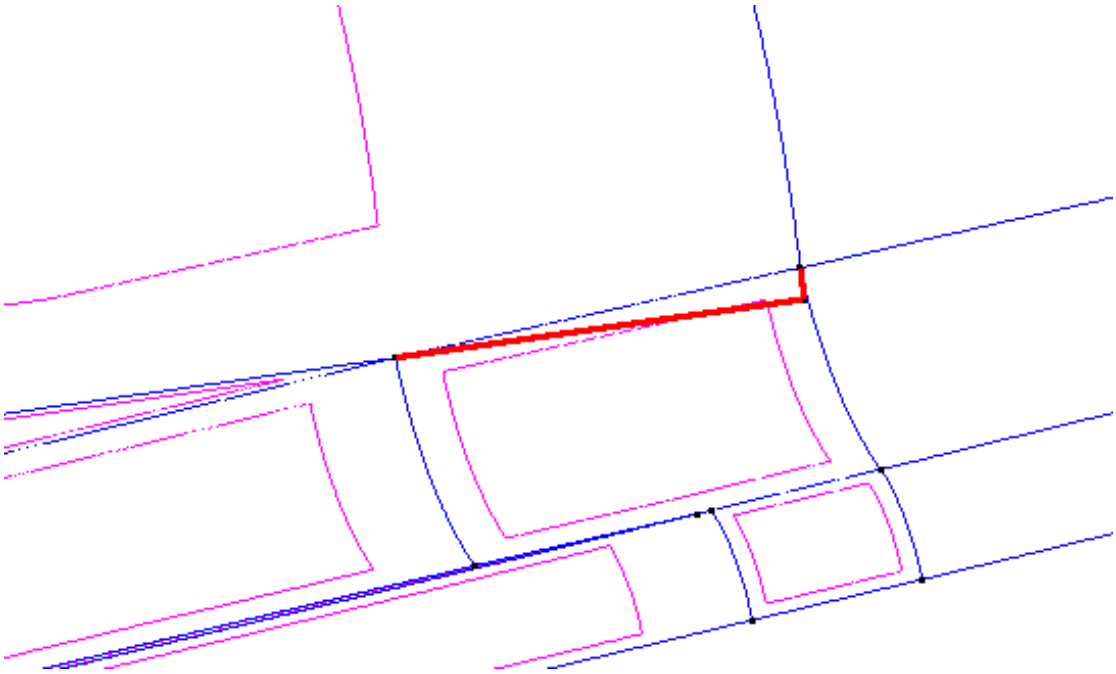


Figure 27. Lines and point that no longer belong to any surface.

Select **Mesh->Generate Mesh**. Then visualize the result using the option **Mesh->View mesh boundary**.

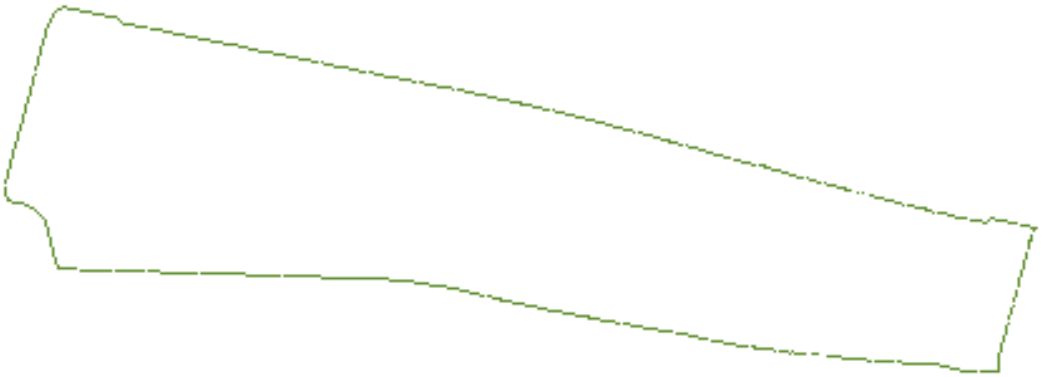


Figure 28. The mesh visualized with the option MeshView mesh boundary.

A conformal mesh has been achieved.

---

[2] In this case, all the visible lines may be selected since the program will only eliminate those which do not have entities covering them, that is, those which belong to the problematic vertices.

8.3.3 Creating a non-conformal mesh

NOTE: Non-conformal meshes may be used with some calculating modules, i.e. stamping a plate. Using non-conformal meshes significantly reduces the number of elements in the mesh. This cuts down on computation time.

1 Select **View->Mode->Geometry**.

2 Select **Geometry->Edit->Uncollapse->Surfaces**. Select all the surfaces in the model. Press **ESC**. A sufficient number of lines is created so that no surface (of the object) shares lines with any contiguous surface.

3 Select **Mesh->Generate Mesh**. When the mesh has been generated, a window appears with information about the mesh (Figure 29). The result is a non-conformal mesh composed of far fewer elements than the meshes generated in the previous section: about 5000 elements instead of the 20.000 needed to generate the conformal mesh.

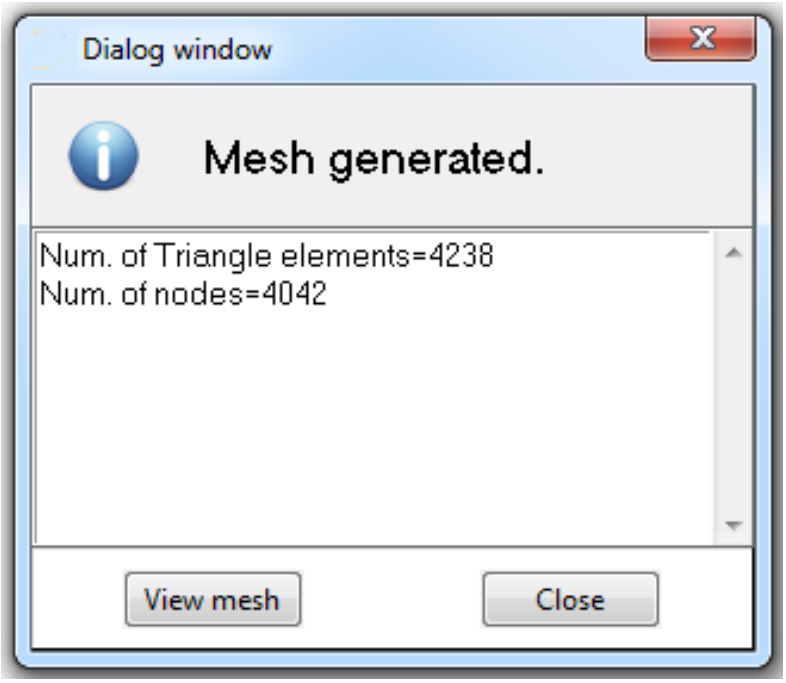


Figure 29. A window containing information about the generated mesh.

4 Visualize the result using **Mesh->View mesh boundary**.

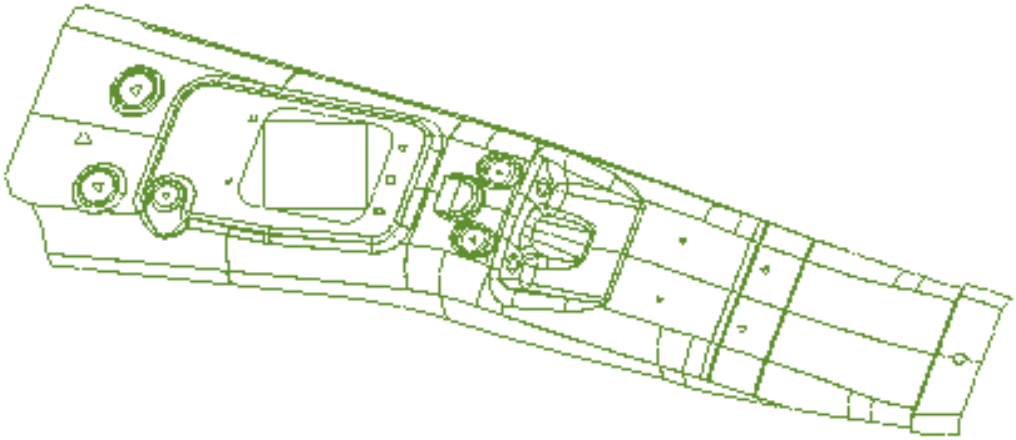


Figure 30. The mesh visualized using Mesh->View mesh boundary.

8.3.4 Optimizing a non-conformal mesh

NOTE: By using **Chordal Error**, the geometry may be discretized with great precision. The chordal error is the distance between the elements generated by the meshing program and the profile of the real object. Entering a sufficiently small chordal error results in small elements in zones where there is

greater curvature. Accordingly, the approximation of the mesh may be improved in zones with greater curvature by using the option "Chordal Error."

