Lab 5 Report

Landon Wang and Yinshi Liu

Contribution: Yinshi Liu wrote the entirety of question 1. Landon Wang wrote the entirety of question 2 and 3. This report is completed by Yinshi Liu.

Q1a

Pseudocode

1. Import functions, and define constants from Lab3. ($x_0 = 1$, $x_c = \frac{c}{2\sqrt{3}}$, m = 1, k = 12)
2. Define initial conditions and set up Euler-Cromer process:
   Setup equations of motion:
$$m\frac{d^2x}{dt^2} = -kx\left(1-\frac{v^2}{c^2}\right)^{\frac{3}{2}} \tag{1.1}$$
$$\frac{dx}{dt} = v \tag{1.2}$$
$$\frac{dv}{dt} = -\frac{k}{m}x\left(1-\frac{v^2}{c^2}\right)^{\frac{3}{2}} \tag{1.3}$$
   rewrite equations of motions for the Euler-Cromer process:
$$x_{i+1} = x_i + v_i\Delta t \tag{1.4}$$
$$v_{i+1} = v_i - \frac{k}{m}x_i\left(1-\frac{v_i^2}{c^2}\right)^{\frac{3}{2}} \tag{1.5}$$
   initialize loop.
   While the time remaining is not zero, update x values and v values using equation 1.4 and 1.5. Append them to their respective arrays.
   Repeat this process for each value of x_0.
3. Plot the results for each x_0.

Code Output

(See next page)

