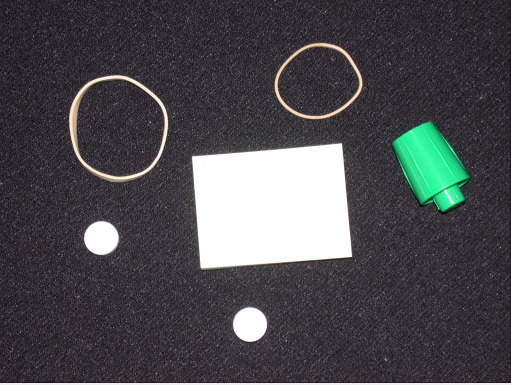
This document show two examples of the toolbox developed for Marco Aurelio and some applications. The main purpose of the toolbox is to convert images in cartesian coordinates and draw the objects of the image in the FEMM.

## ExAMPLE 1

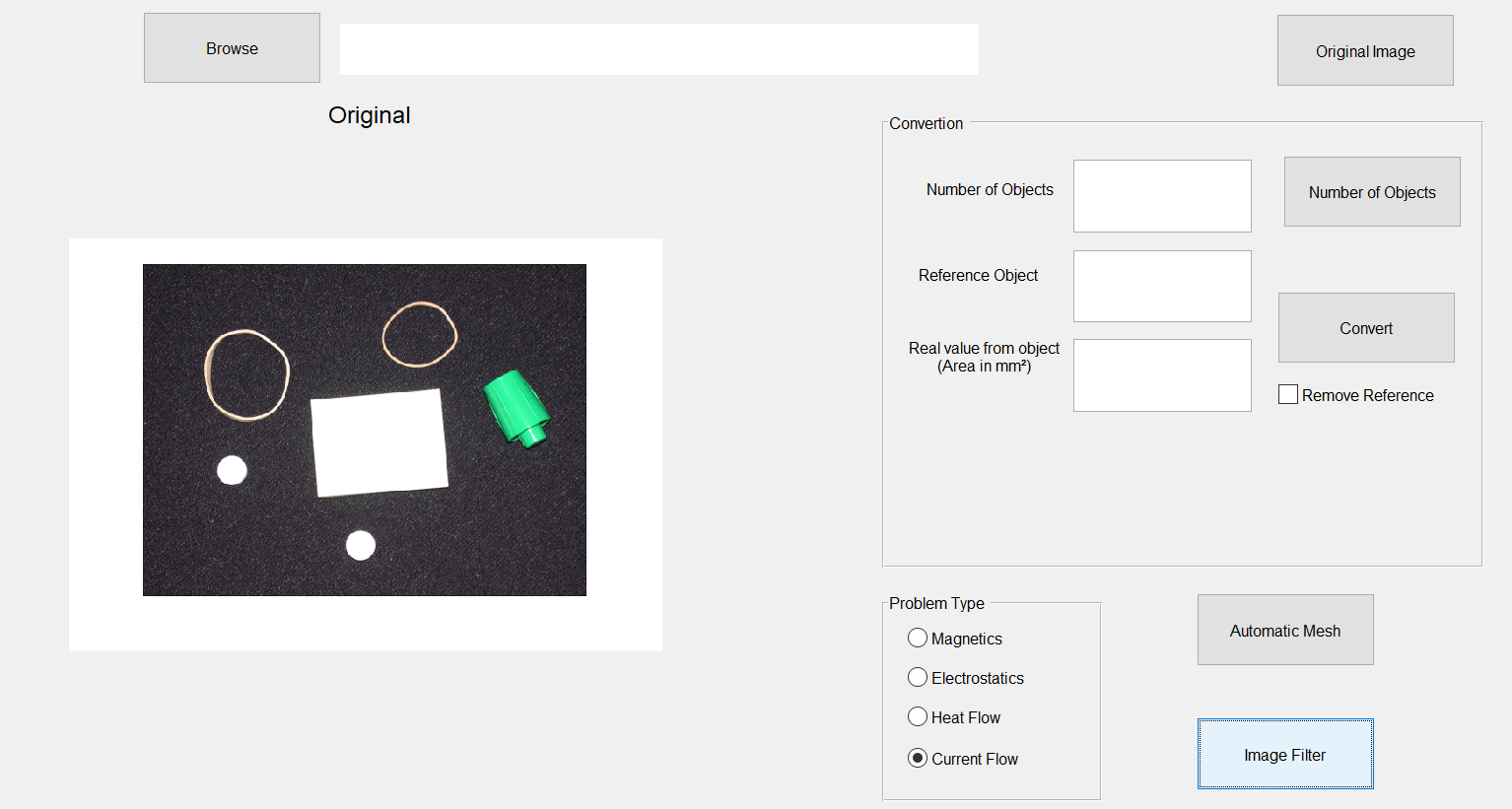
In this first example the focus will be on the image plotting in FEMM softwarewithout proceeding with the simulation, focusing only on the result of converting the image into points. For this example, we will use the image represented below:

Figure 1 - Image of Example 1



Source: MATLAB Image Database

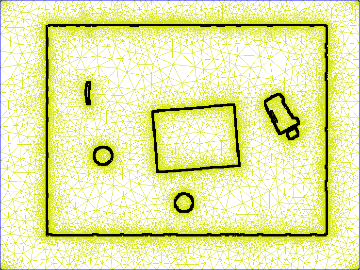
Figure 2 - Inicial window of the developed toolbox



Source: Own elaboration

After the image is loaded into the program, the type of problem is chosen (in this case it can be any type) and the conversion part to actual size will be ignored (will use the pixel size). After all these actions are performed, the button for automatic generation of the drawing is clicked.

Figure 3 - Example 1 Result with Auto Generation



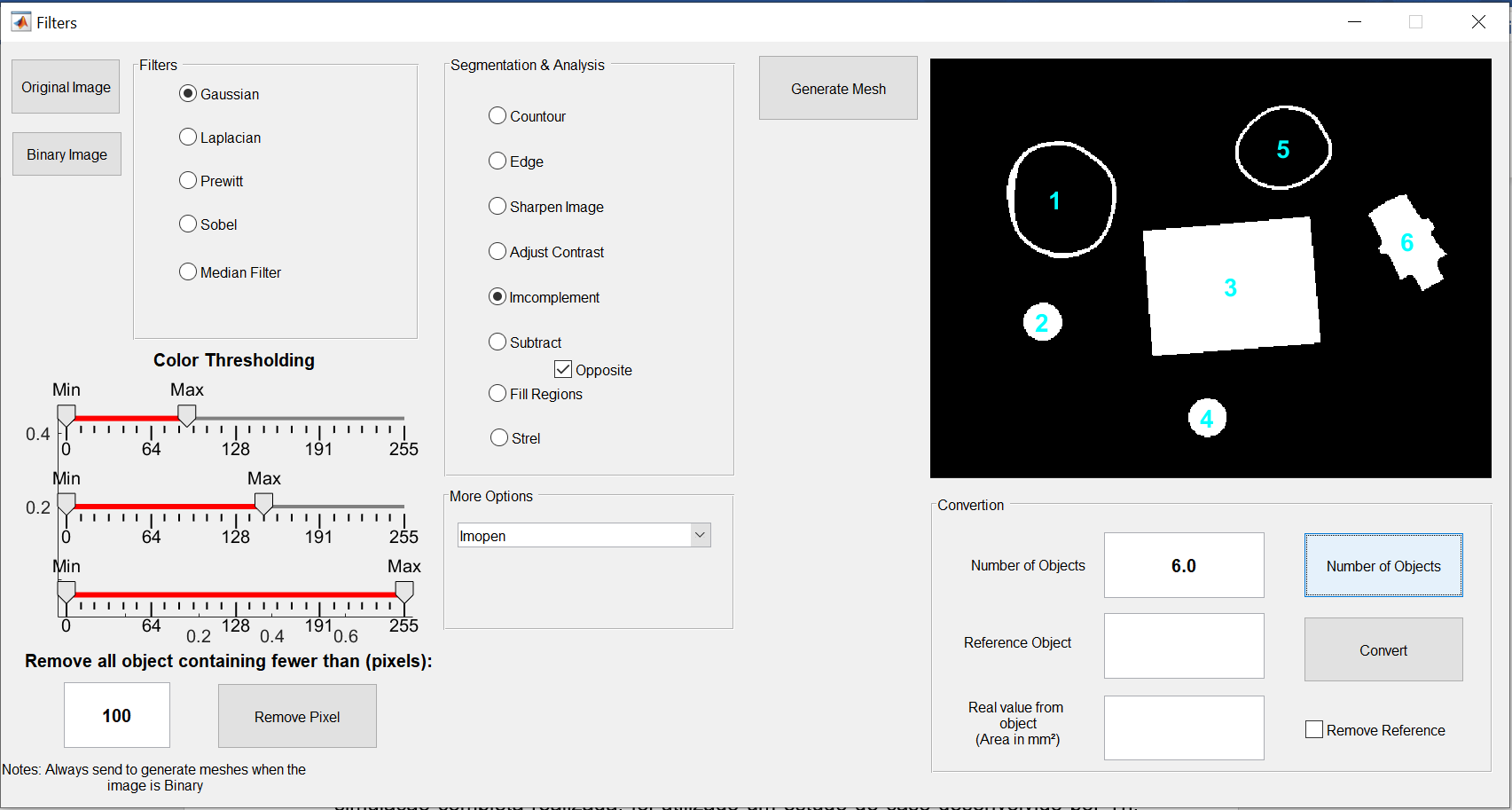
Source: Own elaboration

Figure 3 above represents the result of automatic generation of the design without any user treatment. Looking at the result, objects in the image with whiter tones obtained better representation than other objects. One of the rubber bands appeared only a part and the other did not appear and the green cover was cut out. The difference in color present in the image with distinct shades (light white and darker green and yellow) makes automatic processing difficult and as a result you get a noisy drawing with missing objects.

For the image to be as close to the real as possible, a treatment will be necessary to make the drawing. To do so, keep the choice of the previous problem type and click the button to open the filter window and other tools. In this window, the following process was performed:

1. Selects the *Sharpen Image* option to convert the image to grayscale;
