Data Analysis Exercise

Sun Yiu Samuel Wong

PHY424

January 22, 2020

1 Introduction

I analyzed the image "ellipses88.jpg". The sheet of printed paper from which I made the measurements is attached at the end.

2 Measurement

2.1 Technique

The tools I used are two identical straight rulers, made of metal, with a millimeter scale (the smallest ticks are 1 millimeter apart). The rulers are standard rectangular shape. This is important for the following measurement technique.

To make the measurements, I first made the assumption that the paper is a perfect rectangle, with perfectly straight edges, and the horizontal and vertical axis of the ellipses are perfectly aligned with the edges of the paper when they are printed. A quick examination of the paper attached makes it clear that this is a good assumption, and any error about this assumption is clearly negligible compared to the reading error that follows.

Next, to find the horizontal (or vertical) axis of an ellipse, I needed to align my ruler with two furthest points of the ellipse. I placed the first ruler (I call this the measuring ruler) at the approximate position. Then I take the second ruler (I call this the aligning ruler), align its shorter side with the side of the paper, so that it is perpendicular to the measuring ruler. I then press down on the aligning ruler to fix it on the paper, and slide the measuring ruler along it. This way, I can see the ellipse gradually getting more (or less) covered up by the ruler as I slide. When I see that the amount of ellipse that is covered up by the ruler stops increasing, and just before it starts decreasing, I must be the closest to having my measuring ruler aligned with the horizontal (or vertical) axis of the ellipse. Then I read off the measurement. This method requires that the rulers have perfect 90 degrees angle. Again, this is so accurate that any error is negligible compared to the reading error.

I repeated this measurement for a series of 12 ellipses as well as the calibration square.

2.2 Uncertainties

When I read off the reading on the ruler, I estimated the one standard deviation error coming from the human reading error and the blurriness of the edge of the ellipse on the printed paper. This uncertainties are mostly the same for the various ellipse, except for ellipse 1 and 2, whose y- dimensions are especially thin and make the corresponding errors larger.

3 Analysis

3.1 Initial Data Processing

For each ellipse, I have measured its x- and y- dimensions and estimated their measurement uncertainties. Let's denote these values as x_{meas}, dx_{meas}, y_{meas}, dy_{meas}. I also measured the dimensions (with uncertainties) of the calibration square, which converts the measurement unit to the unit used by the given equation. Let's denote these calibration measurements as c_x, dc_x, c_y, dc_y. Let's denote the calibration square in unit used by the equation as c'_x and c'_y, which are exact values known a priori.

The to find (x, y) in the equation unit, I make the calculation

$$x = x_{meas} \cdot s \cdot \frac{c'_x}{c_x} \tag{1}$$

$$y = y_{meas} \cdot s \cdot \frac{c'_y}{c_y} \tag{2}$$

where s is the scale (an exact value) associated with each ellipse. The uncertainty in x and y is a combination of the measurement uncertainty and the calibration uncertainty (which is a systematic uncertainty) via quadrature:

$$dx = s \cdot c'_x \cdot d\left(\frac{x_{meas}}{c_x}\right) \tag{3}$$

$$dx = s \cdot c'_x \cdot \sqrt{\frac{1}{c_x^2} dx^2 + \frac{x_{meas}^2}{c_x^4} dc_x^2} \tag{4}$$

and

$$dy = s \cdot c'_y \cdot \sqrt{\frac{1}{c_y^2} dy^2 + \frac{y_{meas}^2}{c_y^4} dc_y^2} \tag{5}$$

I made all of the above calculation in the module "process_data.py" (which also includes the raw measurement data) and copied and pasted all the result of x, dx, y, dy into the text file "ellipse_data".

3.2 Data Fitting

I first took the sample code "odr_fit_to_data.py" published on the course website. I coded in the three distribution functions of interest: Lorentizian, Log-Normal, and Absolute Sinc. Next, I fitted the data against each of the three fit function, and adjusted the guess parameters based on the program output.

After some trials, I found that if I use all the data, it approximately fits the absolute Sinc function, with the parameters:

$$y_\mu = 110 \pm 16, \mu = 24.0 \pm 0.6, \sigma = 4.1 \pm 0.2 \qquad \text{(tentative values)}$$

where the absolute Sinc function is given by

$$y(x) = y_\mu \left| \frac{sin(x-\mu)/\sigma}{(x-\mu)/\sigma} \right| \tag{6}$$

A plot of this fit is given below in figure 1. The χ^2 per degree of freedom is 9.18, which is a reasonable number. However, the corresponding CDF is 0%. In other words, the program is saying that it is impossible to get such a good fit.

I quickly found the cause of the problem. Looking at figure 1, it is clear that most of the data points lie relatively close to the fitted curve, with one exceptional outlier, which is the value with the largest x. The residual is a long horizontal line, which shows how far the point is from the curve. However, the algorithm clearly runs into a problem here. The point should be fitted to the second bump just below it, instead of fitting it to the central bump, which is much further to the left. This program fits the data using the

