Phenomenon #10 مشاہدہ #10

Moving String, Static Shape

ثابت شکل کی متحرک ڈوری

Muhammad Aryaan Ijaz, Khadim Mehmood and Muhammad Sabieh Anwar

July 18, 2025

Everyday phenomena often hide surprising physics behind seemingly simple observations. The String Shooter experiment challenges our intuition about flexible materials in motion. When propelled at high speed by motor-driven wheels, a seemingly limp string unexpectedly forms a stable, self-supporting loop. This dynamic loop behaves not as a passive, floppy object but rather as an active, almost spring-like structure.

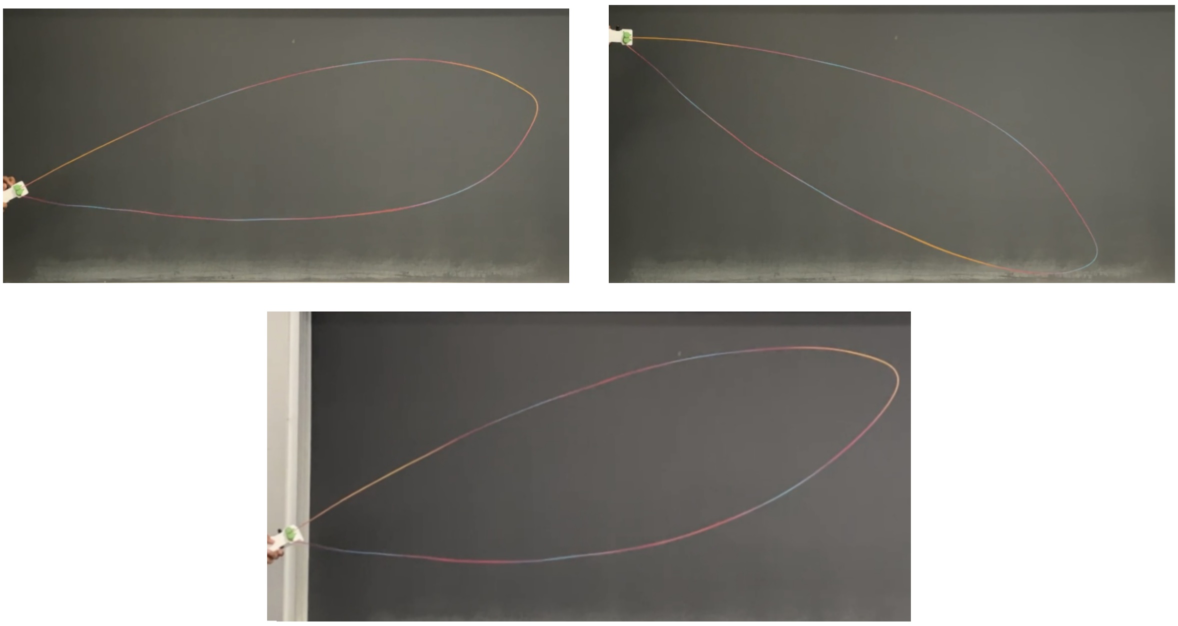


Figure 1: The string shooter loop as seen at different angles.

The underlying physics involve a delicate balance of forces: inertia, which urges the string to continue in motion; tension, generated by the continuous loop and the drive of the wheels; and aerodynamic drag, which interacts with the string as it slices through the air. The String Shooter experiment allows you to explore how drag and other forces work together to sustain a loop in equilibrium.

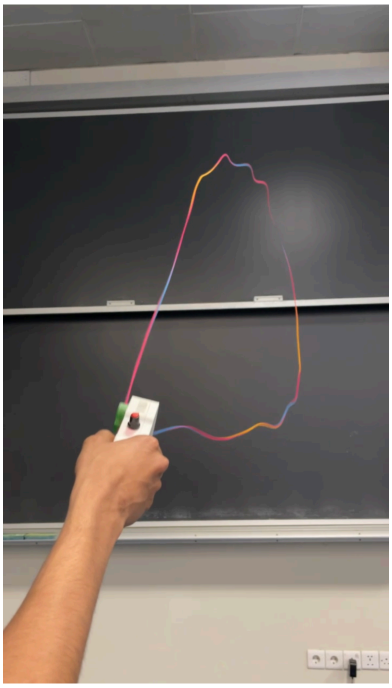
The objectives of the experiment are as follows:

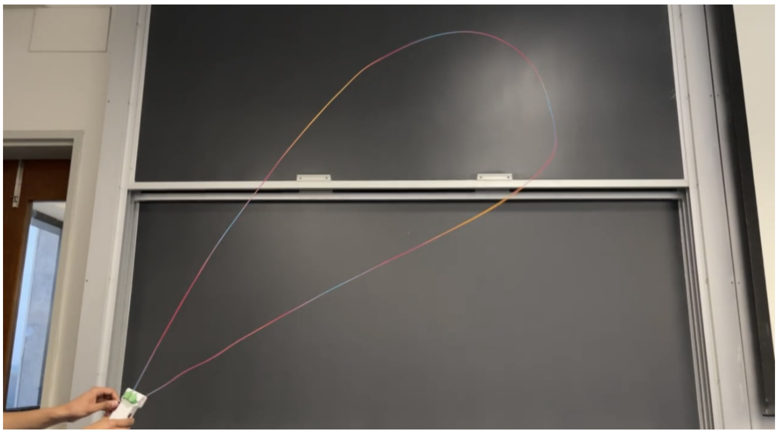
1. **Theoretical Insight:** Deepen your understanding of the interplay between inertia, tension, and drag forces.

2. **Simulation:** Use MATLAB to simulate how the loop's shape changes with varying drag coefficients.

3. **Data Collection:** Capture high-quality images of the string shooter in operation using software such as ImageJ or Automeris.

4. **Data Analysis:** Analyze the images via curve fitting to extract an effective drag coefficient and compare it with simulation predictions.





More images of the string shooter in action.

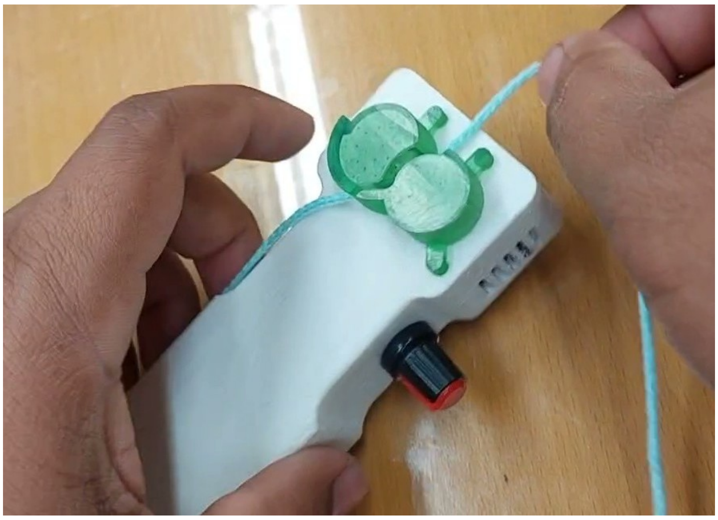


Figure 2: An overview of the string shooter set up.

You will require the following apparatus/measurement software

1. A smartphone/any high quality camera
2. Ruler
3. Analytical balance
4. Image J (calibration software)
5. Automeris (data collection)
6. MATLAB