2. Then select the *Adjust Contrast* option to make the tone differences even more pronounced and make it easier to separate objects;
3. The *Gaussian* filter is *chosen,* which has the function of blurring the image and making the edges softer;
4. Then click the *Binary Image* button to convert the image to binary format;
5. The image will be in inverted colors (where it was supposed to be white, it is black). Therefore, *I m complement* is selected to invert the colors of the image;
6. Finally, select the option *Imopen* , the panel *More The ptions* to close the objects that were faulty or "open";
7. To see if the program will detect all objects, you can click the *Number of Objects* button to see if the number of objects counted by the *toolbox* equals the number of objects observed.

Figure 4 - Toolbox screen with example user choice

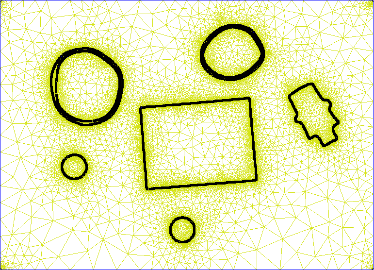


Source: Own elaboration

Figure 4 above illustrates the options selected before sending the drawing to FEMM. In figure 4 only the *Imcomplement* function is selected because it was the last one selected. Since it is a unique selections panel, the last option selected is the one that is highlighted. Therefore, the user simply selects the options then the functions will be applied to the image and will be update in real time. The result of the drawing in FEMM is shown in figure 5.

.

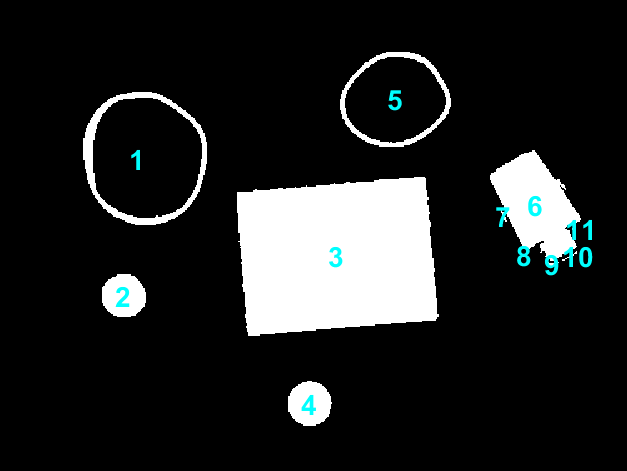
Figure 5 - Drawing result in FEMM software