"Chordal Error" generates an increased number of elements in zones where there is curvature. One way of obtaining accurate meshes with few elements is using structured elements in zones where there is curvature. The option **Allow automatic structured**, located in **Preferences**, may be combined with the option of limiting the chordal error, thus achieving an accurate mesh with fewer elements. It only makes sense to use **Allow automatic structured** when working with a non-conformal mesh.

NOTE: The option **Allow automatic structured** generates highly distorted elements that might, with some calculating modules, lead to erroneous results. In the case of stamping a plate, we recommend using **Allow automatic structured** with the calculating modules.

1 Open **Utilities->Preferences**.
2 On the **Meshing** card, activate the option **Allow automatic structured** and enter the value of 0,9 in the box labeled **Unstructured size transitions**. Click **Accept**. The option **Unstructured size transitions** defines the size gradient of the elements (the value ranging from 0 to 1). The greater the value, the faster the variation of the element sizes in space and so there will befewer elements in the mesh.

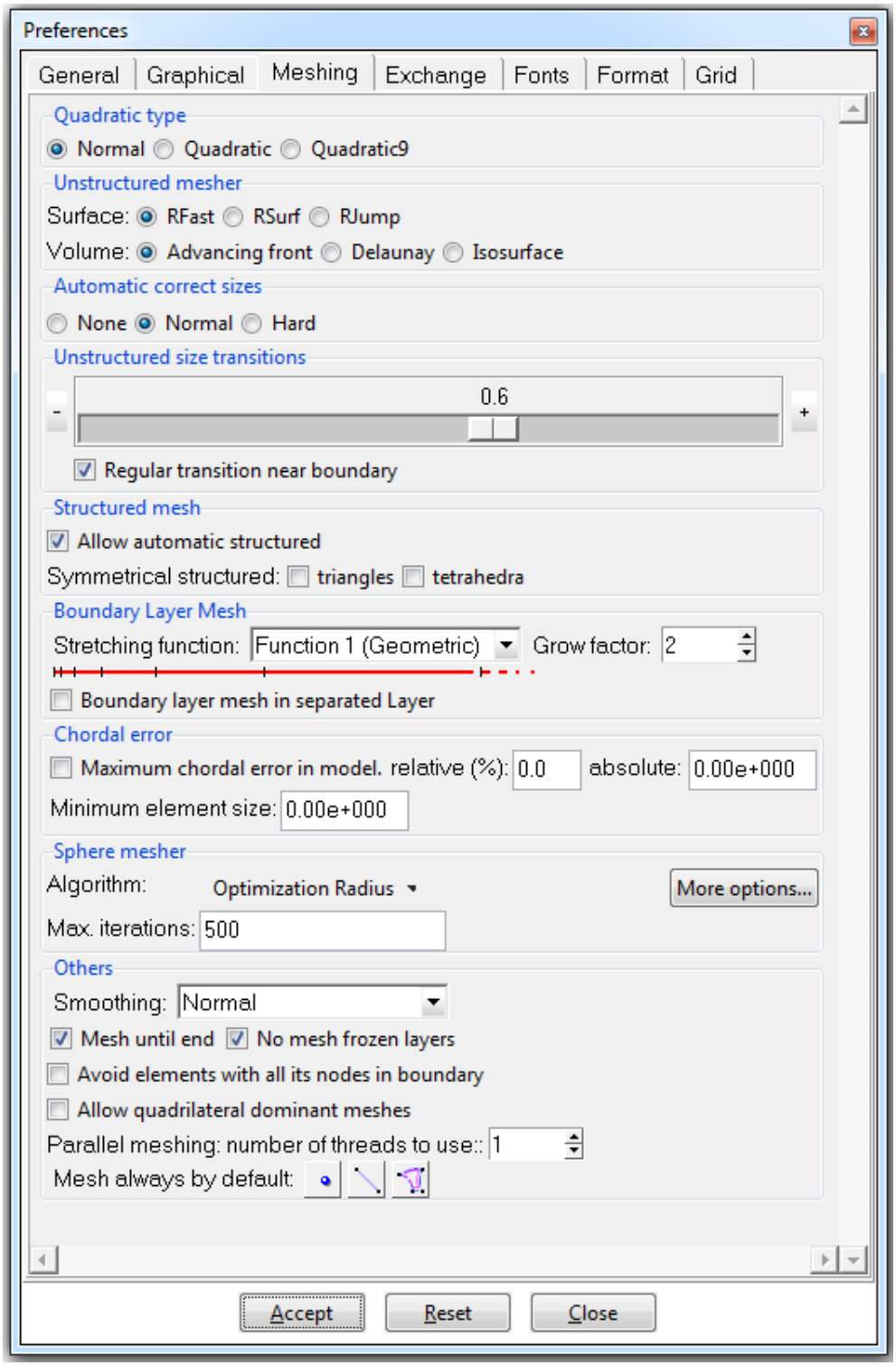


Figure 31. The Preferences window.

3 Select **Mesh->Unstructured->Sizes by Chordal error** and set the values as shown in Figure 32.

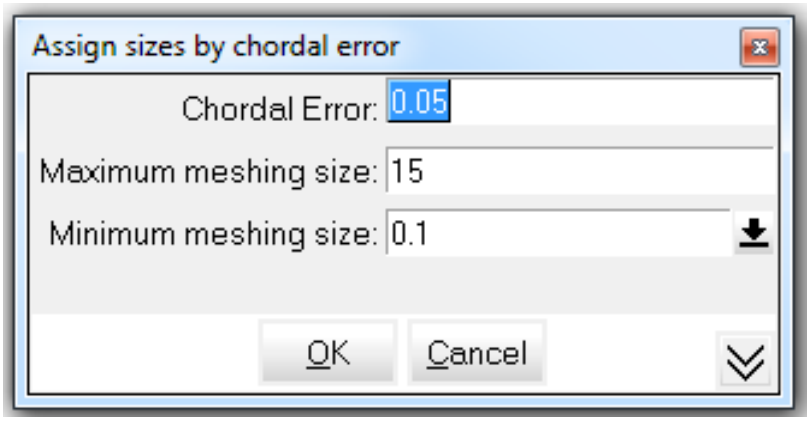


Figure 32. Defining unstructured size by chordal error.

4 Select **Mesh->Generate Mesh**.

5 A window comes up in which to enter the maximum element size. Leave the default value unaltered and click **OK**.

6 Once the process of generating the mesh is finished, a window appears with information about the generated mesh. Click **OK** to visualize the mesh.