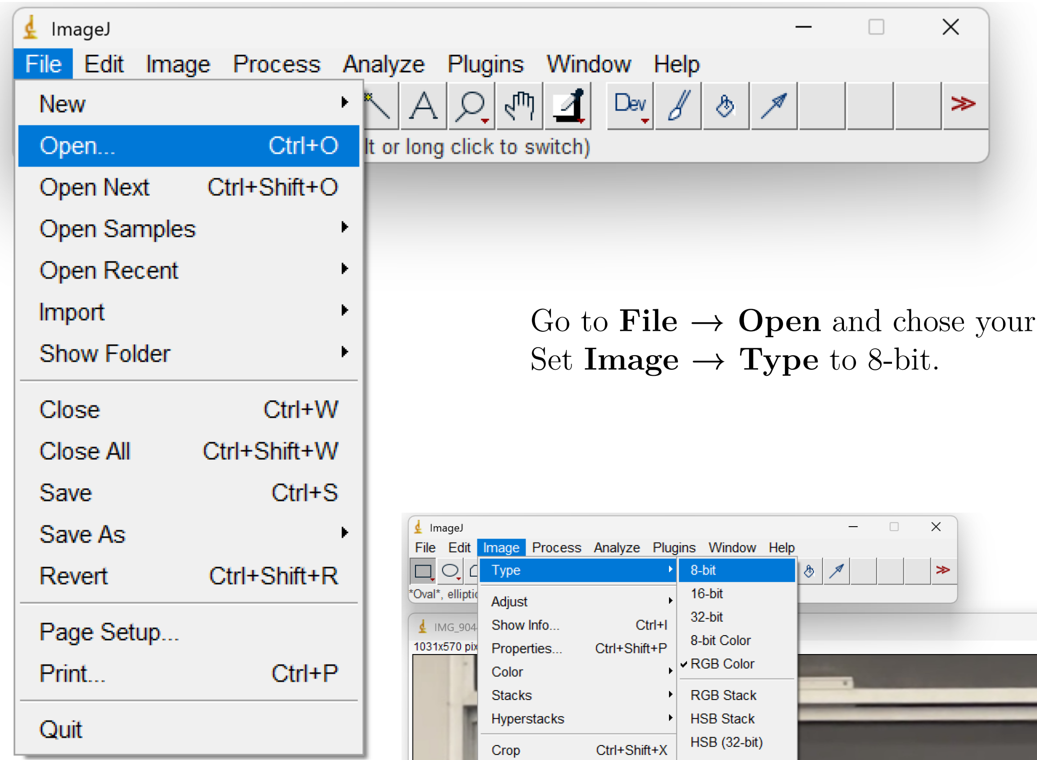
Start by setting up the string shooter using the equipment provided. Measure the length of the string, the mass of the string, and the radius of the wheels on the string shooter. The string loop must be positioned between the wheels before the shooter is turned on. The speed controller on the side allows you to effectively adjust both the shape of the loop and the angle of projection and return. Experiment with a few different speeds and observe the various loop shapes produced. Insert an image here showing the string shooter setup and several loop shapes (e.g., “Figure 1: Experimental Setup and Loop Shapes”).

Once you have observed several loop shapes, select an angle or speed that you find interesting and try to maintain that loop shape. Ask a lab partner to take a high-quality picture of the string loop with a calibration object (such as a ruler) included in the image. This reference will be very helpful later. Make sure the image is of sufficient quality to clearly distinguish the edges of the loop.

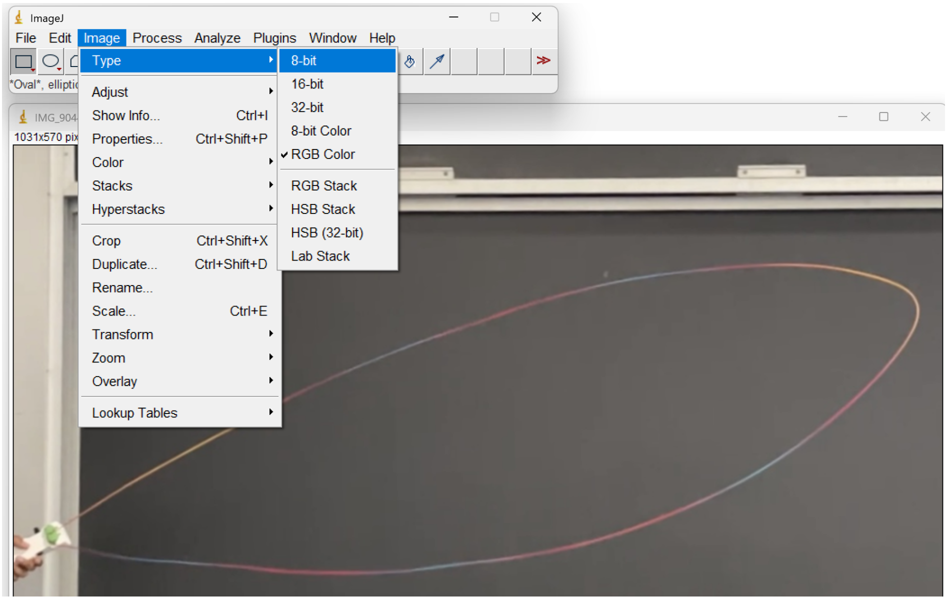
Open the captured image in ImageJ. Convert the image to 8-bit format for better clarity. Using the ruler or another reference object, draw a line to calibrate the distance. Then, using the same procedure, measure the horizontal distance of the loop and the vertical distance between the highest and lowest points of the loop. Next, use the angle measurement tool in ImageJ to record two angles: one from the top of the wheels Sto the bottom of the string (angle of return) and another from the top of the wheel to the top of the string (angle of return).

