Iowa State University
Aerospace Engineering

AER E 322 Lab 01
Practice Experiment and Data Analysis

Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

February 3, 2023

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**

Contents

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**

Chapter 1

Pre-Lab

1.1.   Introduction

Aircraft wings undergo oscillations and other random forces while in flight. This lab replicates and analyzes some of the forces and oscillations a wing will experience in flight while also serving as an introduction to the PASCO tool kits and data processing. To simulate the wing, we used a cantilevered aluminum beam, and to generate and measure the oscillations, we used a PASCO tool kit—specifically the PASCO wave driver, displacement sensor and motion sensor. There were three rounds of testing; each additional round of testing introduced a new variable into the beam movement that changed the shape of the data. The data was collected using the PASCO tool kit and software provided. After the lab, we analyzed and processed the data in Python to how each variable effected the oscillation of the beam.

1.2.   Objectives

During this lab, our primary objectives were to:

1. Learn how to record data under dynamic conditions and analyze or post-process the data.

2. Observe approximately how a common aerospace structural material might respond to oscillatory forces.

3. Gain familiarity with the PASCO tool kit and the PASCO Capstone software.

1.3.   Hypothesis

1.3.1.   Test 1

We predict this test will provide the cleanest data of the three tests. Since the only force acting on the beam should be from the wave driver, we expect the displacement graph to shown a uniform and steady wave—matching the oscillations of the wave driver. The data from this test should closely match the oscillations of an airplane wing in very steady flight.

1.3.2.   Test 2

This test adds a spring with a hanging weight to the cantilevered beam. Due to the oscillations of the spring-weight system, we expect to see sudden highs and lows in the data corresponding with when the spring is in compression or tension respectively. The data from this test should demonstrate the oscillations of the wing in steady flight if there is an additional oscillatory or vibrational force simultaneously acting on the wing.

1.3.3.   Test 3

This test is similar to Test 2 (see Section 1.3.2) except a third significant force has been introduced. Due to the addition of arbitrary impulses being applied by hand to the free end of the beam, we expect the data to show large peaks and dips in the data correlated with the timing of the impulses. The data from this test should demonstrate the oscillations of real flight as described in Section 1.3.2 but also how the wing might react during periods of high turbulence where sudden, large impulses may act on the wing.

Chapter 2

Lab Work

2.1. Variables

2.1.1. Independent Variables

Wave Driver Frequency

The frequency of the wave generated by the PASCO wave driver.

Wave Driver Amplitude

The maximum voltage the wave driver will use when oscillating, proportional to the displacement of the wave driver arm.

Sampling Frequency

The time intervals at which the PASCO motion sensor or displacement sensor poll the position or displacement of the beam. Sensor with a higher sampling frequency will collect more data in the same amount of time.

Smoothing Span

The number of elements after a given element used to calculate a rolling mean while data processing. A small smoothing span will better preserve the shape of the data; whereas, a

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**

larger smoothing span will better reduce noise and outliers.

Time

Each test was run for exactly $15\,\text{s}$, measured in $0.01\,\text{s}$ intervals.

2.1.2.   Dependent Variables

Displacement/Position

The change in location of the cantilevered beam from its equilibrium position, measured closer to the free end along a perpendicular vertical axis.

Best Fit Curve

The line of best fit or fit curve is a normalized curve matching the raw or smoothed data and depends on the shape and magnitude of the displacement data.

2.2.   Work Assignments

Refer to Table 2.1 for the distribution of work during this lab.

2.3.   Materials

The materials required for this lab include: cantilever aluminum beam, spring, weight, Styrofoam pad, tape, round point-tip "needle" displacement sensor, C-clamps, wooden 2"x4", bench vice, string, computer with PASCO Capstone software, clamp stand with clamps, PASCO 850 Universal Interface, PASCO mechanical wave driver, and a PASCO non-contact motion sensor.

2.4.   Apparatus

Figure 2.1 shows the fully assembled lab apparatus. For Test 1, the weight was removed from the spring. For Tests 2 and 3, the weight was added back to the apparatus and the displacement sensor—the needle-like device on top of the beam—was removed.