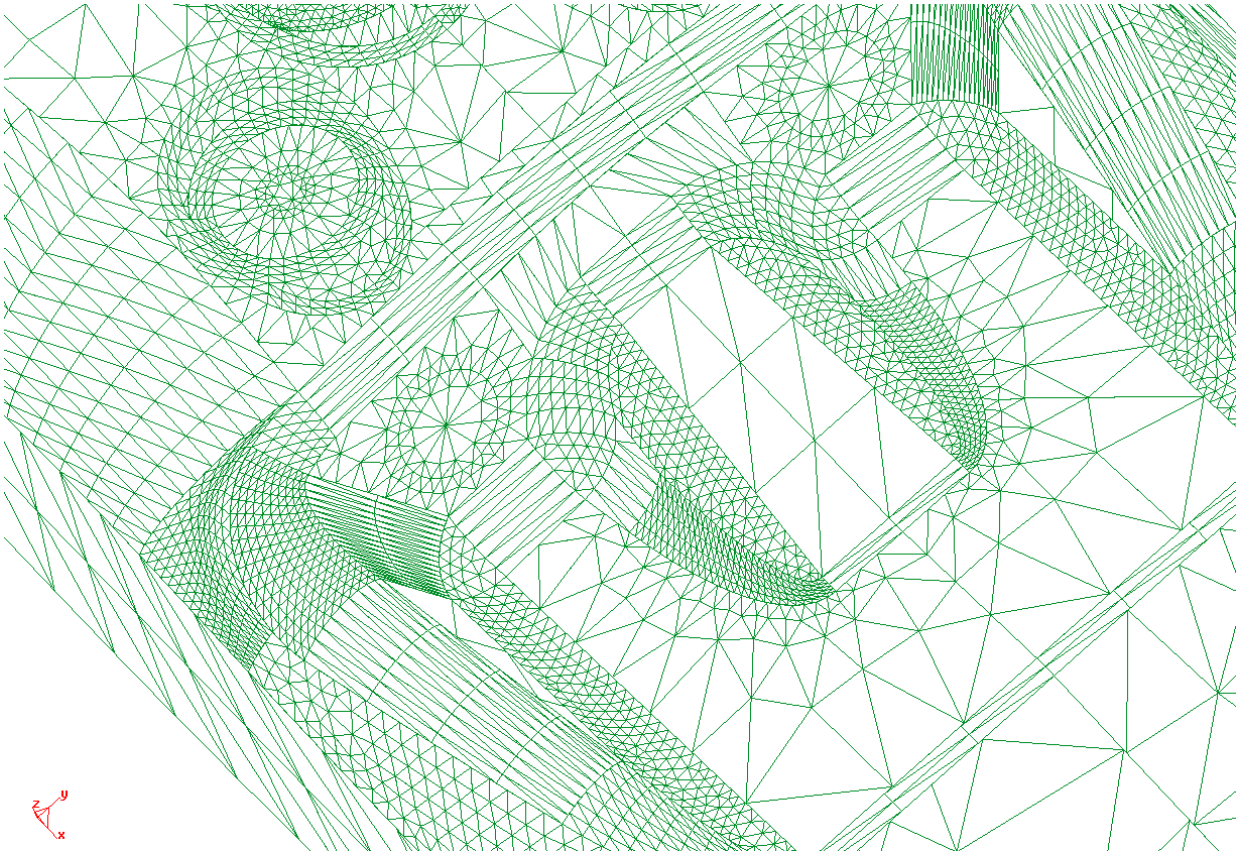


Figure 33. An optimized non-conformal mesh. Structured elements are present on the curved surfaces.

9 DEFINING A PROBLEM TYPE

This tutorial takes you through the steps involved in defining a problem type using GiD. A problem type is a set of files configured by a solver developer so that the program can prepare data to be analyzed.

A simple example has been chosen which takes us through all the associated configuration files while using few lines of code. Particular emphasis is given to the calculation of the centers of mass for two-dimensional surfaces ? a simple formulation both conceptually and numerically.

The tutorial is composed of the following steps:

Starting the 'problemtype'

Creating the materials definition file

Creating the general configurations file

Creating the conditions definition file

Creating the data format file

Creating the calculating program file and the execution files

Executing the calculating module and visualizing the results using GiD

By the end of the example, you should be able to create a calculating module that will interpret the mesh generated in GiD Preprocess. The module will calculate values for each element of the mesh and store the values in a file in such a way as they can be read by GiD Post-process.

9.1 Introduction

Our aim is to solve a problem that involves calculating the center of gravity (center of mass) of a 2D object. To do this, we need to develop a calculating module that can interact with GiD.

The problem: calculate the center of mass.

The center of mass (X_{CM},Y_{CM}) of a two-dimensional body is defined as

$$x_{CM} = \frac{\int_S \rho \; x \; x \; y + \sum_{i=1}^{N} m_i \; x_i}{\int_S \rho \; x \; y + \sum_{i=1}^{N} m_i} \qquad y_{CM} = \frac{\int_S \rho \; y \; x \; y + \sum_{i=1}^{N} m_i \; y_i}{\int_S \rho \; x \; y + \sum_{i=1}^{N} m_i}$$

where ρ (x,y) is the density of the material at point (x,y) and S is the surface of the body; mi are concentrated masses applied on the point (x_i,y_i).

To solve the problem numerically, the integrals will be transformed into sums:

$$x_{CM} = \frac{\sum_{elm} \rho_{elm} V_{elm} \; x_{elm} + \sum_{i=1}^{N} m_i \; x_i}{\sum_{elm} \rho_{elm} V_{elm} + \sum_{i=1}^{N} m_i} \qquad y_{CM} = \frac{\sum_{elm} \rho_{elm} V_{elm} \; y_{elm} + \sum_{i=1}^{N} m_i \; y_i}{\sum_{elm} \rho_{elm} V_{elm} + \sum_{i=1}^{N} m_i}$$

Each of the N elements is treated as concentrated weight whose mass is defined as the product of the (surface) density and the area of the element.

9.1.1 Interaction of GiD with the calculating module

GiD Preprocess makes a discretization of the object under study and generates a mesh of elements, each one of which is assigned a material and some conditions. This preprocessing information in GiD (mesh, materials, and conditions) enables the calculating module to generate results. For the present example, the calculating module will find the distance of each element relative to the center of mass of the object.

Finally, the results generated by the calculating module will be read and visualized in GiD Post-process.

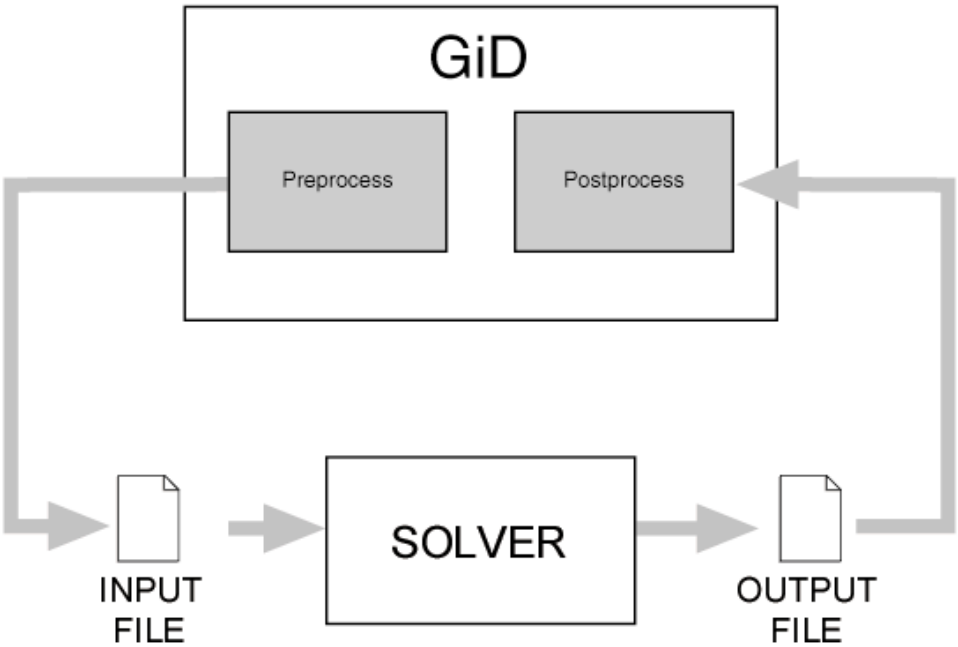


Diagram of the workflow

GiD must adapt these data to deal with them. Materials, boundary and/or load conditions, and general problem data must be defined.

GiD configuration is accomplished through text formatted files. The following files are required:

.prb: configuration of the general parameter (not associated to entities)

.mat: configuration of materials and their properties

.cnd: configuration of the conditions imposed on the calculation

.bas: (template file) the file for configuring the format of the interchange file that mediates between GiD data and the calculating module. The file for interchanging the data exported by GiD has the extension .dat. This file stores the geometric and physical data of the problem.

.bat: the file that can be executed called from GiD. This file initiates the calculating module.

The calculating module (in this example cmas2d.exe) solves the equations in the problem and saves the results in the results file. This module may be programmed in the language of your choice, 'C' in used in this example

GiD Post-process reads the following files generated by the calculating module:

project_name.post.res: results file.

Each element of the mesh corresponds to a value.

project_name.post.msh: file containing the post-process mesh. If this file does not exist, GiD uses the preprocess mesh also for postprocess.

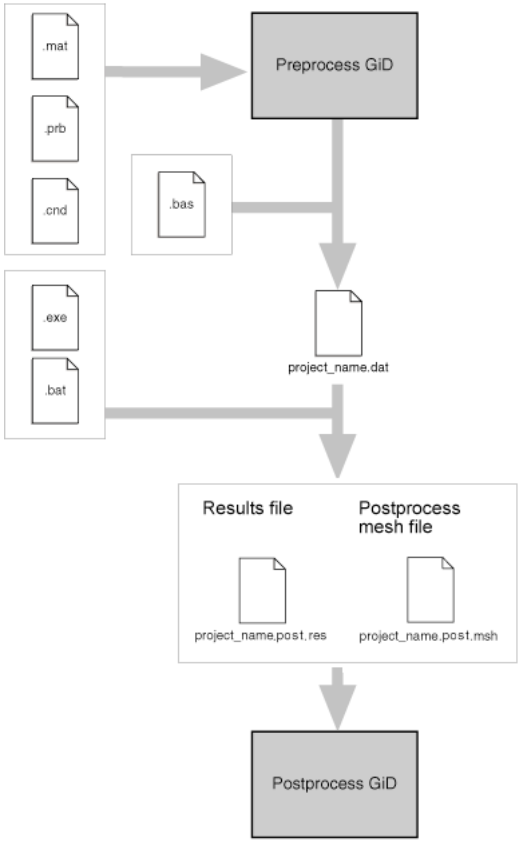


Diagram depicting the files system

9.2 Implementation

Creating the Subdirectory for the Problem Type

Create the subdirectory "cmas2d.gid". This subdirectory has a .gid extension and will contain all the configuration files and calculating module files (.prb, .mat, .cnd, .bas, .bat, .exe).