Figure 1.1

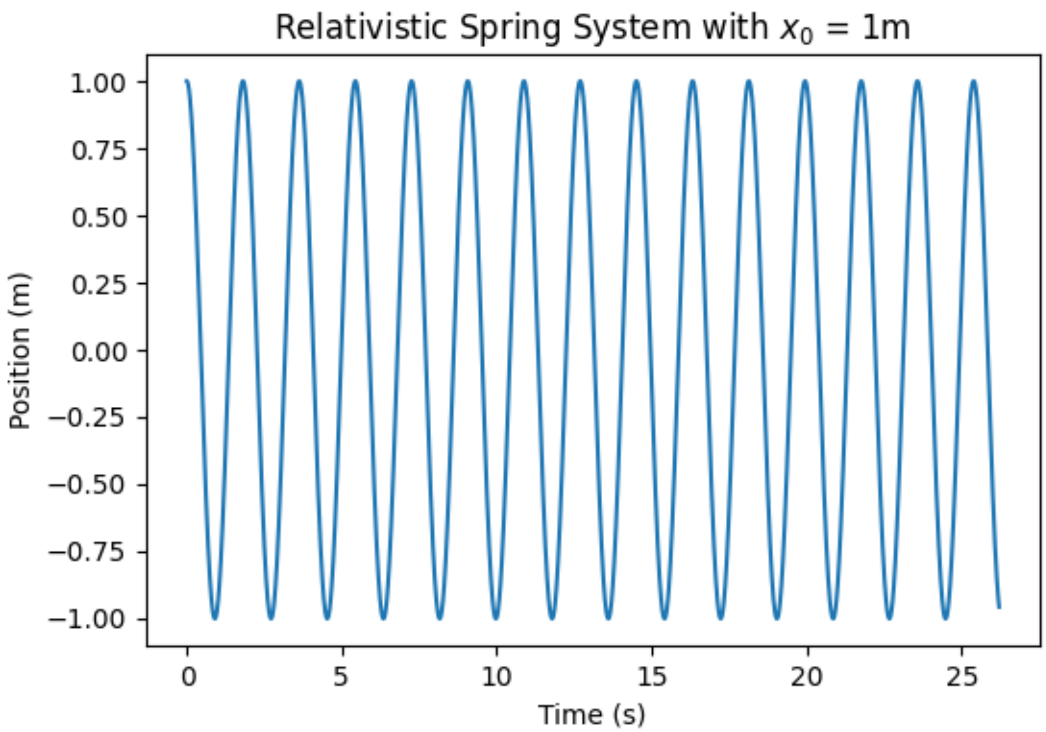


Figure 1.2

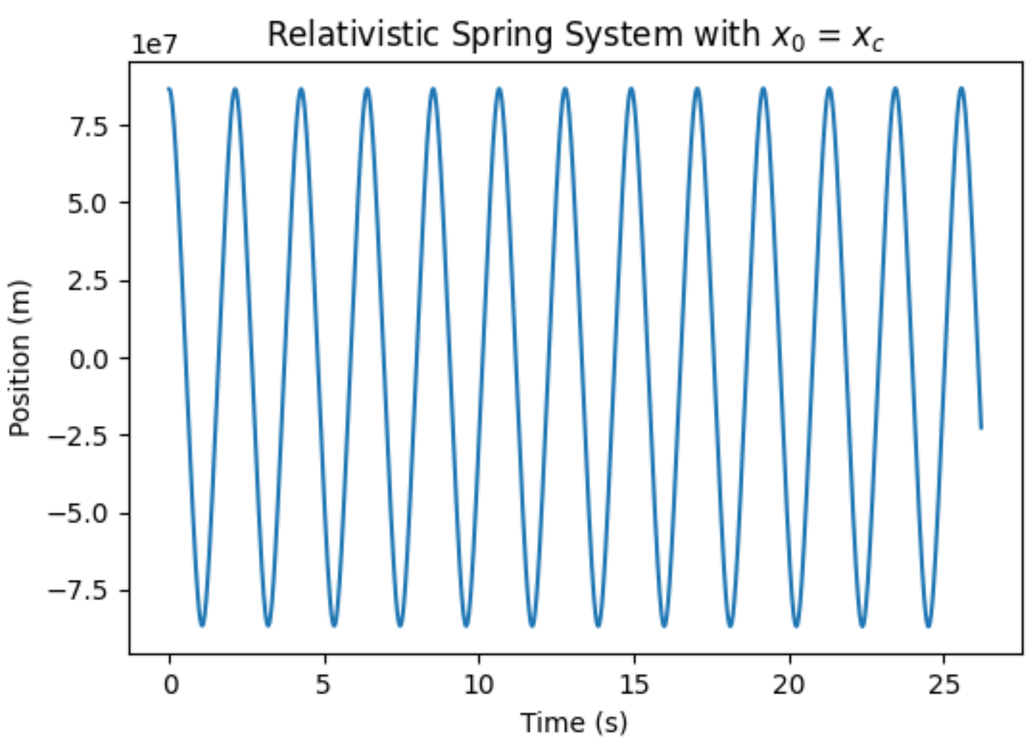
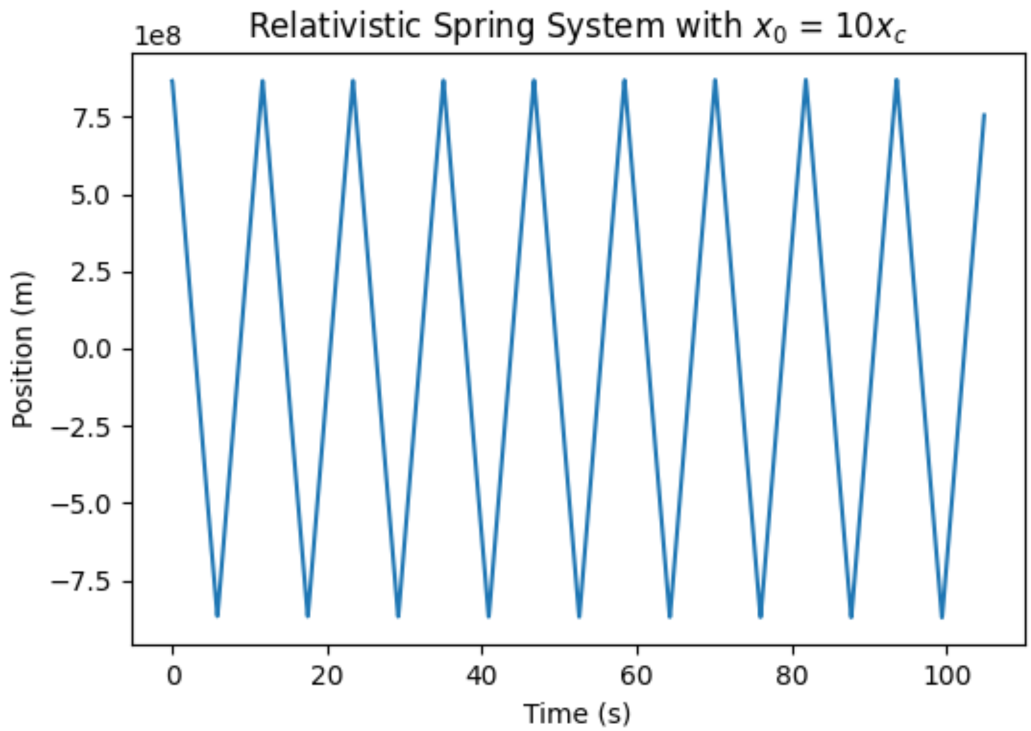


Figure 1.3



The parameters chosen for this part is based on the conditions of Lab 3 Q2, where mass m=1kg, and spring constant k = 12N/m. The initial value x_c is based on the solution of the nonrelativistic system. ($x_c = 8.66 \times 10^7$m), and the length of each array corresponds to a power of 2 (2^{18}, 2^{18}, 2^{20} respectively) in order to have a better result while using FFT. As the plot has shown, as the initial displacement increases, the movement of the particle over time resembles less and less like a sine way, especially when x_0 approaches the relativistic regime.