Table 2.1: Work assignments for lab 01.

Task	Matthew	Peter	Natsuki
Lab Work			
Date Recording		X	
Exp. Setup	X		
Exp. Work			X
Exp. Clean-Up	X	X	X
Data Processing			
Data Import	X		
Smoothing	X	X	
Line of Best Fit	X	X	X
Report			
Introduction			X
Objectives		X	
Hypothesis	X		
Variables		X	
Materials		X	
Apparatus		X	
Procedures	X		
Data	X		
Analysis	X	X	X
Conclusion			X
References			X
Appendix			X
Revisions	X	X	X
Editing	X		

As shown in Figure 2.1, the left side of the beam is fixed to the board and the right side is attached to the wave driver with string. The rectangular, gray box in the foreground is the PASCO 850 Universal Interface, and the blue device in the midground is the motion sensor. The motion sensor must be positioned directly underneath the piece of white foam taped to the bottom of the beam with at least 15 cm between the sensor and the foam.

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**
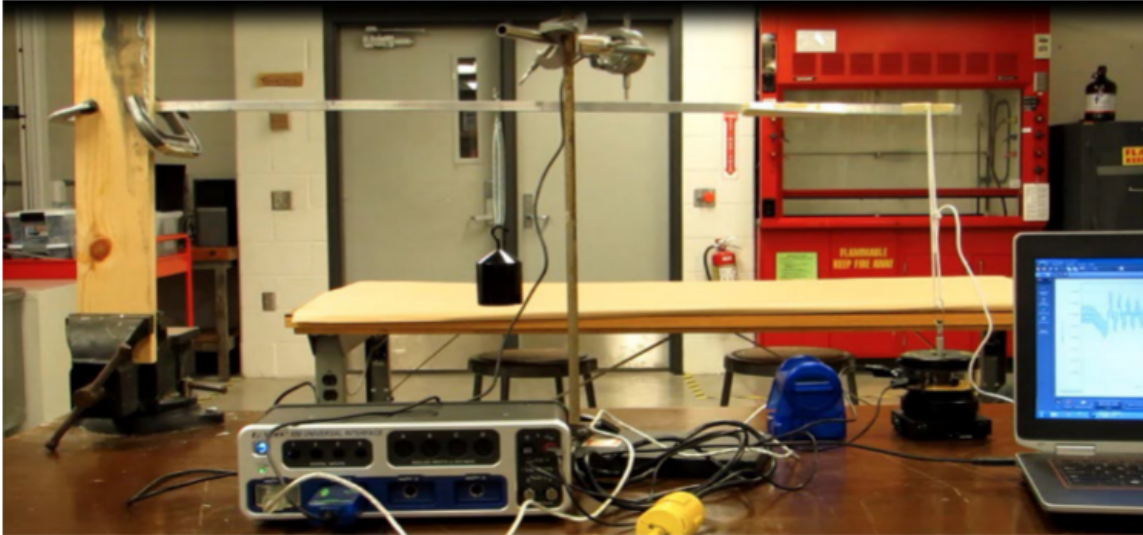
Figure 2.1: Lab apparatus completely set up, including the spring, weight, and displacement sensor. Image from the *Lab 1 Practice Experiment and Data Analysis* manual [1]. Copyright 2023 by Thomas Chiou.

2.5.   Procedures

2.5.1.   Setup

Assemble the apparatus as described in Section 2.4. Ensure that the styrofoam reflector plate is at least 15 cm away from the motion sensor before running any tests. Launch the PASCO Capstone software, and turn on the PASCO 850 Universal Interface. Check that both the "needle" displacement sensor and motion sensors are recognized by the software. Once the sensors are recognized, configure the software to display two graphs "by selecting the corresponding icon from the on-screen template" [1]. Next, click on each axis to label the graphs, and set the y-label of the displacement sensor graph to "displacement" and "position" for the motion sensor graph. For both graphs, the x-label will be "time". Refer to Figure 2.2 to see how the graph layout should look.

