**Code Structure**

* **Class-Based Design:**  
  Mobius Strip encapsulates all relevant methods and parameters.
* **Mesh Generation:**  
  \_generate\_mesh() computes 3D coordinates (X, Y, Z)(X, Y, Z)(X, Y, Z) from parametric equations over a grid of (u,v)(u, v)(u,v).
* **Numerical Derivatives:**  
  compute\_surface\_area() calculates partial derivatives and uses the cross product to estimate infinitesimal area elements.
* **Edge Calculation:**  
  compute\_edge\_length() approximates the length of one boundary edge via Euclidean distances.
* **Visualization:**  
  plot() provides a 3D rendering using matplotlib.

**Surface Area Approximation**

* Uses **numerical integration** over a parameterized surface.
* Computes partial derivatives.
* The magnitude of their cross product gives the local surface area element dA.
* Area is obtained by **double trapezoidal integration** over the parameter domain.

**Challenges**

* **Non-trivial Geometry:** The Möbius strip is non-orientable, which can confuse naive surface integration; however, using a parametric representation avoids topological pitfalls.
* **Numerical Stability:** Gradient estimates are sensitive to resolution n; low n can yield inaccurate derivatives and area.
* **Trapezoidal Rule Assumptions:** Works best with fine grids; accuracy depends heavily on uniform sampling.