Q1b

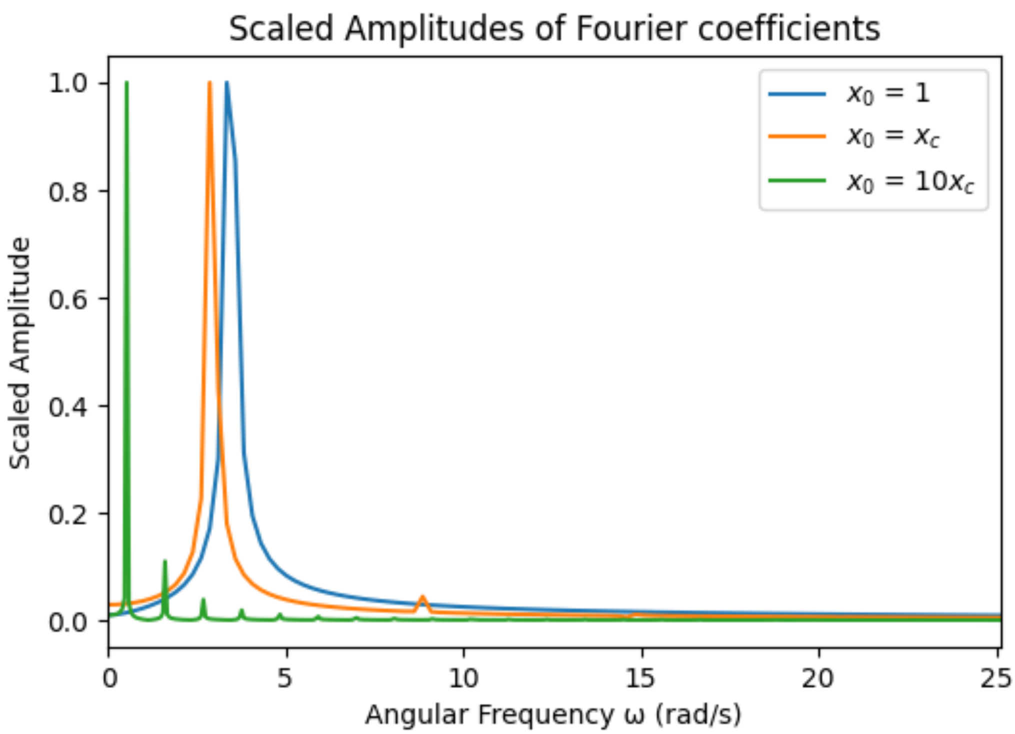
Pseudocode

1. Compute the Fourier transform of position arrays for each case using the fast Fourier transform function numpy.fft.rfft().
2. Scale the amplitude of $\hat{x}$ using the formula:
$$\hat{x} = \frac{|\hat{x}|}{|\hat{x}|_{max}} \tag{1.6}$$
3. Setup time arrays and convert it into angular frequency
4. Plot the angular frequency array against the scaled array.

Code Output

(See next page)

Figure 1.4



According to Figure 1.4, the spectrum of $x_0 = 1$ case has 1 deformed peak, which closely resembles the case of a pure sine wave. This agrees with the theoretical case when x_0 is small, where the correction term approaches zero and the classical case is recovered. However, it is known that correction case is not exactly zero, which means that the function of x against time is not exactly a pure sine wave, hence the peak appears to be deformed. One possible interpretation for this is that these are 2 frequencies located very close, but not identical with each other.

The spectrum for $x_0 = x_c$ has 2 distinct peaks on the graph. This means that the wave produced in this is not a sine wave, but the combination of those different sine waves with different frequencies, which agrees with our observation.

The spectrum for $x_0 = 10x_c$ is composed of many distinct peaks. This means that the wave is made of many different waves with frequencies outlined on the graph. Given that the function of x vs. t is distinctly not sinusoidal, this agrees with the answer to part a.

Q1c

Pseudocode

1. Import functions used in Lab 3, including gaussxw() from Newman, and g(x), t(x) depicting the period of the function:

$$g(x) = c\left\{\frac{k(x_0^2 - x^2)\left[2mc^2 + \frac{k(x_0^2 - x^2)}{2}\right]}{2\left[mc^2 + \frac{k(x_0^2 - x^2)}{2}\right]^2}\right\}^{\frac{1}{2}} \tag{1.7}$$

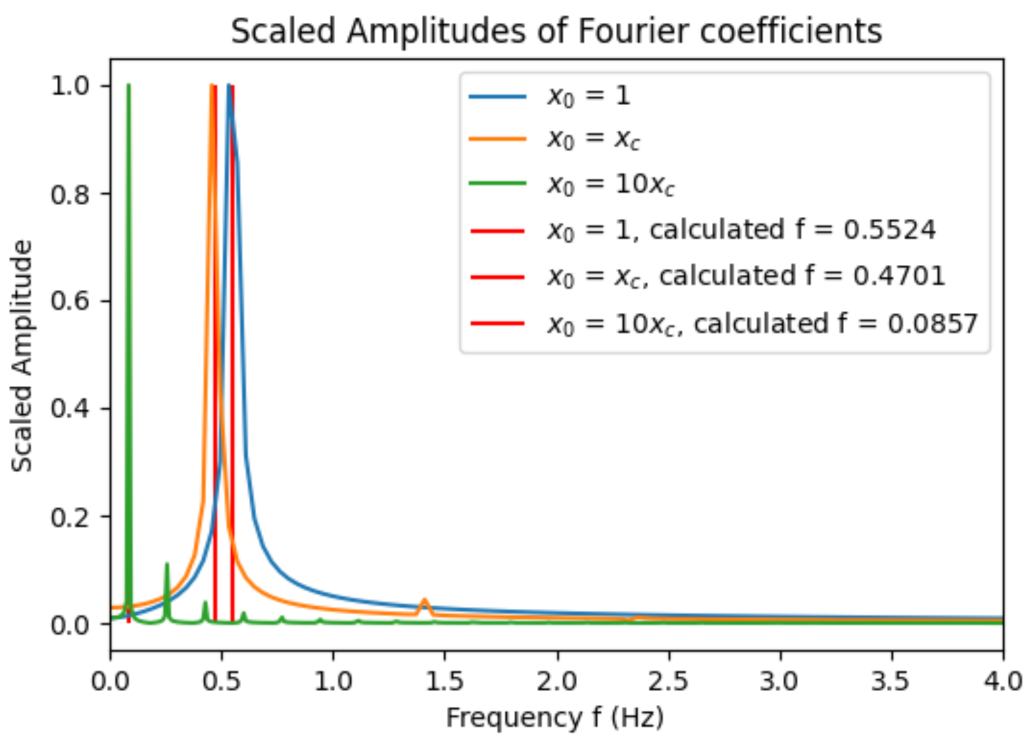
$$t(x) = \frac{4}{g(x)} \tag{1.8}$$

$$T = 4\int_0^{x_0} \frac{dx'}{g(x')} = \int_0^{x_0} t(x) \tag{1.9}$$

2. Define constant $a = 0, b_1 = x_0, b_2 = x_c, b_3 = 10x_c$, and calculate x, w, xp, wp for each case using the gaussxw() function. Set the step N for integration.
3. For each step i of N, calculate the result:
$$T_i = wp_i \cdot t(xp_i) \tag{1.10}$$
and add the result for each step.
4. Calculate the frequency for each case using:
$$f = \frac{1}{T} \tag{1.11}$$
5. Since the calculated frequency is in Hz rather than angular frequency (rad/s), convert the plot from part b to Hz. Plot the frequencies as vertical lines on the graph.