Set up the wave driver by clicking the "Signal Generator" icon on the left-side of the screen. Set the waveform to "Sine", sweep type to "Off", frequency to 10 Hz, amplitude to 10 V, and select "Auto" at the bottom of the window. For reference, compare your "Signal Generator"

Figure 2.2: The graph screen of the PASCO Capstone software fully configured for this lab. Image from the *Lab 1 Practice Experiment and Data Analysis* manual [1]. Copyright 2023 by Thomas Chiou.

window to Figure 2.3.

Next, at the bottom of the screen select "Recording Condition", then select "Stop Condition". Set "Condition Type" to "Time Based", and set the time to 15 s. The "Recording Condition" window will look similar to Figure 2.4.

At the bottom of the screen, select "Motion Sensor", and set the sampling frequency to 100 Hz. The sensor can be zeroed by selecting the icon to the right of the sampling frequency field. Use the same process to set up the displacement sensor, except set the sampling frequency to 5 Hz. To zero the displacement sensor, click the physical O/I button on the sensor. The sensors should be zeroed before each round of testing.

Finally, navigate through the following menus: "Data Summary" icon on the left column of the screen → Motion Sensor → Position (m) → select the gear symbol → Numerical Format → change Number Style to Significant Figures → set the Number of Significant Figures to 5. Follow the exact same process for the displacement sensor.

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**

Figure 2.3: The wave driver or "Signal Generator" configuration window. Image from the *Lab 1 Practice Experiment and Data Analysis* manual [1]. Copyright 2023 by Thomas Chiou.
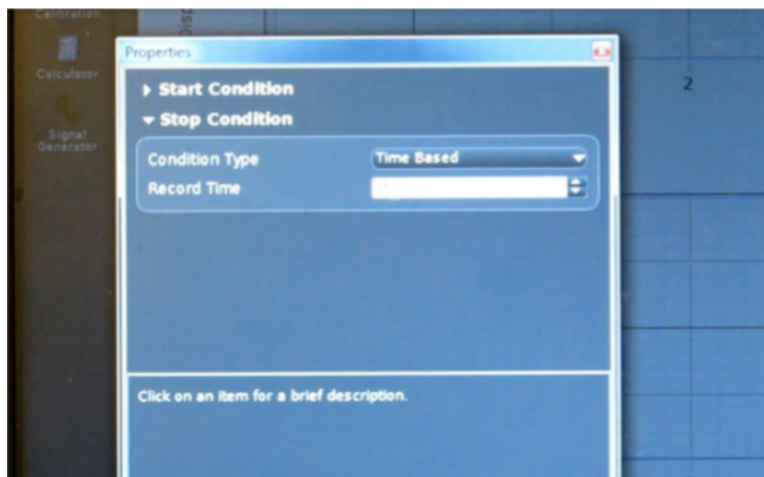

Figure 2.4: The "Stop Conditions" configuration window. Image from the *Lab 1 Practice Experiment and Data Analysis* manual [1]. Copyright 2023 by Thomas Chiou.

2.5.2.   Test 1

For Test 1, we will only utilize the wave driver. Take the hanging weight off the beam before starting and remain clear of the apparatus during operation. Ensure the displacement sensor is making good contact with the beam and approximately halfway depressed. Check the motion sensor is directly under the styrofoam reflector plate with at least 15 cm of buffer. The string that attaches the beam to the axle socket of the wave driver should be taut. Lastly, double-check that the connector tip of the wave driver "is firmly seated inside the axle socket" [1]. Zero both sensors before running the test, then, in the PASCO Capstone software, click "Record" to run the test. Let the test run for the entire 15 s duration.

2.5.3.   Test 2

Before beginning Test 2, remove the displacement sensor from the beam and add the hanging weight to the spring. Begin the test as described in Section 2.5.2. As the test begins, a group member will pull the weight down and release it, allowing the spring-weight system to oscillate for the duration of the test. Be careful not to pull the weight down too hard.

