Q) **How you structured the code ?**

A) The code is organized into a class-based structure named MobiusModel, it contains the following methods with names **__init__( ),_generate_mesh( ), surface_area_calculation( ), edge_length_calculation( ), display_3D_Model( ).**

Working of each method :

__init__( ): This method initializes the strip with parameters like Radis(R), Width(W), Resolution(n).

generate_mesh( ): This method contains the parametric equations of strip and it handles the mesh generation.

surface_area_calculation( ): This method calculates the surface area, mathematical calculation takes place. Surface area is calculated by using the norm of the cross product of the surface tangents.

edge_length_calculation( ): This method calculate the edge length, mathematical calculation takes place. It calculates the total edge length along the Mobius strip boundaries , Euclidean distance is used to calculate the edge length.

display_3D_Model( ): This is the method used to visualize the whole things in a 3-D plane.

Q) **How you approximated surface area?**

A) Surface area is calculated by using the norm of the cross product of the surface tangents.

Step_by_step explanation :

- x,y,z are parametric equations values which calculated in _generate_mesh_() method
- First partial derivatives are calculated for x,y,z of the surface in u and v directions using numpy.gradient.
- The norm of the cross product gives the area of the infinitesimal patch at each mesh point.
- Adding up all these areas of the patches and multiplying by the parameter step sizes (du and dv) provides the approximate total surface area.

Q) **Any challenges you faced?**

A) Some challenges I faced:

- Understanding the parameters correctly inorder to handle geometry and math .
- Surface are calculation to ensure the directions of th gradient computed is right or wrong
- Plotting the 3D model to get the correct colors, orientation and resolution to appealing output