Digital Filtering of Tidal Data*

Chenyuan Qu[1]

[1]Physics and Astronomy, University of Southampton
[*]BSc report for PHYS6017

ABSTRACT

Tide is a significant subject in Ocean-dynamics. The velocity of the ocean current involves the tidal data but with high-frequency interference information.[1] Therefore, we need use filter the interference to get useful tidal data. There are several different ways to do so. In this project, the Discrete Fourier Transform is used, and then a digital filter is used.

Key points: Ocean dynamic; digital filter; Discrete Fourier Transforms

Introduction

Tide is one of the fundamental and essential modules in ocean dynamics. Observing it is very significant for the development of ocean resource, infrastructure development of ocean disaster management, ocean military activity and so on.[1]
While the observation technology is increasingly mature, the techniques to observe ocean activities are more diverse and accurate. The tidal data is observed by mechanical floats and recorders in the past. However, nowadays, the main tidal observation techniques are advanced acoustics and electronics.[2] Because the velocity of the ocean current observed is instantly, and the ocean is affected by the Earth itself, the Sun, the Moon, wind, earthquake and many factors, so it involves massive data. The data can be separated by data of tidal variations and data of other activity[3]. The data involves the wave also because it is instant. As we need to research tides, only the tidal data is interesting, so others are interference. Hence, we need to remove this interference by using the filter for accessing the relatively pure data for tidal research.

Theoretical Background

The variation of velocity of the ocean current is mainly because of tide and wave. So we can express it as Equation (1).

$$v(t) = T(t) + w(t), \tag{1}$$

where v is the velocity of the current, T is the tidal factor, and w is the wave factor.
The variations of tidal data are relatively stable, which are with long periods and low frequencies, while wave data is short period and high frequency.
After considering the different frequency range of wave and tide, we can use a low-pass filter to remove the wave data and keep the tide data. The common ways to filter contains using moving average, the Discrete Fourier Transform, and wavelet transform. Among them, Discrete Fourier Transform is the most suitable way for computing calculation, so we used it to do the filter.

Theoretical Method

Fourier Transforms (FT) depends on a function f(t) is

$$X(f) = \sum_{n=0}^{N-1} x(t) exp(i2\pi ft) \tag{2}$$

Where X(f) is the spectrum of x(t), f is frequency. After wave filtering, we have

$$y(t) = \sum_{n=0}^{N-1} \omega(t-\tau)x(\tau) \tag{3}$$

Now, we can see

$$Y(f) = W(f)X(f) \tag{4}$$

Where Y(f) and W(f) is the spectrum of y(t) and ω(t) respectively// if ω_1(t) is an ideal lowpass filter, its spectrum will be

$$W_1(f) = \begin{cases} 1 & \text{when } |f| > f_c \\ 0 & \text{when } |f| < f_c \end{cases} \tag{5}$$

f_c is the cutoff frequency. We can derive ω_1 by using equations above.

$$\omega_1(t) = \sum_{n=0}^{N-1} W_1(f) exp(-i2\pi ft) = \sum_{n=0}^{N-1} exp(-i2\pi ft) \tag{6}$$

Or,

$$\omega_1(t) = \frac{1}{\pi t} sin(2\pi f_c t) \tag{7}$$

The sin function $\omega_1(t)$ reaches its maximum $2f_c$ when $t \to 0$. It becomes 0 as $t = \frac{1}{2f_c}$ then increases with $|t|$.
Discrete fourier transform, DFT, can be seen as complex form of Fourier series. Its length range is finite, but for discretely function as its name. The function is equation (8).

$$\bar{f_p} = \sum_{n=0}^{N-1} f_n exp(i\omega_p t_n) = \sum_{n=0}^{N-1} f_n exp(i2\pi pn/N), \tag{8}$$

where $\bar{f_p}$ is the spectrum of f_n, ω_p is angular frequency. $w_p t_n = (\frac{2\pi p}{N\Delta t})(n\Delta t)$ is applied.
The backwards equation is:

$$f_n = \frac{1}{N} \sum_{p=-N/2+1}^{N/2} \bar{f_p} exp(-i2\pi pn/N). \tag{9}$$

Experimental Method

The data provided is a data set of the velocities of the ocean current which is recorded every 14 minutes over a 10-day period. We can read and plot the data first as Figure 1.