Source: Own elaboration

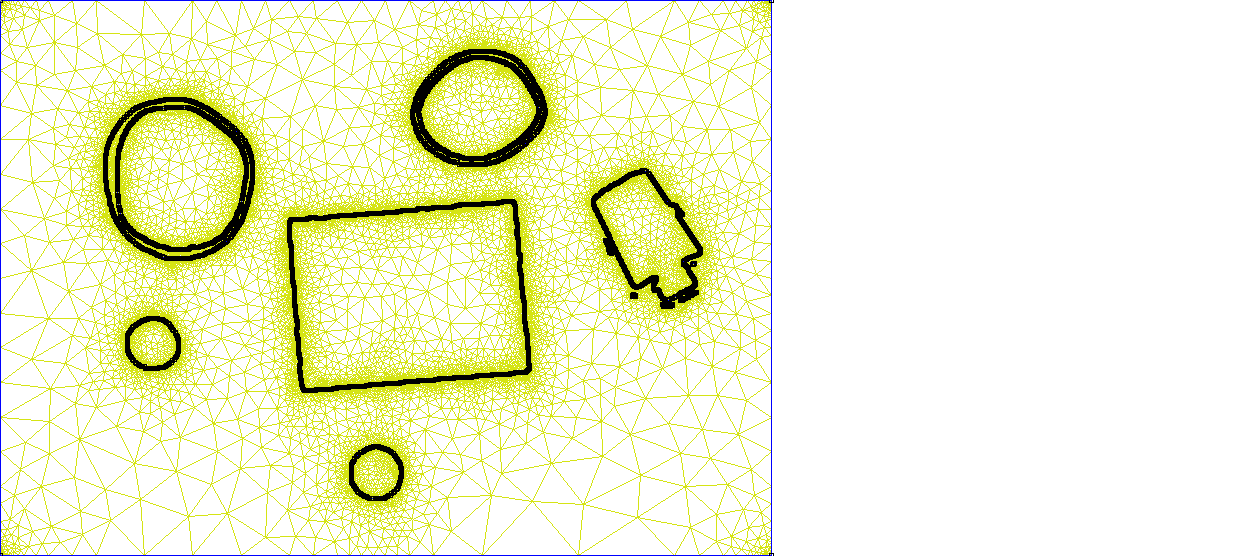
Which process to choose depends on your preference or how you want the design to look. Alternatively, for example, could be used the *sliders* of thresholding( *Color Thresholding*) like it is in Figure 4, shortly after the procedure 1-5 mentioned above, and then, due to the quantity of noises, eliminate pixels with a value of less than 30. Thus, the result would be represented by Figure 6 (binary image) and Figure 7 (drawing in FEMM).