2.5.4.   Test 3

For Test 3, leave the apparatus as it was configured for Test 2 in Section 2.5.3. Start the test as described in Section 2.5.2. After beginning the test, in addition to pulling down and releasing the hanging weight, a group member will randomly tap the free end of the cantilevered beam up or down every 2 s to 3 s for the duration of the test.

2.5.5.   Cleanup

Using the "File" menu in the PASCO Capstone software, save the experiment data to a `.txt` or `.csv` file. Transfer the data to a flash drive for post-processing and analysis.

2.6.   Data

All data analysis was done with Python. The following packages were used for importing, smoothing, and processing the data:

- `pandas` (https://pandas.pydata.org/)

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**

- numpy (https://numpy.org/)

- matplotlib (https://matplotlib.org/)

- scipy (https://scipy.org/)

The data was in a .csv format which the pandas Python package had no difficulty importing and parsing. To smooth the data, we used the pandas.DataFrame.rolling() and pandas.DataFrame.mean() functions. The .rolling() function selects a subset of the elements, and the .mean() function calculates the mean of that subset. See Listing 2.1 for the function from our Python script that smoothed the data.

```python
# Smooths a DataFrame
# Input Args:
#    df: 1 col DataFrame
#    smooth: list of smoothing windows, e.g., [5, 3, 3]
def smooth_df(df, smooth):
    smoothed_data = df
    for win in smooth:
        # min_periods=1 stops the NaNs
        smoothed_data = smoothed_data.rolling(win, min_periods=1).mean()

    return smoothed_data
```

Listing 2.1: The smooth_df() function from our Python data processing script. This function smooths a pandas.DataFrame object once or in multiple passes as defined by the smooth argument.

To generate the line of best fit, we first observed that the data in our graphs looked generally sinusoidal. To generate and graph a sinusoidal line of best fit, our best option was to use the scipy.optimize() function, but since the frequency of the beams displacement curve varied due to the forces we were applying to it—especially in Test 2 and Test 3—the scipy.optimize() function struggled to accurately generate an appropriate line of best fit. Our final solution for generating the line of best fit involved using the scipy.optimize() function and the numpy.fft.fftfreq() and numpy.fft.fft() functions which performs a Fast Fourier Transform (FFT) on a numpy.array object. See Listing 2.2 for our implementation of the sinusoidal line of best fit function.

```python
# Uses a Fast Fourier Transform to generate an initial guess then uses a
    non-linear least squares algorithm to
# generate a matching sine curve
```

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**

```python
def sin_lobf(data, t, stats=False):
    # Fit sin to the input time sequence, and return fitting parameters "
    amp", "omega", "phase", "offset", "freq", "period" and "fitfunc"
    t = np.array(t)
    data = np.array(data)
    ff = np.fft.fftfreq(len(t), (t[1]-t[0]))   # assume uniform spacing
    Fyy = abs(np.fft.fft(data))
    guess_freq = abs(ff[np.argmax(Fyy[1:])+1])   # excluding the zero
    frequency "peak", which is related to offset
    guess_amp = np.std(data) * 2.**0.5
    guess_offset = np.mean(data)
    guess = np.array([guess_amp, 2.*np.pi*guess_freq, 0., guess_offset])

    def sinfunc(t, A, w, p, c):  return A * np.sin(w*t + p) + c
    popt, pcov = curve_fit(sinfunc, t, data, p0=guess) # this is the least
    squares bit
    A, w, p, c = popt
    f = w/(2.*np.pi)
    fitfunc = lambda t: A * np.sin(w*t + p) + c
    dict = {"amp": A, "omega": w, "phase": p, "offset": c, "freq": f, "
    period": 1./f, "fitfunc": fitfunc, "maxcov": np.max(pcov), "rawres": (
    guess,popt,pcov)}

    # Print some stats
    if stats:
        print("Amplitude: %g mm\nFrequency: %g Hz\nLOBF: f(t) = %gsin(%g *
    t + %g) + %g\n
    -----------------------------------------------------------" % (dict['
    amp'], dict['freq'], dict['amp'], dict['omega'], dict['phase'], dict['
    offset']))
    return dict
```

Listing 2.2: The `sin_lobf()` function from our Python data processing script. This function generates a sinusoidal line of best fit for data that is generally sinusoidal. It uses an FFT algorithm from the `scipy` package and an optimization function from the `numpy` package. The function was modified from unsym's post on StackOverflow [2].