Code output

Figure 1.5



The red vertical lines represent the calculated frequency using Gaussian quadrature. As the graph has shown, the calculated frequency closely matched with the main 'spikes' for each case, which represents the simulated frequency of the motion of the particle. However, it appears that the match is not perfect. The calculated frequency appears slightly to the right of the simulated ones, which means that calculated periods are slightly smaller than simulated periods. This is expected due to the nature of the integral, which will become more accurate as the number of steps (N) gets larger.

Q2

Fourier analysis is commonly used in the audio treatments. In general, audio engineers that produces an album will use a series of different signal processing tools including a lowpass filter. Although the lowpass filter is commonly a circuit based unit that directly attached to recoding equipment, in a DSP dominated low cost audio industry, a digital lowpass filter applied after the analogue signal converted to digital wave is very commonly used.

Fourier transformation transform a wave into frequency space, which we can further processing it. For this question, fft and ifft from numpy will be used for its fast calculation time for a rather large data sample.

We will make a lowpass filter for 880Hz, that is for frequency space:

$$\hat{s}_{lp@880Hz}(f) = \begin{cases} 0, & f > 880 \\ \hat{s}(f), & f \leq 880 \end{cases} \qquad 2.1$$

The equation of Fourier transformation follows that

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x} dx \qquad 2.2$$

And the inverse transformation

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{i2\pi\xi x} d\xi \qquad 2.3$$

Noticed the 2π factor in the exponential, this means the result function $\hat{f}$ has its x axis been angular frequency instead of frequency.

This creates needs for change in unit for plot the transformed equation in to frequency space and the filtered frequency has to be unit changed. The relation between angular frequency and frequency follows that

$$\omega = 2\pi f \qquad 2.4$$

Pseudocode:

For (b)

1. Import the audio data using the provided code. Separate the two channels into two datasets
2. Creates a time sequence using np.arange for the length of each data set long and divide by sample rate to give a second united time sequence
3. Plot the wave plot using subplots (figure 2.1)

For (c)

4. Plot a focused wave plot for 1500 samples starting for the 4^{th} second (figure 2.2)

For (d)

5. Apply the Fourier transformation
6. Plot the frequency domain for the original wave (figure 2.3)
7. Set lowpass limit at 880Hz and unit transfer as equation 2.4
8. Apply the filtering as equation 2.1, to do that, set the transformed dataset indices from 880 to n-880 to zero and leave rest unchanged.
9. Plot the frequency domain plot for filtered data (figure 2.4, x axis limit to 1000Hz)
10. Reverse Fourier transformation
11. Plot filtered wave plot (figure 2.5)
12. Plot filtered focused wave plot (figure 2.6)

For (e)