Figure 6 - Binary image resulting from alternative process



Source: Own elaboration

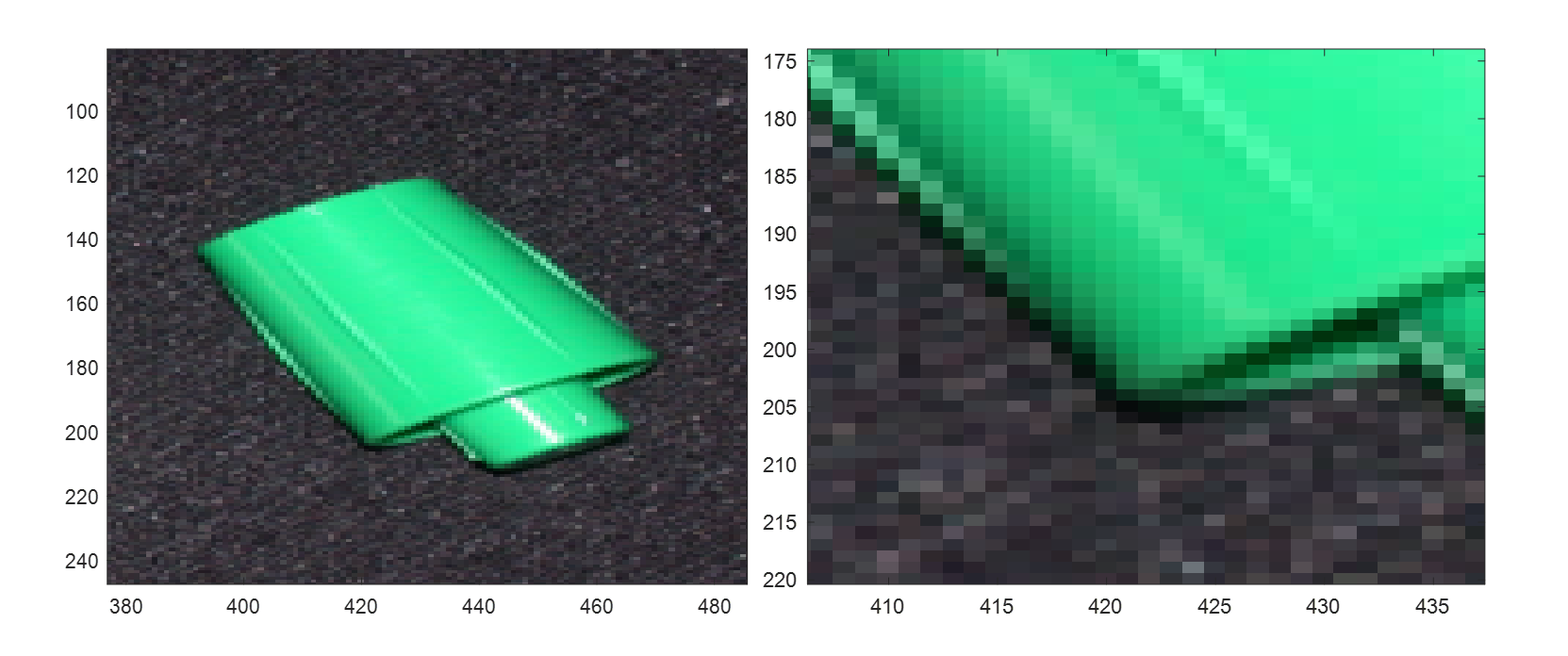
Figure 7 - Alternative Process FEMM Result



Source: Own elaboration

In both cases, it was not possible to obtain a satisfactory result of the green cap in the image. This is due to the existing difficulty in image processing to understand where the edge ends of the object (*foreground*) and starts the back of the image (*background*).

Figure 8 - Green cover removed from image before simulating



Source: Own elaboration

The image above has the zoomon the green bottle cap and is possible to see that there is a shadow that resembles the background of the image. When converting to binary image, or even to grayscale image, the shaded parts of the bottle cap may be confused with the back of the image. For best results in segmentation, it is interesting that the background of the image is neutral or has very different colors from the objects they wish to extract from the image under analysis.

## EXAMPLE 2

The example that will be worked on in this section is a coin of 10 centavos (Brazilian currency) and a circular fridge magnet, considering the actual size of the objects. The image to be simulated is figure 9. The idea of the example is to show the operation of the actual size option and the application of the program in a hypothetical simulation situation applied to everyday life.