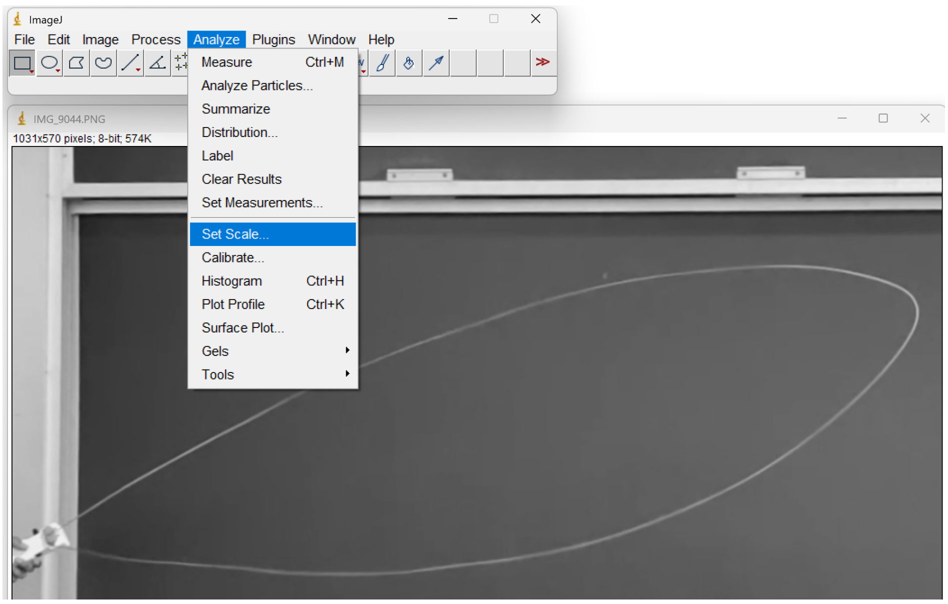
After recording these measurements, save the 8-bit image and load it into the Automeris software. Select ”2D XY Axes” and place the two X and two Y pointers at the same positions you used in ImageJ, then enter the measured distances. Select the pen tool and highlight one part of the string. Data points will appear as red dots. Click on ”View Data” and download the corresponding CSV file. Repeat this process for the lower part of the loop.

1 ImageJ.Js Software

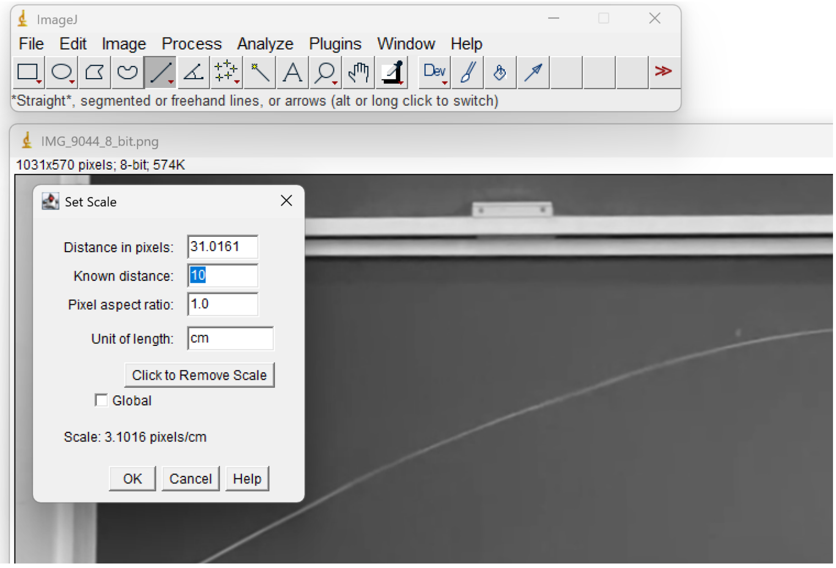


Go to **File** → **Open** and chose your image.
Set **Image** → **Type** to 8-bit.

