By downloading the GitHub Repository you will get a folder containing several matlab functions, some sub-folders and the main code “Slicer\_main”. The entire slicing process is done inside this code.

1. Open the “Slicer\_main” code”.
2. By running the section **STL** you can load the stl file you want to calculate the slicer for (cylinder.stl in the following example). Make sure that this file is saved in the same directory as Slicer\_main.

%% STL

TR = stlread("cilindro.stl");

fv.vertices = TR.Points;

fv.faces = TR.ConnectivityList;

figure;

trisurf(TR, 'FaceColor', 'cyan', 'EdgeColor', 'none');

camlight; lighting phong; title('Original Mesh');

axis equal

grid off

axis off

1. By running the section **Unfolding of the volume in the plane** the volume is cut and unfolded out in the plane.

The next section **Volume visualisation (with normals)** allows you to view the volume as it has just been unfolded. By removing comments from commented lines of code, you can view the unfolded volume and the normals to each triangle in the mesh.

1. 4) At this point there is the volume reconstruction phase. To this end, there are three alternative paths in the code that must be followed depending on the complexity of the object to be processed. The three paths can be followed by running the following sequences of sections:
2. **DIRECT CLOSING OF HOLES**
3. **Upper surface insulation MODE1**/2 -> **Reconstruction of the volume starting from the upper surface 1**
4. **Upper surface insulation MODE2/2** -> **Reconstruction of the volume starting from the upper surface 2**

“a” is the preferred option when the volume has a complex geometry, where the presence of some cantilevered structures could make it difficult to identify and isolate the upper surface of the volume.

“b” is the option to follow when the unfolded object has a simple geometry. In particular, it is the best and simplest option at the computational level when the isolated upper surface has few holes and therefore few edges.

“c” is the most complete and complex option, which allows you to reconstruct the volume of textured or networked geometries, where the upper surface therefore has a high number of edges. It is ideal if you want to reconstruct volumes such as stents.

1. Running the section **Save and view the reconstructed STL file** you can view the newly reconstructed volume and save it to an .stl file.
2. The section **Slicer IMAGObot** p allows you to select all the parameters related to the subsequent slicing phase of the unfolded volume just created. Before running the section, it is essential to correctly set all the parameters present in the code. In particular, the *NonPlanarPrintCheckBox* parameter allows you to choose whether you want to calculate a planar or non-planar slicing, setting it to 0 or 1 respectively.
3. Running the section **Trajectory txt file** a .txt file containing the print trajectories just calculated in the slicing process is generated. It is convenient to save this file in the same directory as Slicer\_main, since it will be called later.
4. Running the section **Import of the trajectory txt file** the .txt file containing the trajectories calculated in the slicing process is imported.

**IF you already have a text file containing the trajectories, just save it in the main directory (where the Slicer\_main file is stored) and start from this point, avoiding all the previous ones.**

1. Running the section **Printer coordinates: deg(°)-ROTATION** the coordinates present in the .txt file imported in 8) are “wrapped” around the mandrel by converting it to cylindrical coordinates.

The fundamental parameters to set before running are the radius (r\_mandrel) of the mandrel and “safe” (the overlap level of the two ends of the rewound volume).

1. The last two phases concern the definition of the commands that must be inserted in the g-code and the actual generation of the .ngc file. Depending on whether you want to generate them for EBB printing or for FDM printing, the following sections must be run:

EBB) **G-code COORDINATES – EBB -> G-code writing – EBB**

FDM) **G-code COORDINATES – FDM -> G-code writing – FDM**