Figure 9 - Coin and magnet

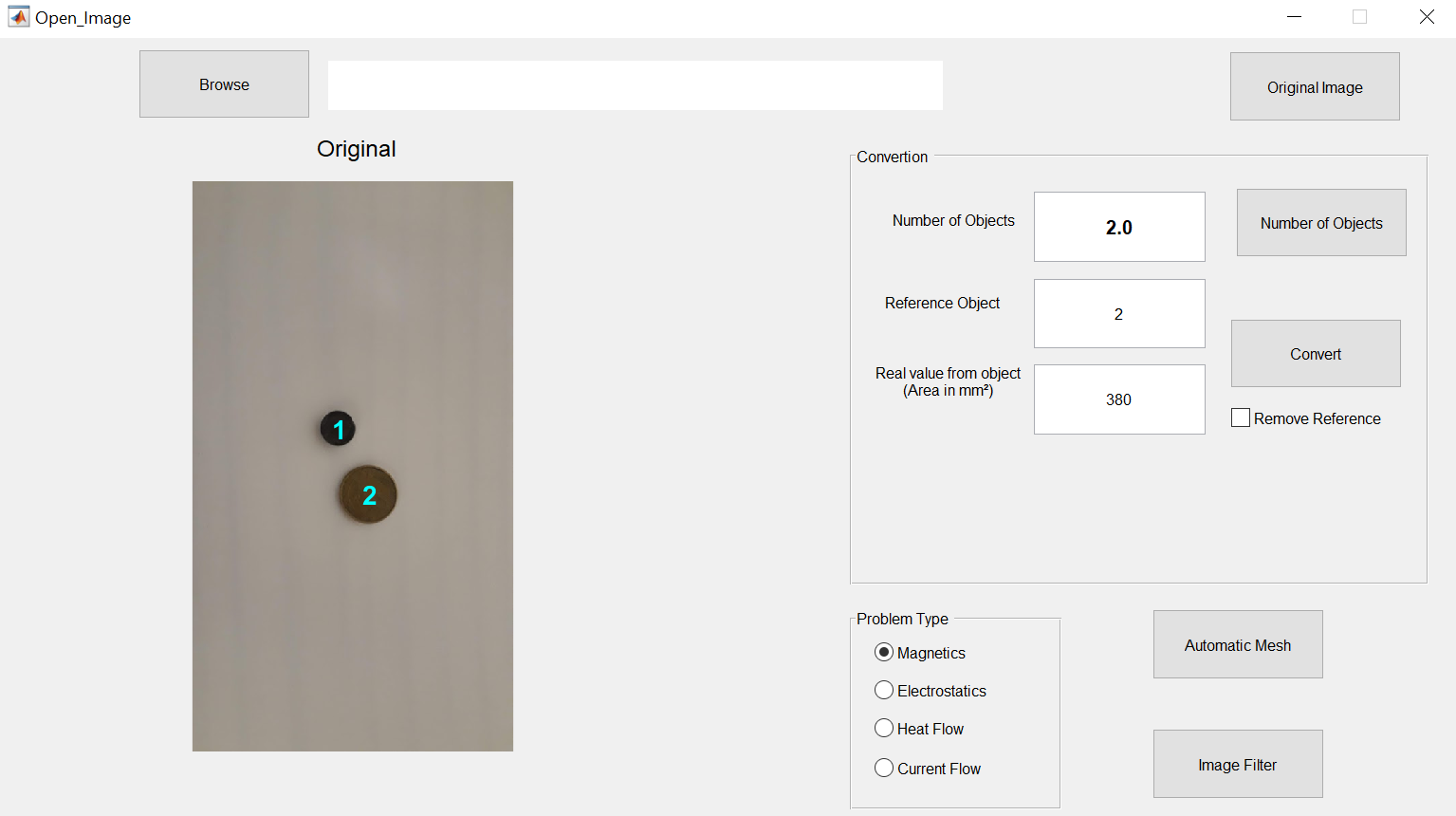


Source: Own elaboration

Right after loading the image into *Open Image* (figure 2), the first action to take is to click the *Number of Objects* button to have the objects identified and their respective numbers to reference the object with known area. The result of the numbering of the objects is shown in figure 10. Figure 10 shows that the magnet is numbered 1 and the coin is numbered 2.

The object with known dimensions is the coin which is 22 mm in diameter and 2.23 mm in thickness. Therefore, in the *Reference Object* field should be placed the number 2 and in the *Real Value from Object* field should be placed the value of the object area which is approximately 380 mm² (π.R², where R is half the diameter). For the current example, there is no need to remove the reference object as it will be used in the example. Figure 10 shows how the options have to be entered.

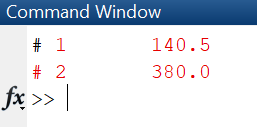
Figure 10 - Open Image Screen



Source: Own elaboration

In order to check if the conversion is in ok, the MATLAB command screen shows the value of the area of each object. The result of the object area is 140.5 mm², as shown in figure 11. The value approximates the real that revolves around 135 and 143 mm² (considering the diameter of the magnet between 13 and 14 mm²).