As we said before, the variation of the ocean current mainly contains the tidal data and wave data, while the wave data is always high-frequency. Thus, we can use Fourier transform to analyze the data and then apply a filter on it. Because the velocity data is discrete, discrete Fourier Transform is applied. The x-axis should be frequency instead of time. Here, we used $hour^{-1}$ as unit for readability (Figure 2). It is obvious that the Fourier amplitude is relatively stable beyond the frequency of $1\,hour^{-1}$ and it changes rapidly between 0 and $1\,hour^{-1}$. Let's zoom in on the graph to the $0-1\,hour^{-1}$ part. We can see some peaks when the frequencies is $0.161\,hour^{-1}$,$0.0805\,hour^{-1}$, $0.0417\,hour^{-1}$ and $2.97*10^{-3}\,hour^{-1}$ respectively. So the corresponding time range is 6.2 hours, 12 hours and 25 mins, 24 hours and 336 hours (Figure 3).

We can see the main variation in the data is the 12 hours and 25 mins' variation, which is caused by the semi-diurnal tide. We need to remove this so the lower frequency variation can be seen. We need to apply a filter to it. We use this filter to filter data twice, once starting at the beginning and once starting at the end, to remove the variation of the semi-diurnal tide. This model can avoid the transient, so the beginning part of the record will not affect the result. I chose a filter with the order of 5 and set the critical frequency be 0.16, which makes a balance between sensitivity and the length of the transient. The data without that variation is as Figure 4

From Figure 3, we can see there is another significant peak which is the peak of 6.2 hours. If we want to research this, the lower frequencies need to be removed. At the same time, a low pass also need to be applied to remove the noise. We called this filtering model the band-pass. After choosing reasonable parameters, the Fourier graph would be as Figure 5

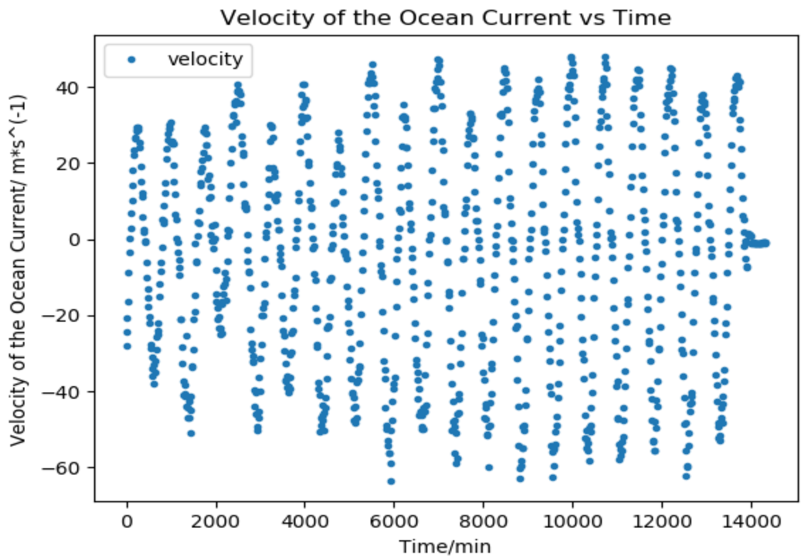

Figure 1. The graph shows the raw data between velocity of the ocean current and time. Because the data is discrete (recorded every 14 minutes), every data shows as one spot.

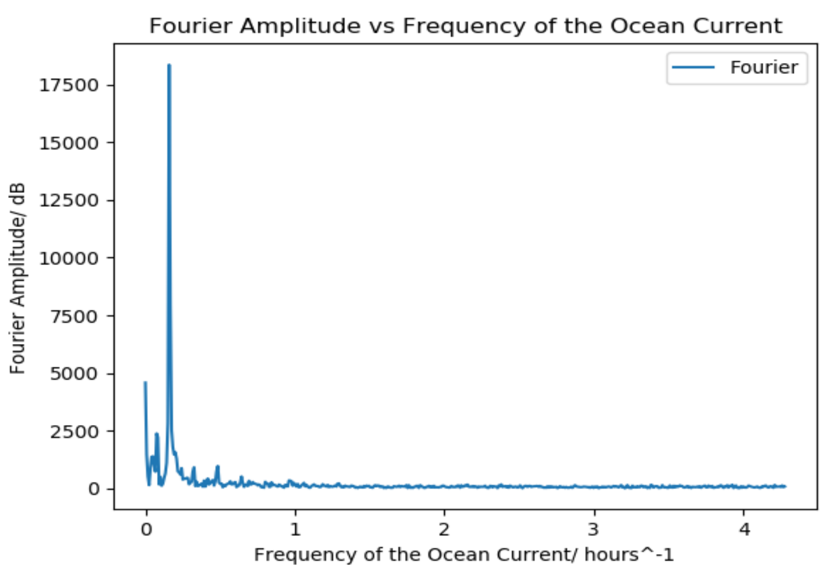

Figure 2. The graph shows the Fourier Amplitude of the raw velocity data against the frequency of the ocean current(unit: $hour^{-1}$).

