

# Karkhana.io -Backend assignment

## WRITE UP

NAME :CHAMIKAR .P

EMAIL: [pchamiakr@gmail.com](mailto:pchamiakr@gmail.com)

PH.NO:8374857655

GITHUB REPOSITORY:

### How I Structured the Code

I used a class-based approach to keep things organized and modular. The class is called MobiusStrip, and it takes in three main inputs: the radius (R), the width of the strip (w), and the resolution (n) for how finely we want to sample points. Inside the class, I defined methods to:

- Generate the 3D mesh using the parametric equations
- Calculate surface area using numerical integration
- Estimate edge length by summing distances along the strip's boundary
- Plot the Mobius strip in 3D using matplotlib

This made the code cleaner and easier to work with, especially when testing or tweaking individual parts.

### How I Approximated Surface Area

To compute the surface area, I used a mathematical technique that involves taking the cross product of partial derivatives from the parametric equations. This gives us the local surface patch area, and I integrated these values over the surface using scipy's dblquad function.

It's basically an approximation of:

$$\text{Area} = \iint \left\| \frac{\partial \vec{r}}{\partial u} \times \frac{\partial \vec{r}}{\partial v} \right\| du dv$$

It worked well for a smooth shape like the Möbius strip, even though it's a bit tricky due to the twist in the surface.

### Challenges I Faced

One challenge was figuring out the derivatives correctly and making sure the cross product gave the right result. If anything was off, the area calculation would be wrong or give weird values.

Another challenge was visualization the plot didn't look good at first because of how the mesh was sampled. I had to experiment with the resolution and rendering options to get a clean 3D plot.

Lastly, making the surface area calculation stable and accurate took a few tries. Higher resolution improves accuracy but also slows down the program.