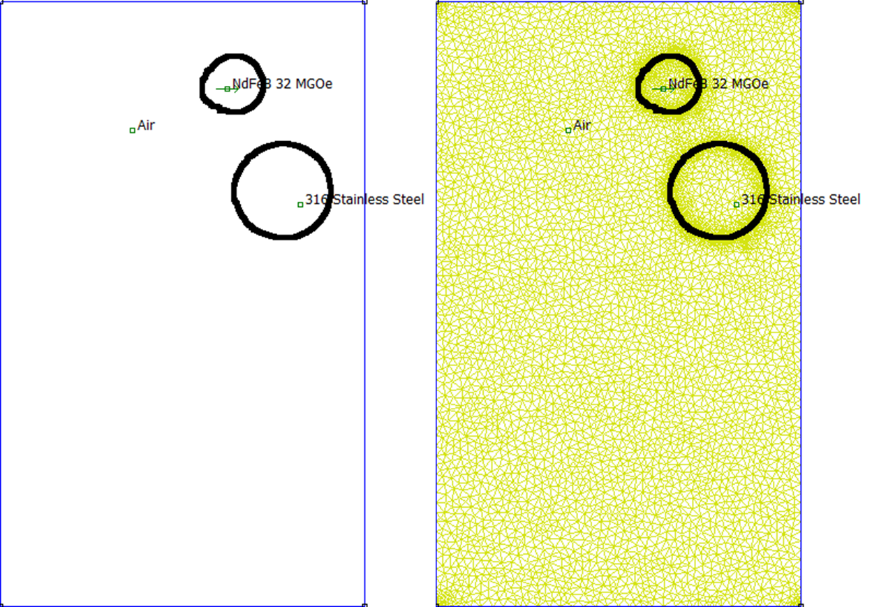
Figure 11 - Object Areas in MATLAB Commmand Window



Source: Own elaboration

With the conversion process verified and validated, click *Automatic Mesh*. The magnet NdFeB 32 MGOe (Neodymium Iron Boron) was considered for the magnet and the material was stainless steel. The result of the drawing in the FEMM *software* and the meshes created is illustrated in figure 12.