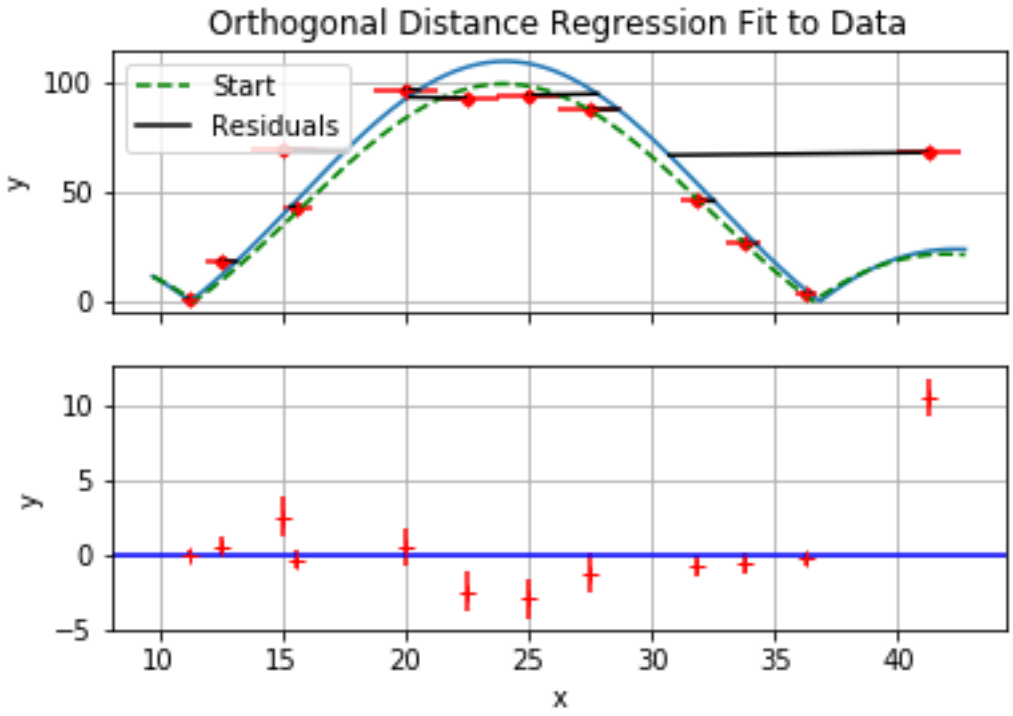

Figure 1: A fit using the absolute Sinc function

orthogonal distance regression, which tries to minimize each point to a "closest" point on the curve. It is know that this algorithm can mistaken which point on the curve is supposed to be the closest.

To remedy this problem, I reran the fitting program but took out the outlier data point. The new fit is shown in figure 2.

This time, my results are

$$y_\mu = 102 \pm 5, \mu = 23.7 \pm 0.2, \sigma = 4.0 \pm 0.1 \qquad \text{(outlier-adjusted values)}$$

The χ^2 per degree of freedom is 1.4, which is much smaller than before and indicates a good fit. The CDF is 18.6%, which meas there is a 18.6% chance that the χ^2 per degree of freedom we found is true. This is a reasonable result. From the plot, the fit is mostly a good fit.

The other two models, Lorentizian and Log-Normal, do not fit the data at all. The fitted parameters have uncertainties of 10^8; it is unneeded to show the corresponding plots to know that these models are not related to the data.

4 Conclusion

Overall, the fit is rather good after I took out the point that not only is an outlier relative to the rest of the data, but also a point that the orthogonal distance regression misses in finding the closest point of the curve. Based on the plots, the uncertainties I estimated were reasonable. If I had more time, I would use a different program to fit the data so the issue with orthogonal distance would not arise.

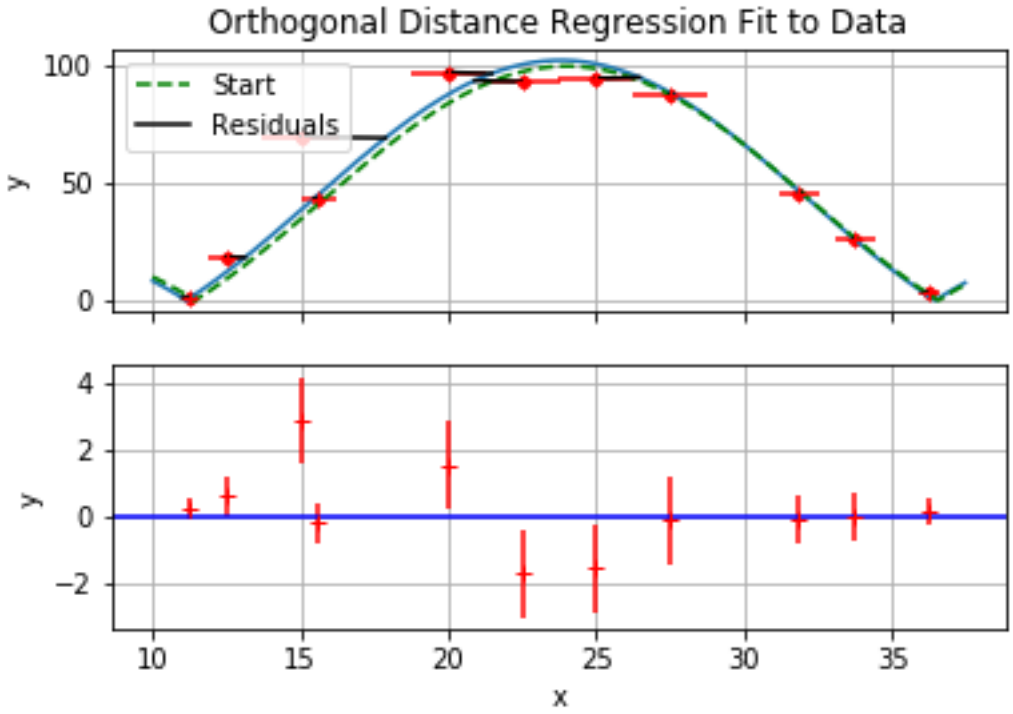

Figure 2: A fit using the absolute Sinc function, but with the outlier removed.