NOTE: If you want the problem type to appear in the GiD Data?Problem type menu, create the subdirectory within "problemtypes", located in the GiD folder ? for instance, C:\GiD\Problemtypes\cmas2d.gid

9.2.1 Creating the Materials File

Create the materials file "cmas2d.mat". This file stores the physical properties of the material under study for the problem type. In this case, defining the density will be enough.

Enter the materials in the "cmas2d.mat" file using the following format:

MATERIAL: Name of the material (without spaces)

QUESTION: Property of the material. For this example, we are interested in the density of the material.

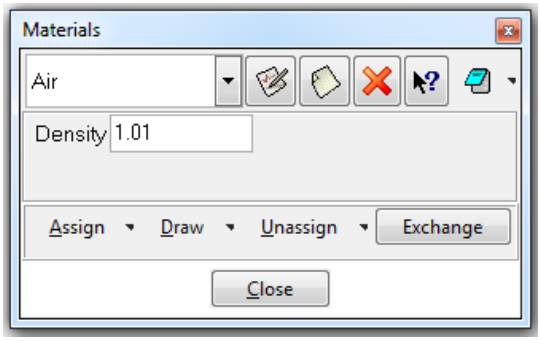
VALUE: Value of the property

HELP: A help text (optional field)

…

END MATERIAL

In GiD, the information in "cmas2d.mat" file is managed in the materials window, located in **Data->Materials**.



The GiD Materials window, for assigning materials

MATERIAL: Air

QUESTION: Density

VALUE: 1.01

HELP: material density

END MATERIAL

MATERIAL: Steel

QUESTION: Density

VALUE: 7850

HELP: material density

END MATERIAL

MATERIAL: Aluminium

QUESTION: Density

VALUE: 2650

HELP: material density

END MATERIAL

9.2.2 Creating the General File

Create the "cmas2d.prb" file. This file contains general information for the calculating module, such as the units system for the problem, or the type of resolution algorithm chosen.

Enter the parameters of the general conditions in "cmas2d.prb" using the following format:

PROBLEM DATA

QUESTION: Name of the parameter. If the name is followed by the #CB# instruction, the parameter is displayed as a combo box. The options in the menu must then be entered between parentheses and separated by commas.

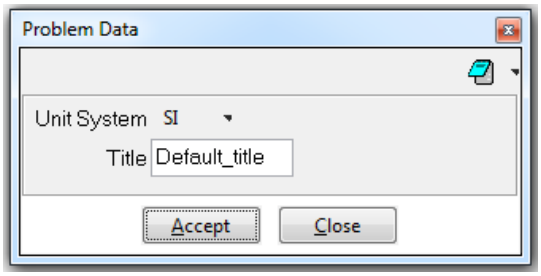
For example, Unit_System#CB#(SI,CGS,User).

VALUE: The default value of the parameter.

…

END GENERAL DATA

In GiD, the information in the "cmas2d.prb" file is managed in the problem data window, which is located in **Data?Problem Data**.



The GiD Problem Data window, for configuring of the general conditions of the cmas2d module

```
PROBLEM DATA
QUESTION: Unit_System#CB#(SI,CGS,User)
VALUE: SI
QUESTION: Title
VALUE: Default_title
END GENERAL DATA
```

9.2.3 Creating the Conditions File

Create the "cmas2d.cnd" file, which specifies the boundary and/or load conditions of the problem type in question. In the present case, this file is where the concentrated weights on specific points of the geometry are indicated.

Enter the boundary conditions using the following format:

CONDITION: Name of the condition

CONDTYPE: Type of entity which the condition is to be applied to. This includes the parameters "over points", "over lines", "over surfaces", "over volumes" or "over layers". In this example the condition is applied "over points".

CONDMESHTYPE: Type of entity of the mesh where the condition is to be applied. The possible parameters are "over nodes", "over body elements" or "over face elements". In this example, the condition is applied on nodes.

QUESTION: Name of the parameter of the condition

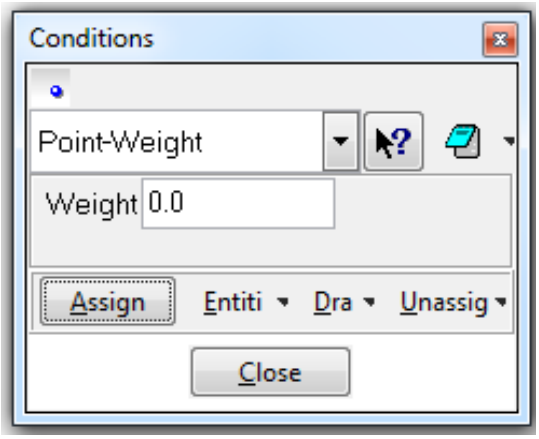
VALUE: Default value of the parameter

…

END CONDITION

…

In GiD, the information in the "cmas2d.cnd" file is managed in the conditions window, which is found in **Data? Conditions**.



The GiD Conditions window, for assigning the cmas2d boundary and load conditions

```
CONDITION: Point-Weight
CONDTYPE: over points
CONDMESHTYPE: over nodes
QUESTION: Weight
VALUE: 0.0
HELP: Concentrated mass
END CONDITION
```

9.2.4 Creating the Data Format File (Template file)

Create the "cmas2d.bas" file. This file will define the format of the .dat text file created by GiD. It will store the geometric and physical data of the problem. The .dat file will be the input to the calculating module.

NOTE: It is not necessary to have all the information registered in only one .bas file. Each .bas file has a corresponding .dat file.

Write the "cmas2d.bas" file as follows:

The format of the .bas file is based on commands. Text not preceded by an asterisk is reproduced exactly the same in the .dat file created by GiD. A text preceded by an asterisk is interpreted as a command.

Example:

.bas file .dat file

```
%%%%        ?   %%%% Problem Size
Problem         %%%%
Size %%%%
                Number of Elements &
Number of       Nodes:
Elements &      5379 4678
Nodes:
*nelem
*npoin
```

The contents of the "cmas2d.bas" file must be the following:

.bas file

```
=================================================================
General Data File
=================================================================
Title: *GenData(Title)
%%%%%%%%%%%%%%%%% Problem Size %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Number of Elements & Nodes:
*nelem *npoin
```

In this first part of "cmas2d.bas" file, general information on the project is obtained.

***nelem:** returns the total number of elements of the mesh.

***npoin:** returns the total number of nodes of the mesh.

```
Coordinates:
Node X Y
*loop nodes
*format "%5i%14.5e%14.5e"
*NodesNum *NodesCoord(1,real) *NodesCoord(2,real)
*end nodes
```

This command provides a rundown of all the nodes of the mesh, listing their identifiers and coordinates.

***loop, *end:** commands used to indicate the beginning and the end of the loop. The command ***loop** receives a parameter.

***loop nodes:** the loop iterates on nodes

***loop elems:** the loop iterates on elements

***loop materials:** the loop iterates on assigned materials

***format:** the command to define the printing format. This command must be followed by a numerical format expressed in C syntax.

***NodesNum:** returns the identifier of the present node

***NodesCoord:** returns the coordinates of the present node

***NodesCoord (n, real):** returns the x, y or z coordinate in terms of the value n:

n=1 returns the **x** coordinate

n=2 returns the **y** coordinate

n=3 returns the **z** coordinate

```
Connectivities:
Element Node(1) Node(2) Node(3) Material
*set elems(all)
*loop elems
*format "%10i%10i%10i%10i%10i"
*ElemsNum *ElemsConec *ElemsMat
*end elems
```

This provides a rundown of all the elements of the mesh and a list of their identifiers, the nodes that form them, and their assigned material.

***set elems(all):** the command to include all element types of the mesh when making the loop.

***ElemsNum:** returns the identifier of the present element

***ElemsConec:** returns the nodes of an element in a counterclockwise order

***ElemsMat:** returns the number of the assigned material of the present element

```
Begin Materials
Nº Materials= *nmats
```

This gives the total number of materials in the project

***nmats:** returns the total number of materials

```
Mat. Density

*loop materials
*format "%4i%13.5e"
*set var PROP1(real)=Operation(MatProp(Density, real))
*MatNum *PROP1
*end
```

This provides a rundown of all the materials in the project and a list of the identifiers and densities for each one.