13. Output data into wav file using provided code

Figure 2.1

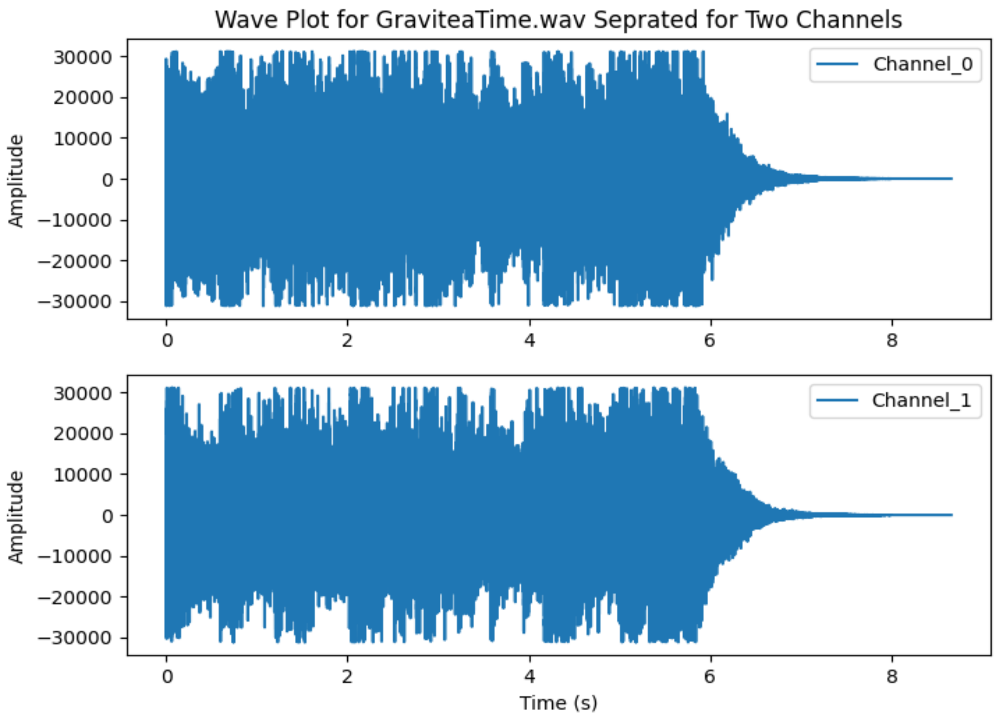


Figure 2.2

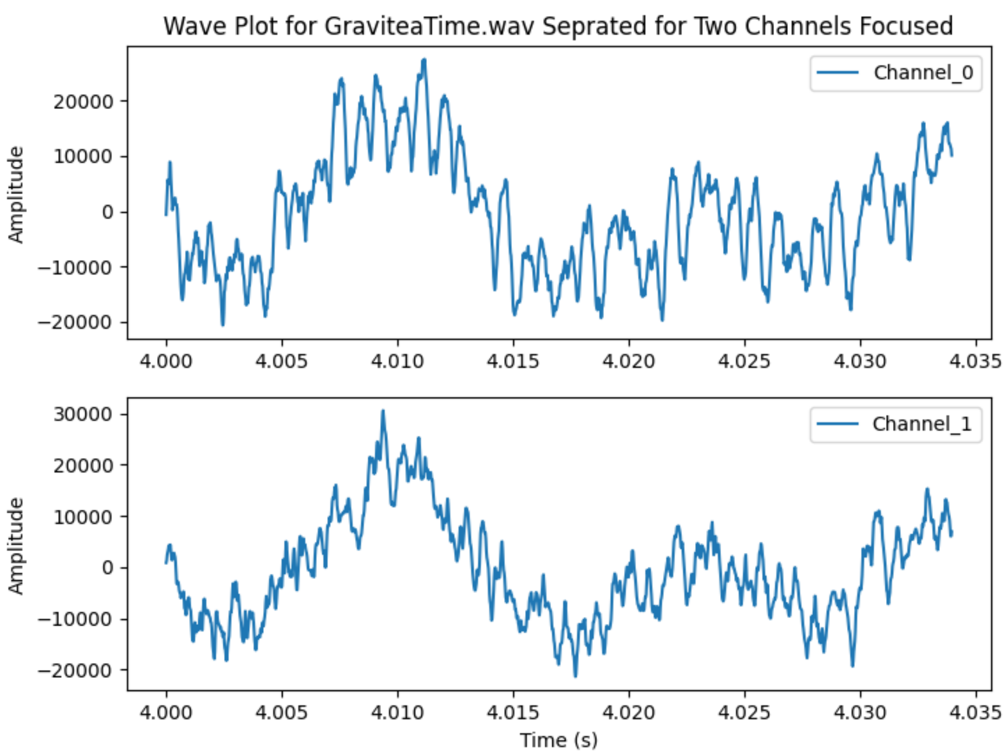


Figure 2.3

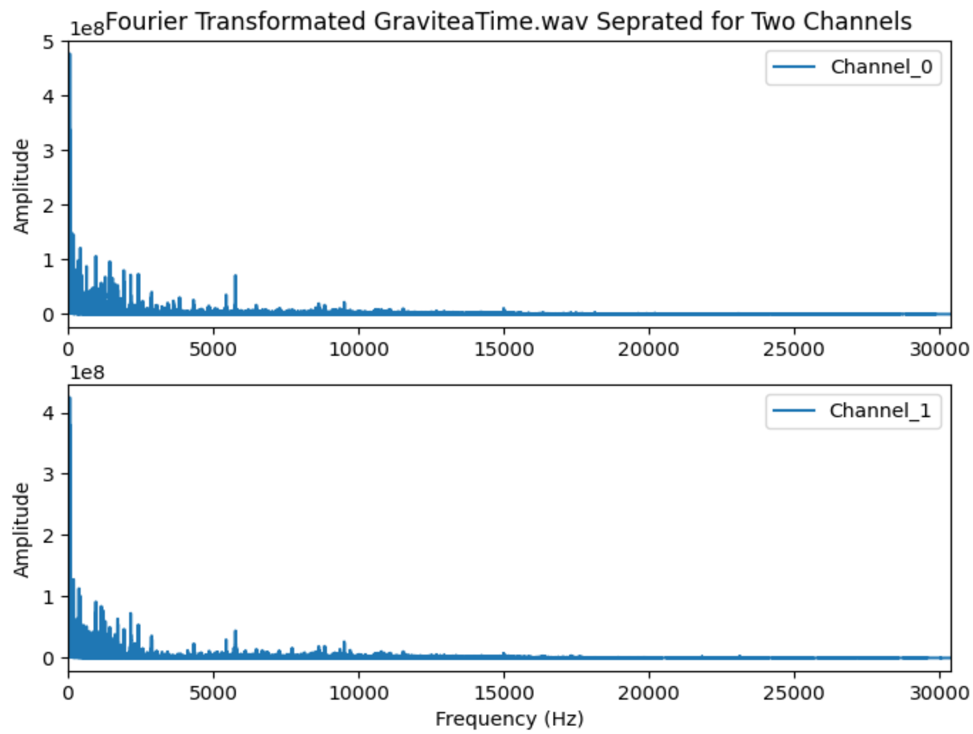


Figure 2.4

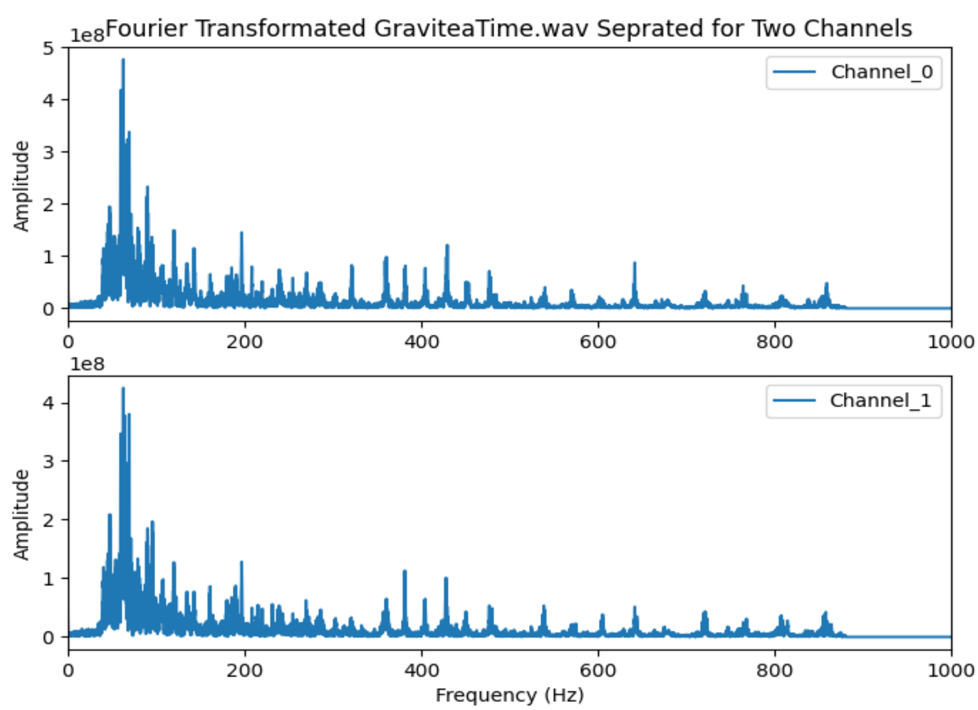


Figure 2.5

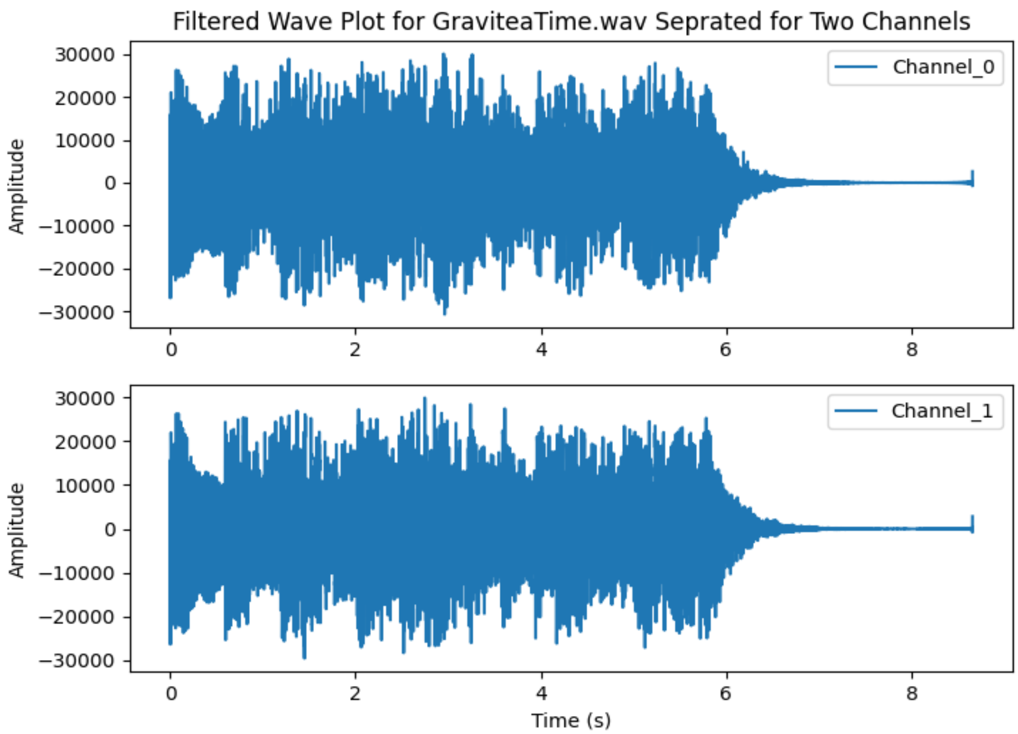