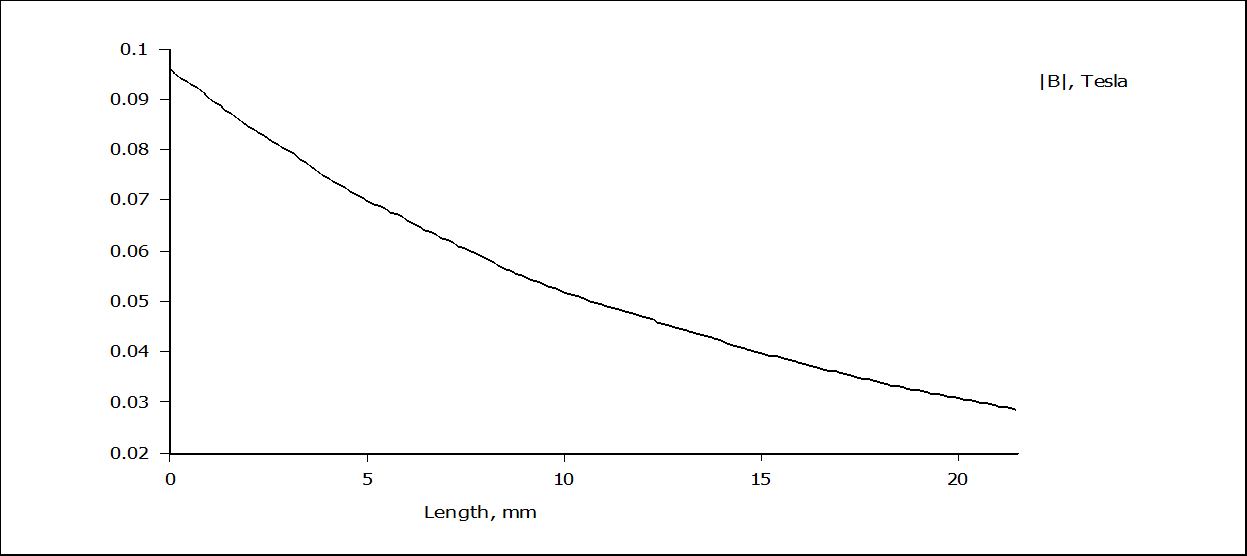
Figure 12 - Image drawn in FEMM



Source: Own elaboration

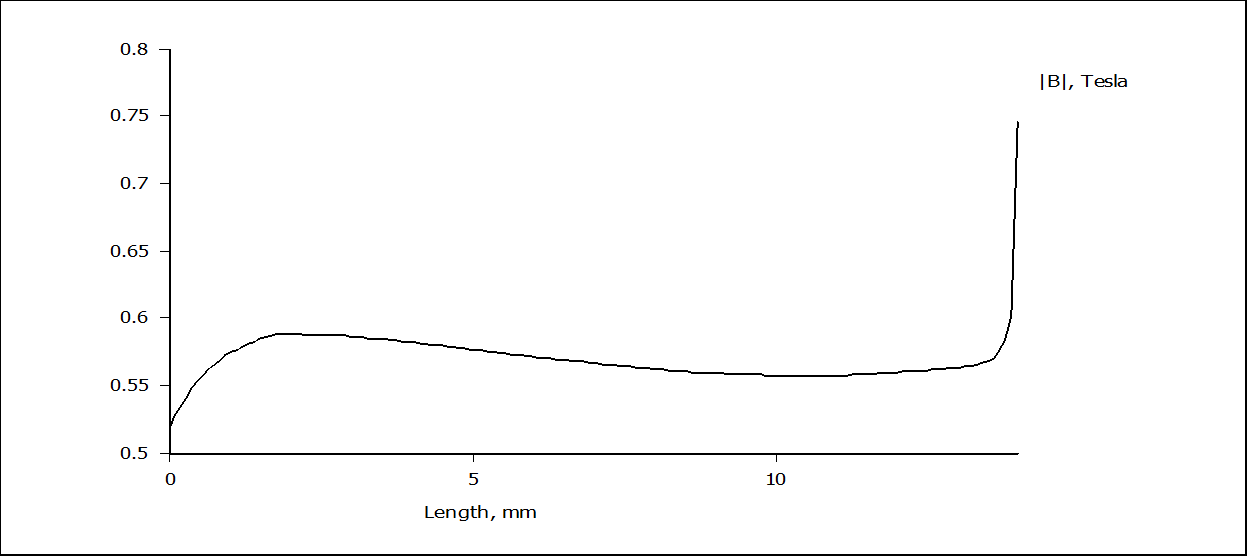
The simulation results are observed in figures 13, 14 and 15.

Figure 13 - Magnitude of flux density in coin



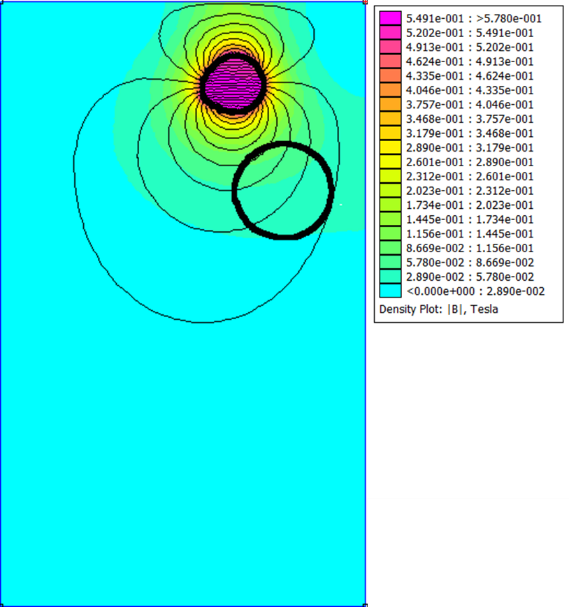
Source: Own elaboration

Figure 14 - Magnetic flux density of magnet



Source: Own elaboration

Figure 15 - Flux Density



Source: Own elaboration

Note that the magnetic field in the coin is low and not strong enough to draw the coin near the magnet, which was intended at the time of the photo. When approaching the coin from the region where the field lines are closest to each other, the coin is attracted to the magnet.