***MatProp (density, real):** returns the value of the property "density" of the material in a "real" format.

***Operation (expression):** returns the result of an arithmetic expression. This operation must be expressed in C.

***Set var PROP1(real)=Operation(MatProp(Density, real)):** assigns the value returned by MatProp (which is the value of the density of the material) to the variable PROP1 (a "real" variable).

***PROP1:** returns the value of the variable PROP1.

***MatNum:** returns the identifier of the present material.

```
Point conditions

*Set Cond Point-Weight *nodes

*set var NFIX(int)=CondNumEntities(int)

Concentrate Weights

*NFIX
```

This provides the number of entities with a particular condition.

***Set Cond Point-Weight *nodes:** this command enables you to select the condition to work with from that moment on. For the present example, select the condition "Point-Weight".

***CondNumEntities(int):** returns the number of entities with a certain condition.

***Set var NFIX(int)= CondNumEntities(int):** assigns the value returned by the command CondNumEntities to the NFIX variable (an "int" variable).

***NFIX:** returns the value of the NFIX variable.

```
Potentials Prescrits:

Node Tipus

Valor/Etiqueta

*loop nodes *OnlyInCond

*NodesNum *cond(1)

*end
```

This provides a rundown of all the nodes with the condition "Point-Weight" with a list of their identifiers and the first "weight" field of the condition in each case.

***loop nodes *OnlyInCond:** executes a loop that will provide a rundown of only the nodes with this condition.

***cond(1):** returns the number 1 field of a condition previously selected with the *set cond command. The field of the condition may also be selected using the name of the condition, for example cond(weight).

cmas2d.bas

==========

General Data File

==========

%%%%%%%%%%%%%%%%%%% Problem Size %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Number of Elements & Nodes:

*nelem *npoin

%%%%%%%%%%%%%%%%%%%% Mesh Database %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Coordinates:

Node X Y

*set elems(all)

*loop nodes

*format "%5i%14.5e%14.5e"

*NodesNum *NodesCoord(1,real) *NodesCoord(2,real)

*end nodes

.....

Connectivities:

Element Node(1) Node(2) Node(3) Material

*loop elems

*format "%10i%10i%10i%10i%10i"

*ElemsNum *ElemsConec *ElemsMat

*end elems

.....

Begin Materials

Nº Materials= *nmats

Mat. Density

.....

*loop materials

*format "%4i%13.5e"

*set var PROP1(real)=Operation(MatProp(Density, real))

*MatNum *PROP1

*end

.....

Point conditions

*Set Cond Point-Weight *nodes

*set var NFIX(int)=CondNumEntities(int)

Concentrated Weights

*NFIX

.....

Potentials Prescrits:

Node Tipus

Valor/Etiqueta

*Set Cond Point-Weight *nodes

*loop nodes *OnlyInCond

*NodesNum *cond(1)

*end

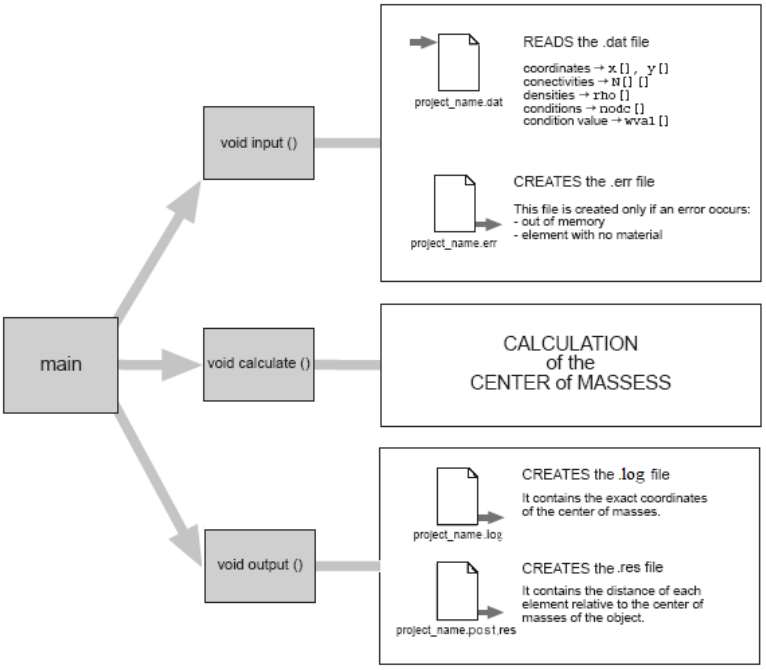
.....

9.2.5 Creating the Execution file of the Calculating Module

Create the file "cmas2d.c". This file contains the code for the execution program of the calculating module. This execution program reads the problem data provided by GiD, calculates the coordinates of the center of mass of the object and the distance between each element and this point. These results are saved in a text file with the extension .post.res.

Compile and link the "cmas2d.c" file in order to obtain the executable cmas2d.exe file.

The calculating module (cmas2d.exe) reads and generates the files described below.



cmas2d.c solver structure

NOTE: The "cmas2d.c" code is explained in the appendix.

9.2.6 Creating the Execution File for the Problem Type

Create the "cmas2d.win.bat" file. This file connects the data file(s) (.dat) to the calculating module (the cmas2d.exe program). When the GiD Calculate option is selected, it executes the .bat file for the problem type selected.

When GiD executes the .bat file, it transfers three parameters in the following way:

(parameter 3) / *.bat (parameter 2) / (parameter 1)

parameter 1: project name

parameter 2: project directory

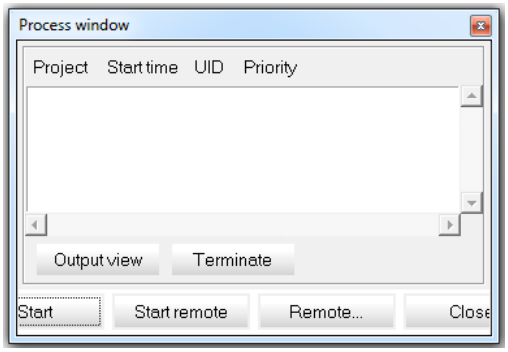
parameter 3: Problem type location directory

NOTE: The .win.bat fiile as used in Windows is explained below; the shell script for UNIX systems is also included with the documentation of this tutorial.

```
rem OutputFile: %2\%1.log
```

A comment line such as "rem OutputFile: file_name.log" means that the contents of the file indicated will be shown if the user clicks Output View in **Calculate?Calculate window**.

In this example the .log file is shown. This file contains the coordinates of the center of mass.



The Process window.

```
rem ErrorFile: %2\%1.err
```

A comment line such as "rem ErrorFile: file_name.err" means that the indicated file will contain the errors (if any). If the .err file is present at the end of the execution, a window comes up showing the error. The absence of the .err file indicates that the calculation is considered satisfactory.

GiD automatically deletes the .err files before initiating a calculation to avoid confusion.

```
del %2\%1.log
del %2\%1.post.res
```

This deletes results files from any previous calculations to avoid confusion.

```
%3\cmas2d.exe %2\%1
```

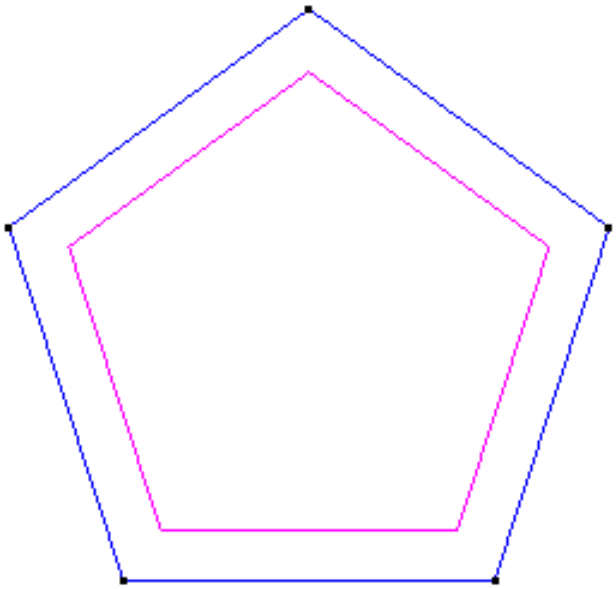
This executing the cmas2d.exe and provide the .dat as input file file.

9.3 Using the problemtype with an example

In order to understand the way the calculating module works, simple problems with limited practical use have been chosen. Although these problems do not exemplify the full potential of the GiD program, the user may intuit their answers and, therefore, compare the predicted results with those obtained in the simulations.