We also tested using a piece-wise line of best fit generator, *i.e.*, the line of best was generated for smaller domains rather than the entire 0 s to 15 s of data. The graphs generated using this method can be found in Appendix A. The entirety of the data processing script is shown in Appendix B.

Aerospace Structures Laboratory Report
Lab 01 Practice Experiment and Data Analysis
Section 4 Group 2
Matthew Mehrtens, Peter Mikolitis, and Natsuki Oda

AER E 322   February 3, 2023   **Spring 2023**

The following sections discuss the data from Tests 1 through 3. For conciseness, only the most pertinent graphs are discussed below. All the graphs are included in Appendix A for reference.

2.6.1. Test 1

Shown in Figures 2.5 and 2.6 is the raw data and smoothed data respectively for Test 1. The raw data shows the measurements from the displacement sensor and the motion sensor. The smoothed data omits the raw data from the displacement sensor. For clarity, only the data from 5 s to 8 s is shown below.
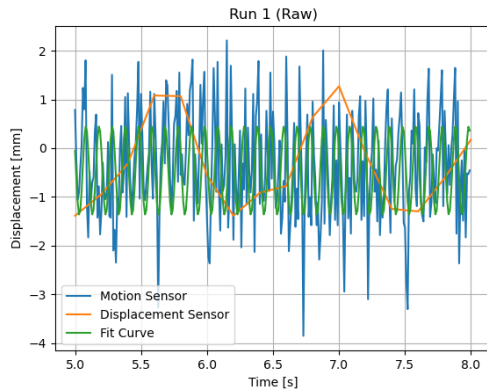
Figure 2.5: Raw data from Test 1, zoomed to only show 5 s to 8 s.
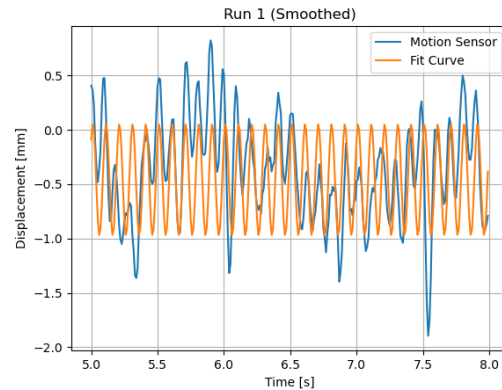
Figure 2.6: Smoothed data from Test 1, zoomed to only show 5 s to 8 s.

From these graphs, it is clear the displacement sensor's polling rate was too slow. In retrospect, the reason is clear. As shown from Figure 2.6 and as described in Section 2.5.1, the wave driver was forcing the beam to oscillate at a rate of approximately 10 Hz, but the polling rate of the displacement sensor was 5 Hz. In other words, the resolution of the displacement sensor was too low to accurately measure the true displacement of the beam. For this reason, the displacement sensor was omitted from the second and third tests.

2.6.2. Test 2

Shown in Figures 2.7 and 2.8 is the raw data and smoothed data respectively for Test 2. At the beginning of this run, a force was applied to a weight hanging from a spring connected to the beam, causing the weight to oscillate (see Section 2.4 and 2.5.1 for details). For clarity, only the data from 5 s to 8 s is shown below.
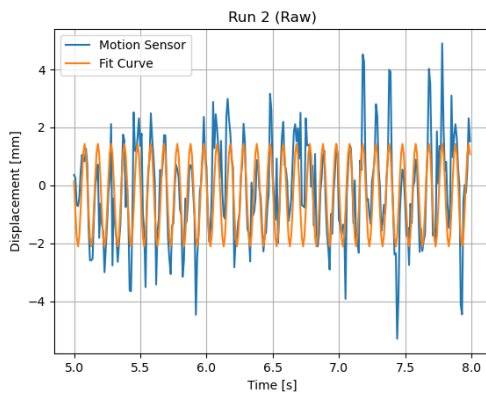