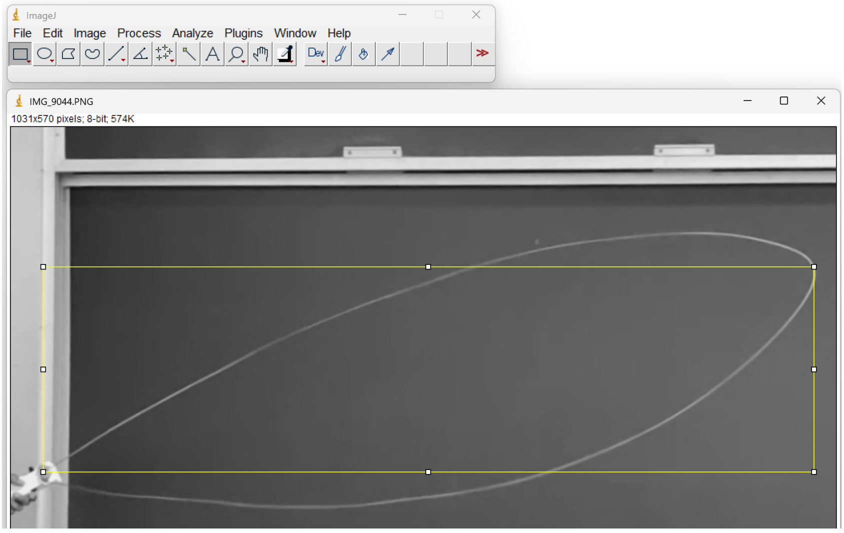




Use the Line Tool. Go to **Analyse** → **Measure** and calibrate by using Set Scale.



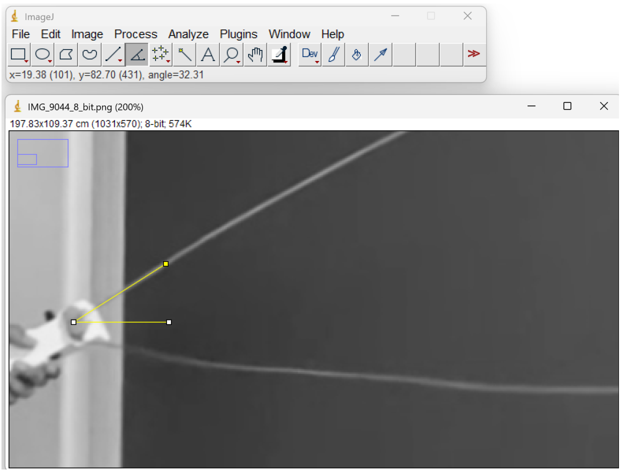
Set the value of known distance and units then press enter.



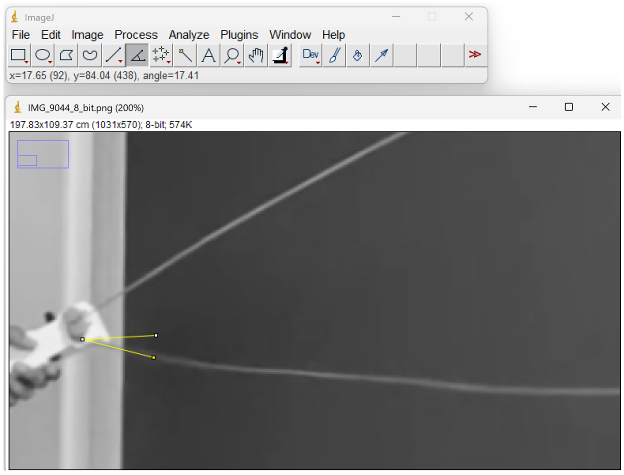
Measure the horizontal and vertical distance as specified by the rectangle in the image.

1.1 Angle measure

Click on the angle tool in the toolbar.