Create a surface, for example from the menu **Geometry->Create->Object->Polygon**

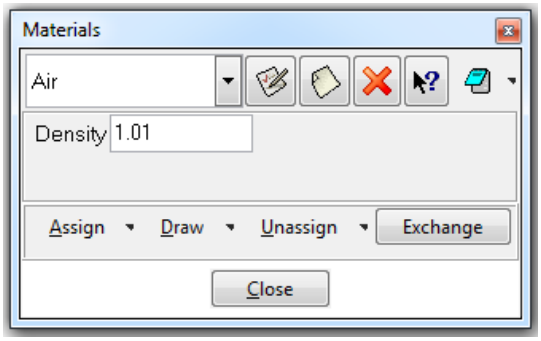
Create a polygon with 5 sides, centered in the (0,0,0) and located in the XY plane (normal = 0,0,1) and whit radius=1.0



Surface used for this example

Load the problemtype: menu **Data?Problem type?cmas2d**.

Choose **Data?Materials**. The materials window is opened. From the Materials menu in this window, choose the option Air.

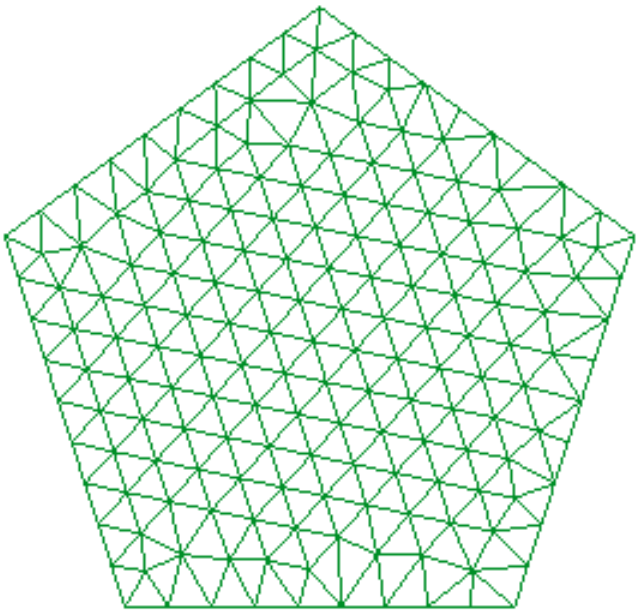


Materials window

Click **Assign?Surfaces** and select the surface. Press ESC when this step is finished.

Choose the **Mesh?Generate** option.

A window appears in which to enter the maximum element size for the mesh to be generated. Accept the default value and click OK. The mesh shown will be obtained.

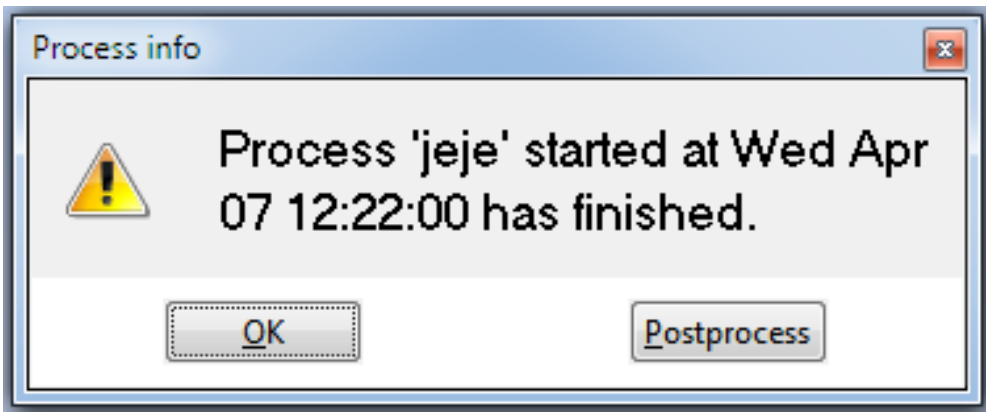


The mesh of the object

Now the calculation may be initiated, but first the model must be saved (**Files->Save**), use 'example_cmas2d' as name for the model.

Choose the Calculate option from the Calculate menu to start the calculation module.

Wait until a box appears indicating the calculation has finished.

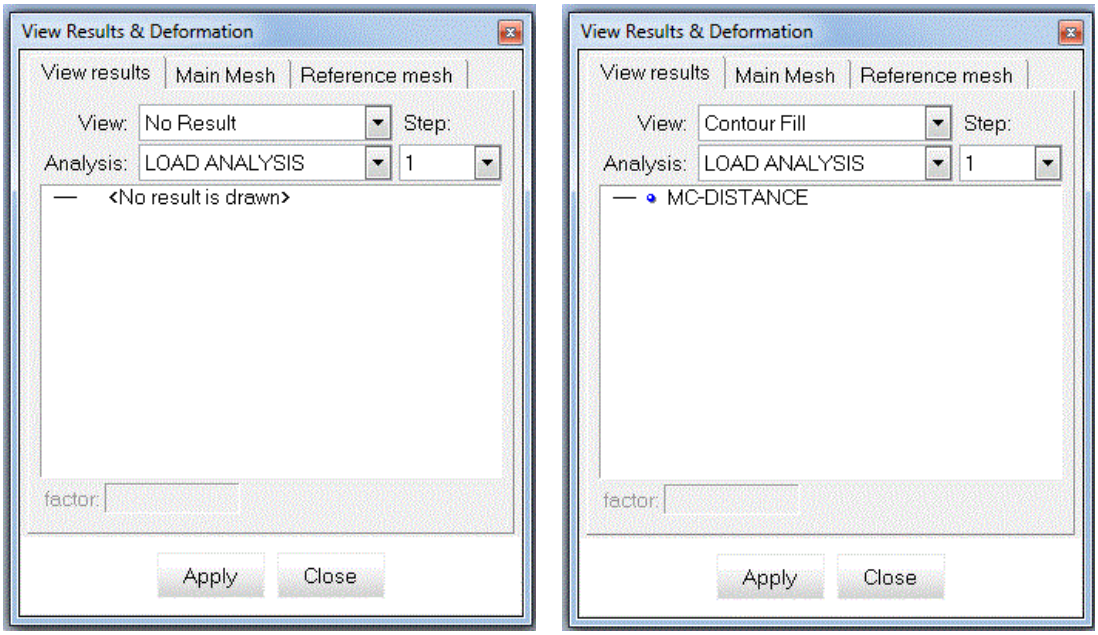


Process information box

Choose the option **Files?Postprocess**.

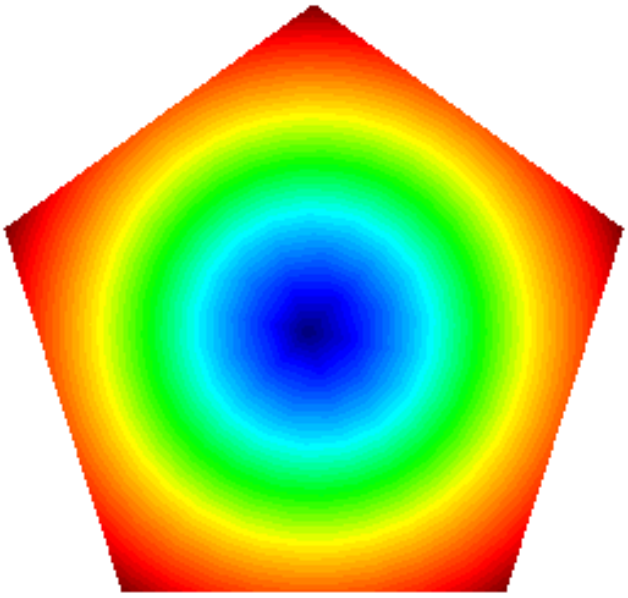
From the Windows menu, choose the 'View results' option. A window appears from which to visualize the results. By default when changing to postprocesses mode no results is visualized.

From the View combo box in the View Results window, choose the Contour Fill option. A set of available results (only one for this case) are displayed.



The View Results window.

Now choose the MC-DISTANCE result and click Apply. A graphic representation of the calculation is obtained.



Visualizing the distance (MC-DISTANCE) from the center of mass of the objectto each element, for an object of homogeneous material

The results shown on the screen reproduce those we anticipated at the outset of the problem: the center of mass of an object made of homogeneous material coincides with its geometric center. The .log file will provide the exact coordinates of this point.