Figure 2.7: Raw data from Test 2, zoomed to only show 5 s to 8 s.
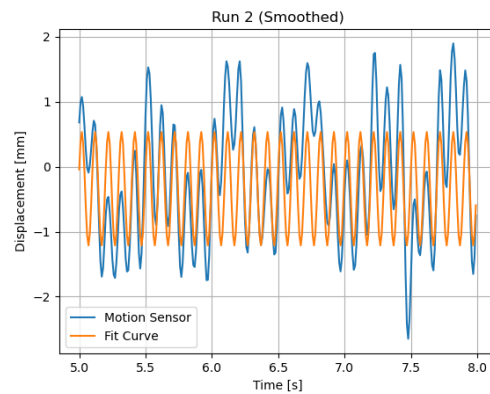
Figure 2.8: Smoothed data from Test 2, zoomed to only show 5 s to 8 s.

The line of best fit does not account for the secondary oscillations due to the hanging weight—it averages between the high and low secondary oscillations–but from looking at the motion sensor measurements shown in Figure 2.8, you can observe the periods of oscillation where the weight was applying a downwards force on the beam, *e.g.*, from 5.2 s to 5.5 s.

2.6.3. Test 3

Shown in Figures 2.9 and 2.10 is the raw data and smoothed data respectively for Test 3. At the beginning of this run, a force was applied to a weight hanging from a spring connected to the beam causing the weight to oscillate. At random times during the run, a member from our group applied an arbitrary impulse with their hand to the end of the beam (see Section 2.5.4 for details). For clarity, only the data from 5 s to 8 s is shown below.

We observe from Figures 2.9 and 2.10 the same secondary oscillations we observed in Test 2 (Section 2.6.2. In addition to the secondary oscillations, we also note the large spikes at
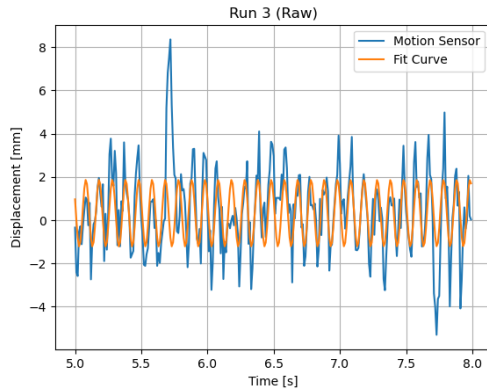
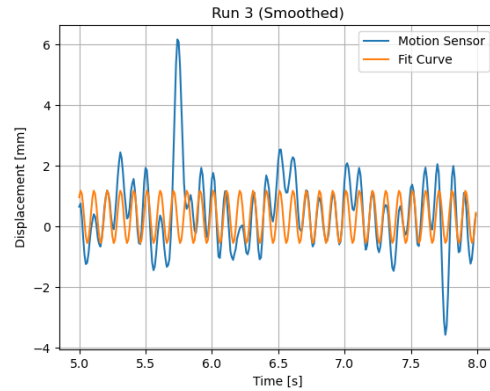Figure 2.9: Raw data from Test 3, zoomed to only show 5 s to 8 s.

Figure 2.10: Smoothed data from Test 3, zoomed to only show 5 s to 8 s.

5.75 s and 7.75 s. These spikes were due to the arbitrary impulses being applied to the beam by one of our group members.

Chapter 3

Conclusion

3.1.   Analysis

3.1.1.   Sources of Error

3.2.   Conclusion

References

[1]   Thomas Chiou. *Lab 1 Practice Experiment and Data Analysis*. Jan. 26, 2023, pp. 2–5.
[2]   unsym. *How do I fit a sine curve to my data with pylab and numpy?* Feb. 19, 2017. URL: https://stackoverflow.com/a/42322656 (visited on 02/03/2023).

Appendix A

Graphs

Appendix B

Code