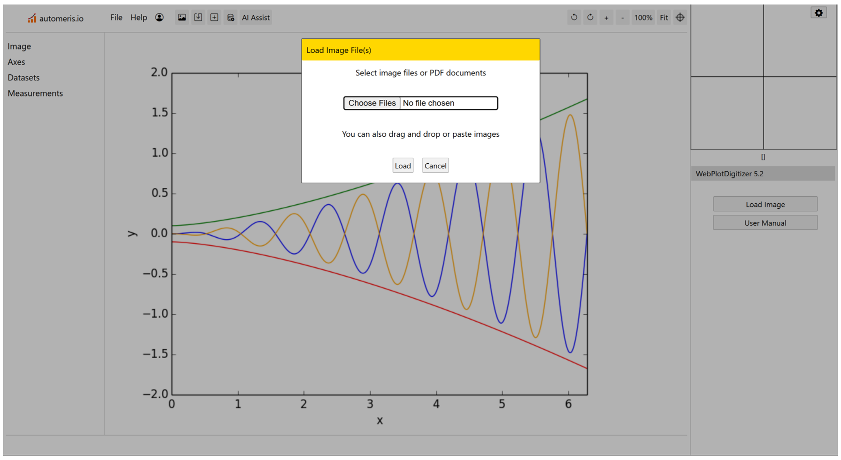


Measure the angle of projection



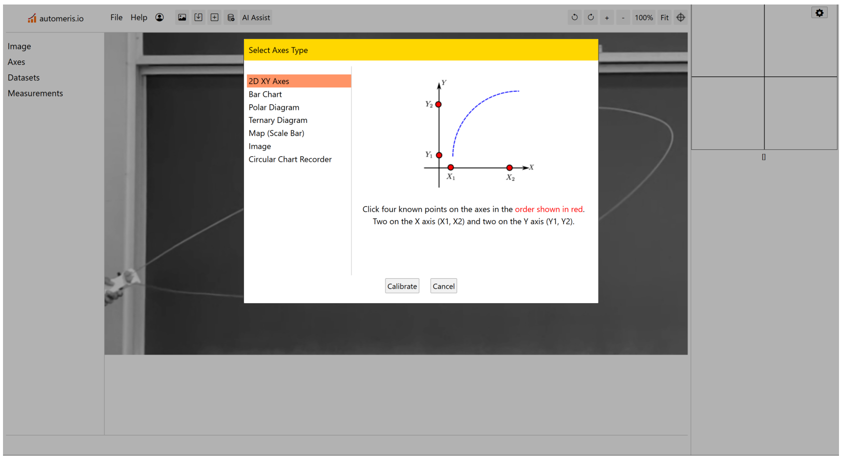
Measure the angle of return

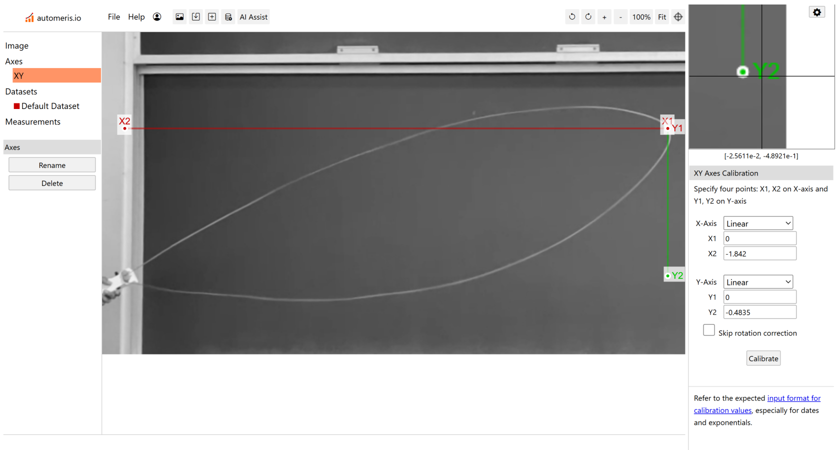
2 Automeris software



Click on choose files, and load your image.

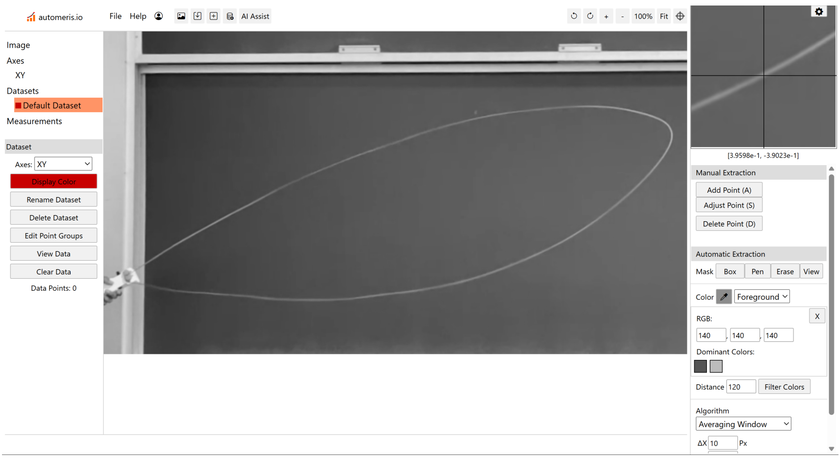
Select 2D XY Axes and then click on calibrate.

