# Mobius Strip Assignment Write-Up

## Understanding the Mobius Strip

To approach this assignment, I began by learning what a Mobius strip is. I explored several resources and videos:

- What is a Mobius Strip?:

<https://youtu.be/77DPTLeorJE?si=2T8Z9dOpsr5q9f4U>

- Mobius strip is Non-Orientable surface :

<https://youtu.be/S48JsV-pCBo?si=1jpu_tGu2kzEmBaZ>

I came to know that

* Mobius strip is a non-orientable surface
* it has only one side and one boundary

## Parametric Equations

Later I started understanding the parametric eqns of the mobius strip

<https://youtu.be/uC-2Y8c_O1k?si=WU1QEwDK9R-BqAuU>

I understand

* The parametric form of the Mobius strip is:

x(u,v) = (R + v \* cos(u/2)) \* cos(u)

y(u,v) = (R + v \* cos(u/2)) \* sin(u)

z(u,v) = v \* sin(u/2)

Where:

- u ∈ [0, 2π]

- v ∈ [-w/2, w/2]

## 3D Plotting with Matplotlib

I practiced 3d plotting using matplotlib by learning from the youtube video

<https://youtu.be/melJBPTAQCw?si=pe_x7ybA3Vd86s0y>

I practiced by plotting the Mobius strip with a basic Python script using random parameters for R, w, and resolution n.

Code :

from mpl\_toolkits import mplot3d

import matplotlib.pyplot as plt

import numpy as np

R = 1

w = 1

n = 30

u = np.linspace(0, 2 \* np.pi, n)

v = np.linspace(-w / 2, w / 2, n)

u, v = np.meshgrid(u, v)

x = (R + v \* np.cos(u / 2)) \* np.cos(u)

y = (R + v \* np.cos(u / 2)) \* np.sin(u)

z = v \* np.sin(u / 2)

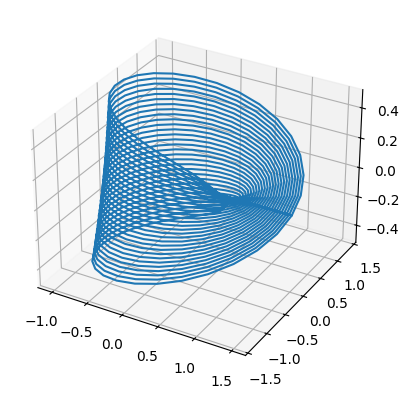
fig = plt.figure()

ax = plt.axes(projection="3d")

ax.plot(x,y,z)

plt.show()

result:



## Code Structure

Then I started to write a Mobius strip class which plots the 3d visualization with given parameters , computes surface area, and also the edge length

Steps:

- Initialization: Takes R, w, and n as inputs.

- Mesh Generation: Computes the (x, y, z) mesh grid using the parametric equations.

- Surface Area Calculation: Computed using numerical integration over the mesh.

- Edge Length Calculation: Calculated by taking the square root over sum of squares of differences between the all consecutive points at the edge

## Concepts used from numpy

* np.gradient – to calculate the partial derivative
* np.diff – to find the sqrt of sum of differences b/w the consecutive points at the edge

## Challenges Faced

- Understanding the geometry of a Mobius strip from scratch.

- Translating the parametric equations into programmatical language.

- computing the surface area and edge length.

- Adjusting the mesh resolution for 3D plotting.

## Conclusion

This assignment helped me understand

* what Mobius strip is?
* How to plot 3d plots using matplotlib?
* Some concepts of numpy like np.gradient(), np.diff() etc.

Starting from no prior knowledge, I gained insight into non-orientable surfaces and how to model and analyze them using Python.