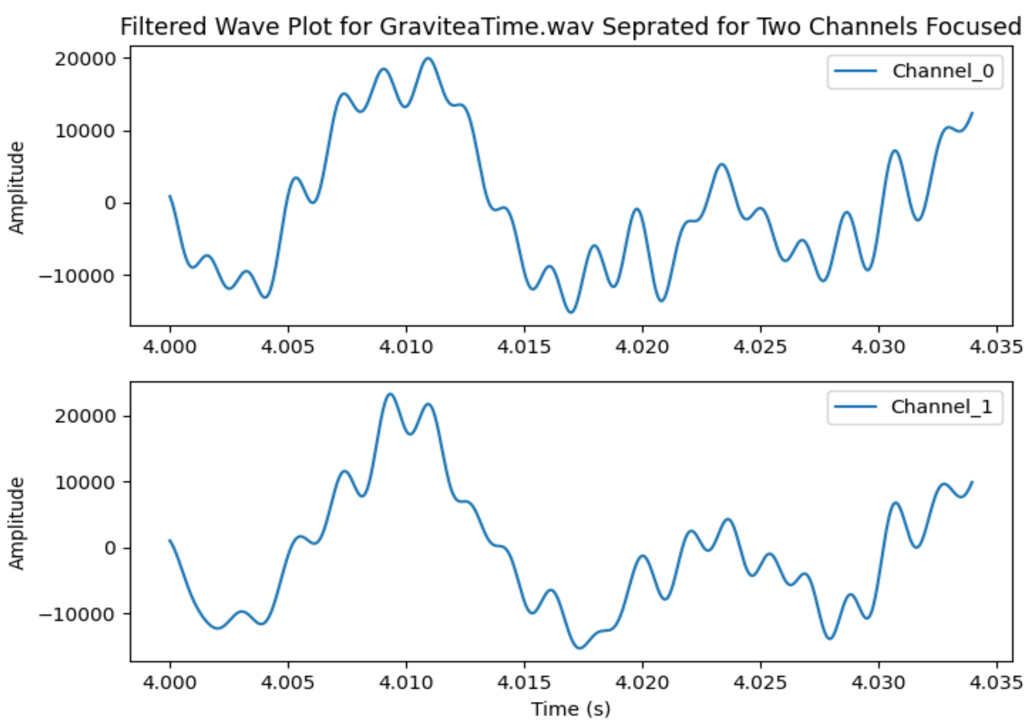


Figure 2.6



It can be clearly observed by compare the figure 2.2 and 2.6, the focused wave plot is much smoother for the filtered result, and form the comparison between figure 2.1 and 2.5, we observed lower overall amplitudes on the wave plot which means energy have been reduced as expected.