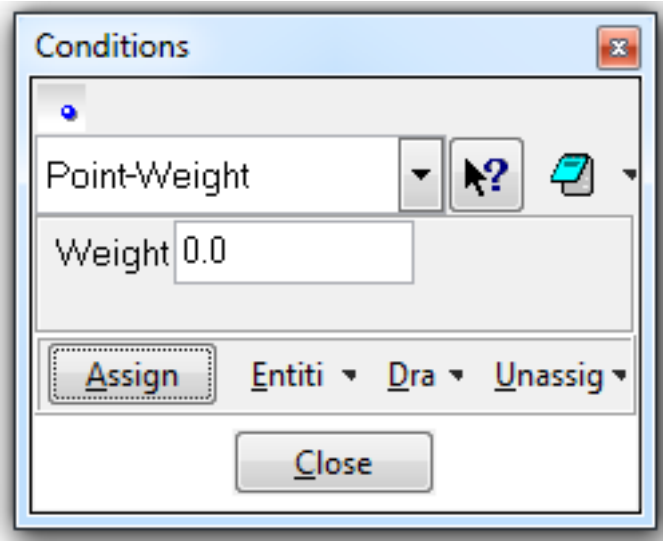
9.3.1 Executing the calculation with a concentrated weight

Executing the calculation for an object of heterogeneous material and subject to external point-weight

Choose the **Files?preprocess** option (to go back to preprocess).

Choose the **Data?Conditions** option. A window is opened in which the conditions of the problem should be entered.

Since the condition to be entered acts over points, select over points from the Type menu in the Conditions window.



The Conditions window

Enter the value 1e3 in the Weight box. Click Assign and select the upper corner point. Press ESC when this step is finished.

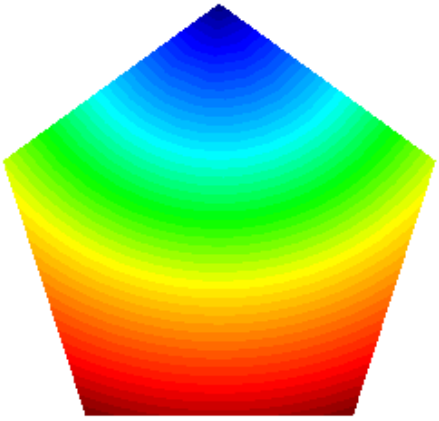
Choose **Mesh?Generate**.

A window appears in which to enter the element size for the mesh to be generated. click OK.

Choose the Calculate option from the Calculate menu, thus executing the calculating module.

Choose the **Files?Postprocess** option.

Visualize the new results.



Visualization of the distance from the mass center to each element, for an object of heterogeneous material subject to point weight

Now the condition is external point-weight. As anticipated, the new center of mass is displaced toward the point under weight.

9.4 Aditional information

NOTE: In this example, a code for the program will be developed in C. Nevertheless, any programming language may be used.

The code of the program that calculates the center of mass (cmas2d.c) is as follows:

The cmas2d.c file

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <math.h>

#define MAXMAT 1000
#define MAXCND 1000
char projname[1024];
int i, ielem, inod, icnd;
double *x, *y;
int *N, *imat;
int nodc[MAXCND];
double rho[MAXMAT], wval[MAXCND];
int Nelem, Nnod, Nmat, Ncnd;
double x_CG, y_CG;

void input(void);
void calculate(void);
void output(void);
```

Declaration of variables and constants used in the program.

```
void main (int argc, char *argv[]) {
strcpy (projname, argv[1]);
input();
calculate();
output();
}
```

9.4.1 The main program

The main program is called from the cmas2d.win.bat file and has as parameter the name of the project. This name is stored in the variable projname.

The main program calls the input (), calculate () and output () functions.

The input function reads the .dat file generated by GiD. The .dat file contains information about the mesh. The calculate function read and processes the data and generates the results. The output function creates the results file.

```
void input () {
char filename[1024], fileerr[1024], sau1[1024], sau2[1024];
```

```
FILE *fp, *ferr;
int aux,j, error=0;
void jumpline (FILE*);
strcpy(filename, projname);
strcat(filename,".dat");
fp=fopen(filename,"r");
```

The first part of the input function links the project name with the .dat extension, thus obtaining the name of the file that is to be read. This file is opened in order to be read.

The jumpline(FILE*) function is declared. This function simply reads a line from the file that it receives as a parameter, It is used to jump lines of the text when reading the .dat file.

```
for (i=0; i<6; i++) jumpline (fp);
fscanf(fp, "%d %d", &Nelem, &Nnod);
```

The first six lines of the .dat file are jumped over since these are lines of information for the user (see .bas file). Then the total number of elements and nodes of the project are read and stored in the variables Nelem and Nnod respectively.

```
x=(double *) malloc((Nnod+1)*sizeof(double)); if (x==NULL) {error=1;}
y=(double *) malloc((Nnod+1)*sizeof(double)); if (y==NULL) {error=1;}
N= (int *) malloc((Nelem+1)*3*sizeof(int)); if (N==NULL) {error=1;}
imat=(int *) malloc((Nelem+1)*sizeof(int)); if (N==NULL) {error=1;}
if (error) {
strcpy(fileerr, projname);
strcat(fileerr,".err");
ferr = fopen(fileerr,"w");
fprintf(ferr, "***** ERROR: Not enough memory. ***** ");
fprintf(ferr, "(Try to calculate with less elements) ");
fclose(ferr);
exit(1);
}
for (i=0; i<6; i++) jumpline (fp);
```

Space is reserved for storing the coordinates of the nodes (pointers x, y), the connectivities (pointer N), and the materials corresponding to each element (pointer imat).

In case of error (insufficient memory), a file is created with the extension .err. This file contains information about the error and the program is aborted.

The next six lines are jumped over.

```
/* reading the coordinates */
for (inod=1; inod<=Nnod; inod++)
```

```
fscanf (fp, "%d %lf %lf", &aux, &x[inod], &y[inod]);
for (i=0; i<6; i++) jumpline (fp);
```

The coordinates of the nodes are read and stored in the x and y variables. The node identifier indexes the tables of coordinates.

```
/* reading connectivities */
for (ielem=1; ielem<=Nelem; ielem++){
fscanf (fp, "%d", &aux);
for(j=0;j<3;j++) fscanf (fp, "%d", &N[(ielem-1)*3+j]);
fscanf (fp, "%d", &imat[ielem]);
if (imat[ielem]==0){
strcpy(fileerr, projname);
strcat(fileerr,".err");
ferr = fopen(fileerr,"w");
fprintf(ferr, "**ERROR: Elements with no material!!** ");
fclose(ferr);
exit(1);
}
}
```

The connectivities are read and the N variable is saved. This variable is a Nelem x 3- size table with two fields. The nodes (assumed triangles of 3 nodes) forming the element are saved in the first field. The element identifiers are saved in the second one.

All the elements are checked, ensuring that they have been assigned a material. If the identifier of the material is 0 (meaning that no material has been assigned to the element), an .err file is created containing information about the error and the program is aborted.

```
for (i=0; i<5; i++) jumpline (fp);
fscanf(fp, "%s %s %d",sau1, sau2, &Nmat );
for (i=0; i<3; i++) jumpline (fp);
/* reading density of each material */
for (i=1; i<=Nmat; i++)
fscanf (fp, "%d %lf", &aux, &rho[i]);
/* reading conditions*/
for (i=0; i<4; i++) jumpline (fp);
fscanf(fp, "%d", &Ncnd);
for (i=0; i<6; i++) jumpline (fp);
for (icnd=1; icnd<=Ncnd; icnd++) {
```

```
fscanf (fp, "%d %lf", &nodc[icnd], &wval[icnd]);
jumpline (fp);
}
fclose (fp);
}
```

Reading the remaining information in the .dat file.

The total number of materials is read and stored in the Nmat variable.

The density of each material are read and stored in the rho table. The material identifier indexes the densities.

The total number of conditions is read and stored in the Ncnd variable.

The nodes associated with a condition are read and stored in the nodc table indexed by the condition identifier. The value of the condition is stored in wval, another table indexed by the condition identifier.

```
void calculate ()
{
double v,aux1,aux2,aux3;
int n1, n2, n3;
int mat;
double x_CGi, y_CGi;
double x_num=0, y_num=0, den=0;
```

This is the function that calculates the center of mass.

Declaration of the local variables used in calculate().