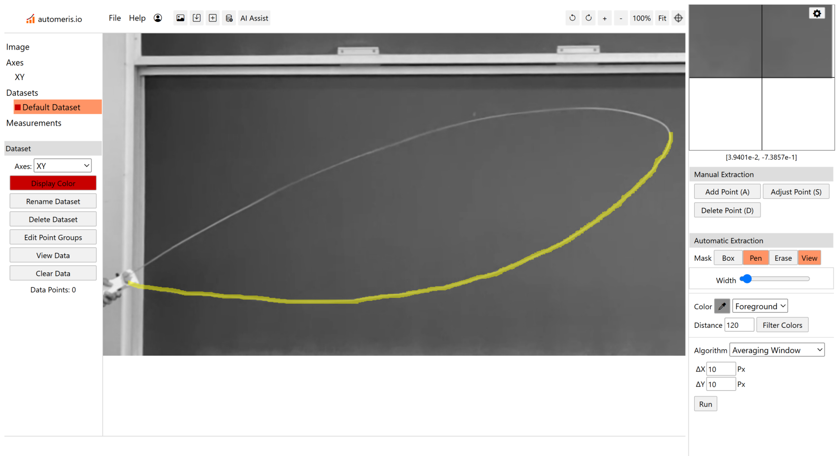




Position X1, Y1, X2 and Y2 on your image in the following manner.

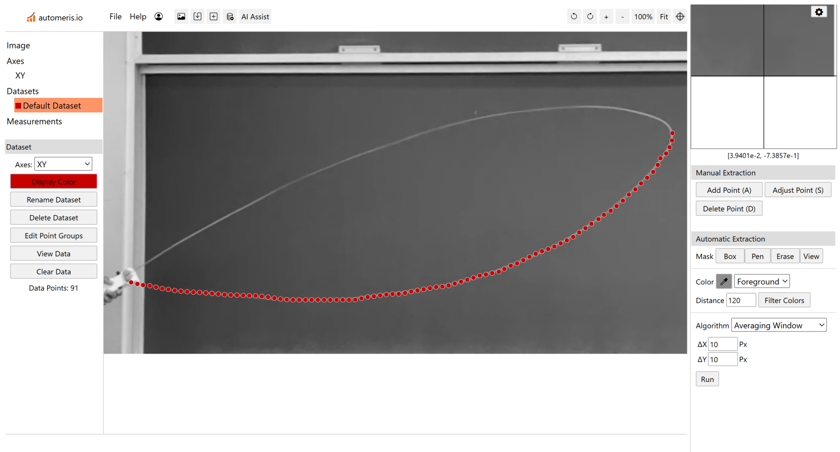
Click on the pen tool and choose the string as your reference colour

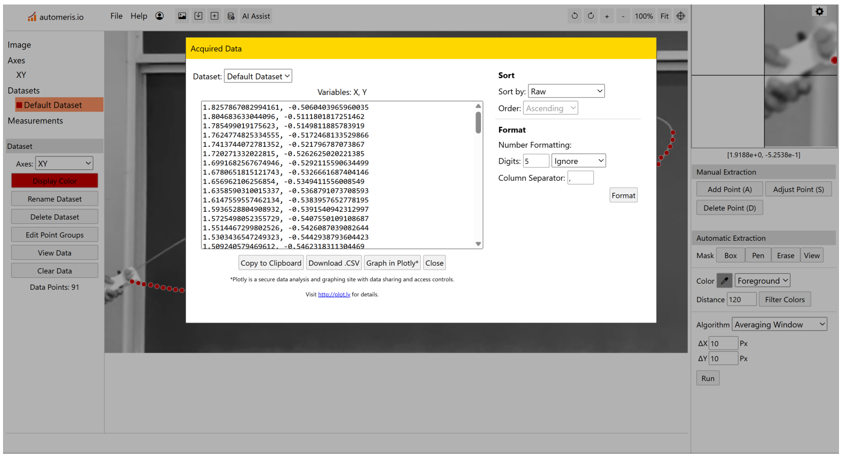




Using the pen tool, highlight one part of the string, we will be masking the lower and the upper parts separately.