We noticed a boom sound at the end of filtered audio sample which is due to the error of Fourier and the error of filtration and its reverse transformation. We noticed that for lower lowpass frequency, the stronger the boom will be. But for any actual audio work, lowpass filter is generally used to remove some ultra high frequency due to signal interference. For such work, the problem of having the boom at the end is very minimal and almost unnoticeable.

Q3

From the lab manual, we learned that sea level pressure data can be decomposed into a series of waves of the form

$$SPL_m = A(t)\cos\big(m\lambda + \phi(t)\big) \tag{3.1}$$

Therefore, the SPL function should be

$$SPL(t,\lambda) = \sum_m A(t)\cos\big(m\lambda + \phi(t)\big) \tag{3.2}$$

Obviously, we have two variables, which is a problem. However, we are trying to decompose the given SPL data into simplified terms. As the question asked, we are interested for the cases m = 3 and m = 5. This can be done fairly easily using the Fourier transformation given fixed t

$$\begin{aligned} SPL(t_0,\lambda) &= \sum_m A(t_0)\cos\big(m\lambda + \phi(t_0)\big) \\ &= \sum_m K\cos(m\lambda + c) \end{aligned} \tag{3.3}$$

Where K and c are constants. We notice this form is capable for Fourier transformation with respect to λ, which is rather unusual, but certainly doable.

After application of Fourier, we can than filter the components of m = 3 and m = 5 separately, then convert back to wave space.

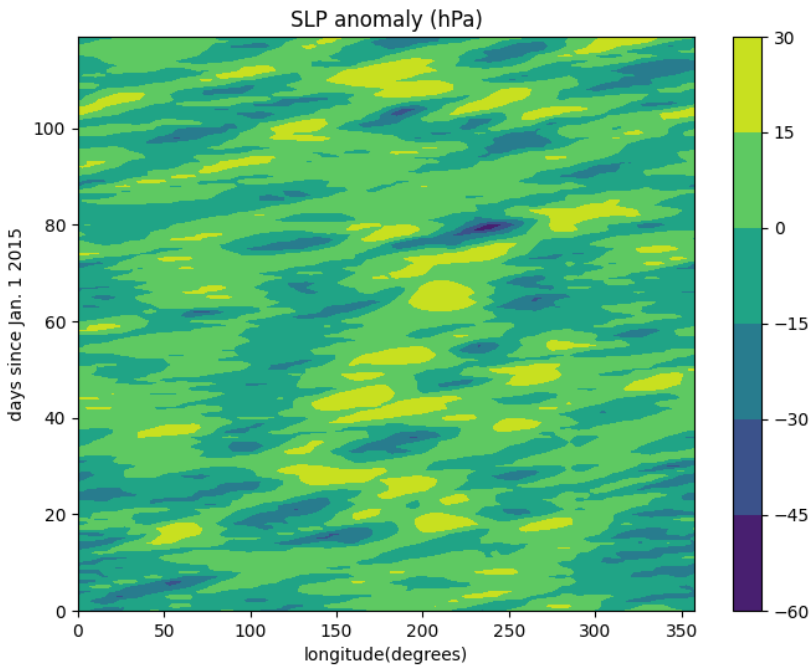
(a)

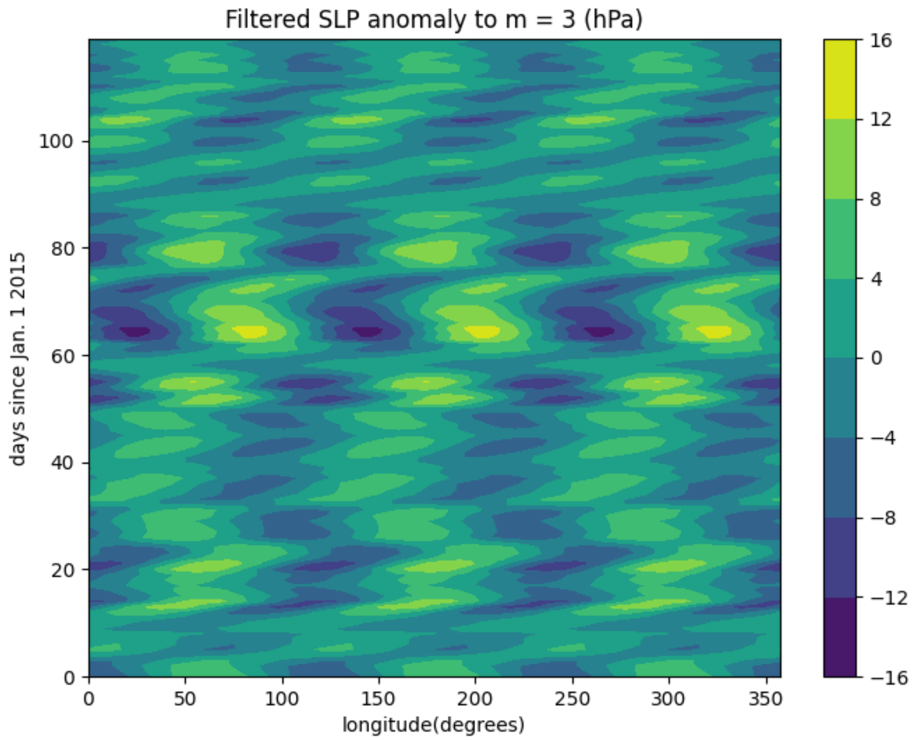
Pseudocode:

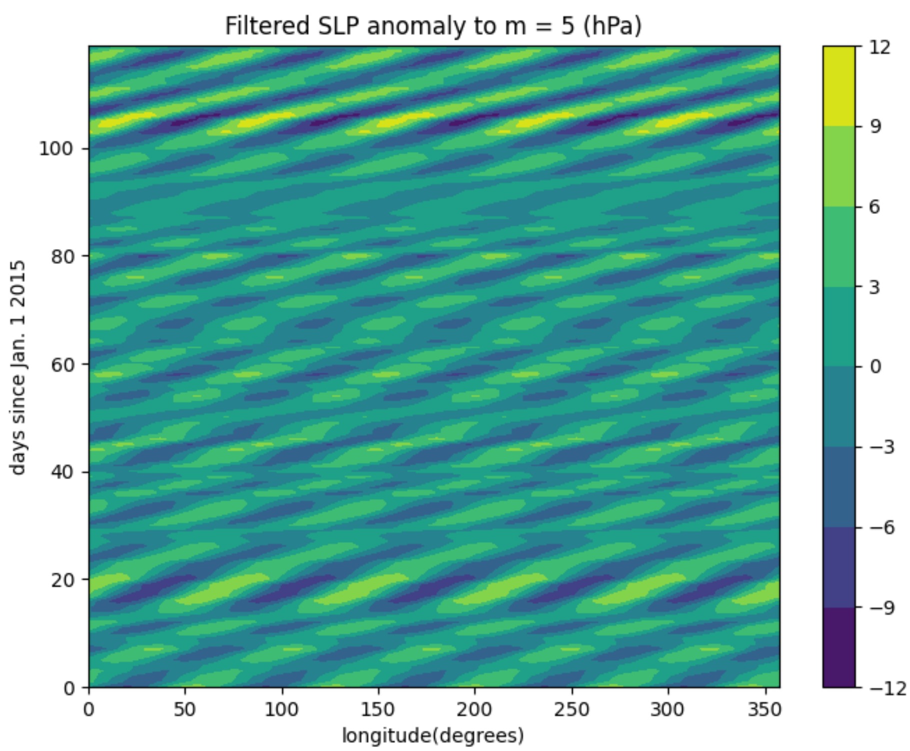
1. Import data following the provided code
2. Plot the original SLP plot using provided code (figure 3.1)
3. Create two empty sets of data in the same shape as the SLP data provided
4. Use the method described earlier in equation 3.3, fix the date variable by looping over all dates
5. Extract SPL data for the specific date and temporarily store for further use
6. Apply the Fourier transformation
7. Create a zero data set for filtered frequency space values
8. Filter the frequency space data by assign the respect data from transformed SPL to the new data set (for both indices ±3 and ±5)
9. Reverse transforms the frequency space data back to wave space
10. Store the date based data into the empty set created in step 3

11. Plot the result like in figure 3.1 (fig. 3.2 for m=3, fig. 3.3 for m=5)

Code Output
Figure 3.1-3.3




Filtered SLP anomaly to m = 3 (hPa)
days since Jan. 1 2015
longitude(degrees)
0
20
40
60
80
100
0
50
100
150
200
250
300
350
16
12
8
4
0
−4
−8
−12
−16


Filtered SLP anomaly to m = 5 (hPa)
days since Jan. 1 2015
longitude(degrees)
0
20
40
60
80
100
0
50
100
150
200
250
300
350
12
9
6
3
0
−3
−6
−9
−12

(b)

We would like to point out that the plot, although is a plane, is a cut of a cylinder. That is, the right most part of the data to the left most data has the same separation as any other two adjacent data points. This means we should not count the peaks and valleys of plot that goes around multiple times.

From the figure 3.2, we can clearly observe three peaks and three valleys for a specific date sample that extract m=3 out. This means the m from the equation 3.2 is in fact the number of full wavelengths of longitudinal wave has (independent of time).

Similarly, the m=5 case shown in figure 3.3 indicates 5 full wavelength for a circumference of the earth.

If we track a peak of the wave in figure 3.2 from day 0, we observed that the same peak (because the earth is round) will come back to to approximately the same longitude approximately 20 days. We noticed that a very fast movement at around day 17 almost made us loose the track, but it is apparently correct (compared to much slower wave progression on m=5 case, we apparently only have a faster than normal propagation, not a reverse direction of the wind) for the 20 days period.

In contrast, at about 20 days, the peak we tracked for m=5 case in figure 3.3 only travels around 180° in longitude, which is significantly slower compared to m=3 wave. This consist with the postulate in the lab manual.

From the plots, it is obvious that wave propagates differently at each day. Apparently, in general, the summer winds blows eastwards based on the data, which we have not experienced anyway. From day 40 to day 80, it is apparently that wave propagation rate is periodically increased and decreased, which could indicate large scale whirlwinds. At about day 88, there seems to be a revered wind direction with very fast east winds prior and after that day suggest it could be some extreme fast wind speed that moves the m=5 wave almost one wavelength east in a single day. (it is totally possible its just a west wind though) Lastly, starting from the 100th day, which could be the start of winter, the wave propagation rate very consistent until the end of the dataset.