```
for(ielem=1;ielem<=Nelem;ielem++) {
n1= N[0+(ielem-1)*3];
n2= N[1+(ielem-1)*3];
n3= N[2+(ielem-1)*3];
/* Calculating the volume (volume is the area for surfaces) */
v=fabs(x[n1]*y[n2]+x[n2]*y[n3]+x[n3]*y[n1]-x[n1]*y[n3]-x[n2]*y[n1]-x[n3]*y[n2])/2;
x_CGi= (x[n1]+x[n2]+x[n3])/3;
y_CGi= (y[n1]+y[n2]+y[n3])/3;
mat= imat[ielem];
x_num+= rho[mat]*v*x_CGi;
y_num+= rho[mat]*v*y_CGi;
den+= rho[mat]*v;
}
/* puntual weights */
```

```
for(icnd=1;icnd<=Ncnd;icnd++) {
inod= nodc[icnd];
x_num+= wval[icnd]*x[inod];
y_num+= wval[icnd]*y[inod];
den+= wval[icnd];
}
x_CG= (x_num/den);
y_CG= (y_num/den);
```

The identifiers of the nodes of the present element are saved in n1, n2, n3.

This loop makes a rundown of all the elements in the mesh. The volume is calculated for each element. (Here, the volume is the area, provided we are dealing with 3D surfaces). The volume calculations are stored in the v variable.

The geometric center of the element is calculated (coinciding with the center of gravity) and the coordinates are stored in the x_Cgi and y_Cgi variables.

The numerator sums are calculated. When the loop is finished, the following sums are stored in the x_num and y_num variables. Finally, the result of dividing the x_num and y_num variables by the den variable is stored in the x_CG and y_CG variables.

```
void output() {
char filename[1024];
FILE *fp, *fplog;
double v;
```

The output() function creates two files: .post.res, and .log.

The results to be visualized in GiD Post-process are stored in the .post.res file. It is this file that stores the data which enables GiD to represent the distance of each point from the corresponding center of mass.

The numerical value of the center of mass is saved in the .log file. The accuracy of this value is directly proportional to the element size.

```
/* writing log information file */
strcpy(filename, projname);
strcat(filename,".log");
fplog=fopen(filename,"w");
fprintf(fplog, "CMAS2D routine to calculate the mass center ");
fprintf(fplog, "project: %s ", projname);
fprintf(fplog, "mass center: %lf %lf ", x_CG, y_CG);
fclose(fplog);
```

Creating the .log file: the .log extension is added to the project name and a file is created that will contain the numerical value of the position of the center of mass, which in turn is stored in the x_CG and

y y_CG variables of the program.

Creating the .post.res file. The output data (results) are stored in this file.

The format of the .post.res file is explaned in the GiD help, see section

Posprocess data files ->Postprocess results format.

```
/* writing .post.res */
strcpy(filename,projname);
strcat(filename,".post.res");
fp=fopen(filename,"w");
fprintf(fp,"GiD Post Results File 1.0 ");
fprintf(fp,"Result MC-DISTANCE \"LOAD ANALYSIS\" 1 Scalar OnNodes ");
fprintf(fp,"ComponentNames MC-DISTANCE ");
fprintf(fp,"Values ");
for(inod=1;inod<=Nnod;inod++) {
/* distance or each node to the center of masses */
v=sqrt((x_CG-x[inod])*(x_CG-x[inod])+(y_CG-y[inod])*(y_CG-y[inod]));
fprintf(fp,"%d %lf ",inod,v);
}
fprintf(fp,"End values ");
fclose(fp);
```

In this example only a scalar result , with a single time step, is written in the .res file.

This is the full source code of this program:

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <math.h>

#define MAXMAT 1000
#define MAXCND 1000
char projname[1024];
int i, ielem, inod, icnd;
double *x, *y;
int *N, *imat;
int nodc[MAXCND];
double rho[MAXMAT], wval[MAXCND];
int Nelem, Nnod, Nmat, Ncnd;
```

```
double x_CG, y_CG;

void input(void);
void calculate(void);
void output(void);

void main (int argc, char *argv[]) {
strcpy (projname, argv[1]);
input();
calculate();
output();
}

void input () {
char filename[1024], fileerr[1024], sau1[1024], sau2[1024];
FILE *fp, *ferr;
int aux,j, error=0;
void jumpline (FILE*);
strcpy(filename, projname);
strcat(filename,".dat");
fp=fopen(filename,"r");
for (i=0; i<6; i++) jumpline (fp);
fscanf(fp, "%d %d", &Nelem, &Nnod);
x=(double *) malloc((Nnod+1)*sizeof(double)); if (x==NULL) {error=1;}
y=(double *) malloc((Nnod+1)*sizeof(double)); if (y==NULL) {error=1;}
N= (int *) malloc((Nelem+1)*3*sizeof(int)); if (N==NULL) {error=1;}
imat=(int *) malloc((Nelem+1)*sizeof(int)); if (N==NULL) {error=1;}
if (error) {
strcpy(fileerr, projname);
strcat(fileerr,".err");
ferr = fopen(fileerr,"w");
fprintf(ferr, "***** ERROR: Not enough memory. ***** ");
fprintf(ferr, "(Try to calculate with less elements) ");
fclose(ferr);
exit(1);
}
for (i=0; i<6; i++) jumpline (fp);
```

```
/* reading the coordinates */
for (inod=1; inod<=Nnod; inod++)
fscanf (fp, "%d %lf %lf", &aux, &x[inod], &y[inod]);
for (i=0; i<6; i++) jumpline (fp);
/* reading connectivities */
for (ielem=1; ielem<=Nelem; ielem++){
fscanf (fp, "%d", &aux);
for(j=0;j<3;j++) fscanf (fp, "%d", &N[(ielem-1)*3+j]);
fscanf (fp, "%d", &imat[ielem]);
if (imat[ielem]==0){
strcpy(fileerr, projname);
strcat(fileerr,".err");
ferr = fopen(fileerr,"w");
fprintf(ferr, "**ERROR: Elements with no material!!** ");
fclose(ferr);
exit(1);
}
}
for (i=0; i<5; i++) jumpline (fp);
fscanf(fp, "%s %s %d",sau1, sau2, &Nmat );
for (i=0; i<3; i++) jumpline (fp);
/* reading density of each material */
for (i=1; i<=Nmat; i++)
fscanf (fp, "%d %lf", &aux, &rho[i]);
/* reading conditions*/
for (i=0; i<4; i++) jumpline (fp);
fscanf(fp, "%d", &Ncnd);
for (i=0; i<6; i++) jumpline (fp);
for (icnd=1; icnd<=Ncnd; icnd++) {
fscanf (fp, "%d %lf", &nodc[icnd], &wval[icnd]);
jumpline (fp);
}
fclose (fp);
}
```

```
void calculate () {
double v;
int n1, n2, n3;
int mat;
double x_CGi, y_CGi;
double x_num=0, y_num=0, den=0;
for(ielem=1;ielem<=Nelem;ielem++) {
n1= N[0+(ielem-1)*3];
n2= N[1+(ielem-1)*3];
n3= N[2+(ielem-1)*3];
/* Calculating the volume (volume is the area for surfaces) */
v=fabs(x[n1]*y[n2]+x[n2]*y[n3]+x[n3]*y[n1]-x[n1]*y[n3]-x[n2]*y[n1]-x[n3]*y[n2])/2;
x_CGi= (x[n1]+x[n2]+x[n3])/3;
y_CGi= (y[n1]+y[n2]+y[n3])/3;
mat= imat[ielem];
x_num+= rho[mat]*v*x_CGi;
y_num+= rho[mat]*v*y_CGi;
den+= rho[mat]*v;
}
/* puntual weights */
for(icnd=1;icnd<=Ncnd;icnd++) {
inod= nodc[icnd];
x_num+= wval[icnd]*x[inod];
y_num+= wval[icnd]*y[inod];
den+= wval[icnd];
}
x_CG= (x_num/den);
y_CG= (y_num/den);
}
void output() {
char filename[1024];
FILE *fp, *fplog;
double v;
/* writing log information file */
```

```
strcpy(filename, projname);
strcat(filename,".log");
fplog=fopen(filename,"w");
fprintf(fplog, "CMAS2D routine to calculate the mass center ");
fprintf(fplog, "project: %s ", projname);
fprintf(fplog, "mass center: %lf %lf ", x_CG, y_CG);
fclose(fplog);
/* writing .post.res */
strcpy(filename,projname);
strcat(filename,".post.res");
fp=fopen(filename,"w");
fprintf(fp,"GiD Post Results File 1.0 ");
fprintf(fp,"Result MC-DISTANCE \"LOAD ANALYSIS\" 1 Scalar OnNodes ");
fprintf(fp,"ComponentNames MC-DISTANCE ");
fprintf(fp,"Values ");
for(inod=1;inod<=Nnod;inod++) {
/* distance or each node to the center of masses */
v=sqrt((x_CG-x[inod])*(x_CG-x[inod])+(y_CG-y[inod])*(y_CG-y[inod]));
fprintf(fp,"%d %lf ",inod,v);
}
fprintf(fp,"End values ");
fclose(fp);
free(x);
free(y);
free(N);
free(imat);
}
void jumpline (FILE* filep) {
char buffer[1024];
fgets(buffer,1024,filep);
}
```