Data points should appear as red dots on your image.

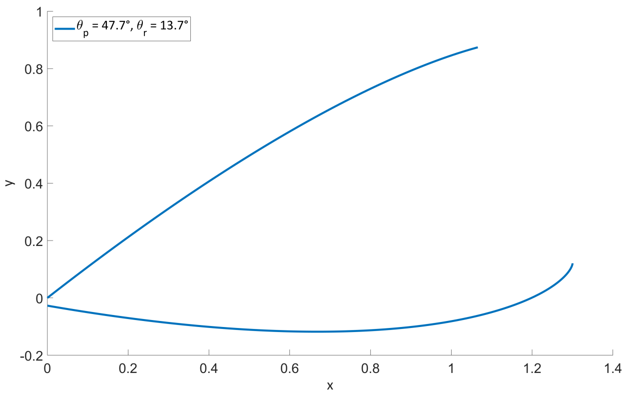




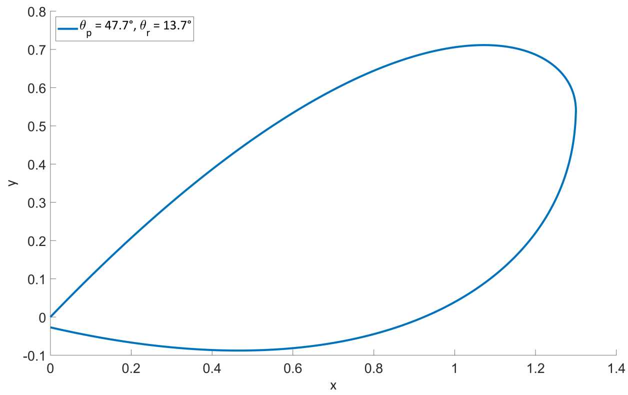
Click on view data and download your csv file.

Plotting and optimization

Input the measured quantities into the provided MATLAB code for plotting. Choose a value for the drag coefficient f, then run the code. The upper and lower parts of the loop might not align properly; this is because the selected drag coefficient may not match the current loop shape being plotted. Manually adjust the drag coefficient until the two ends align. Record this value of f; we will compare it to the optimized value obtained later.



$f = 0.025894\ \text{kg/s}^2$

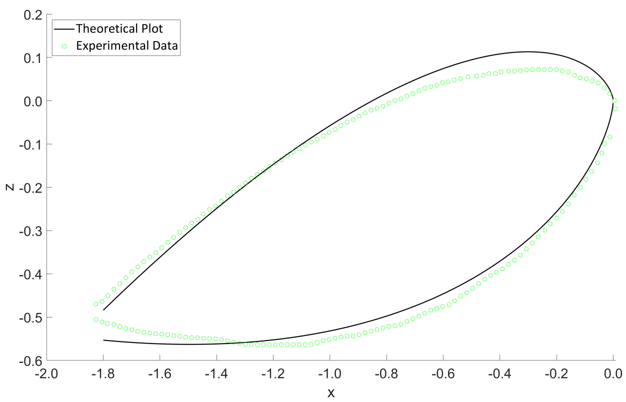


$f = 0.015894\ \text{kg/s}^2$

Once you have the data saved as CSV files from the previous section, you will need to import them into MATLAB. Import the data into four variables: x_{upper}, y_{upper}, x_{lower}, and y_{lower}, corresponding to the x- and y-coordinates of the upper and lower parts of the string loop. After importing the data, define the variables in the provided optimization code and run your program. The output will include the optimized values of x_+, x_-, and R. Use the following formula to calculate the drag coefficient:

$$R = \frac{\mu g}{f}.$$

This optimized value of f, though similar to our manual choice, should give a better plot.




```
<stopping criteria details>
Optimized x_plus: -0.30061
Optimized x_minus: -1.4949
Optimized R: 0.33445
>>
```

Output for optimization.