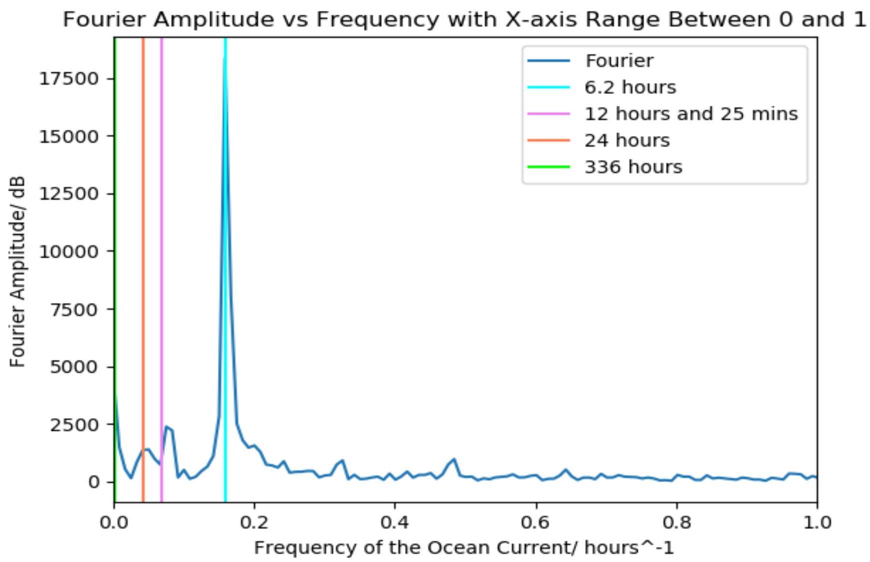

Figure 3. The graph shows $0-1hour^{-1}$ part of the Fourier Amplitude of the raw velocity data against the frequency of the ocean current. The blue line is the data after Fourier transform, the aqua, violent, orange and lime lines are the vertical lines when time is 6.2 hours, 12 hours and 25 mins, 24 hours and 336 hours.

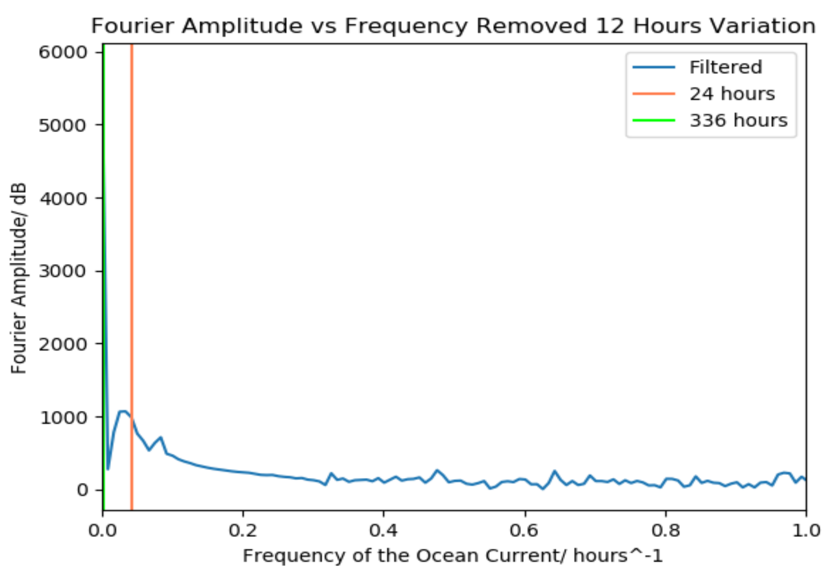

Figure 4. The graph shows the Fourier Amplitude after the variation of 12.4 hours has been removed against the frequency of the ocean current.

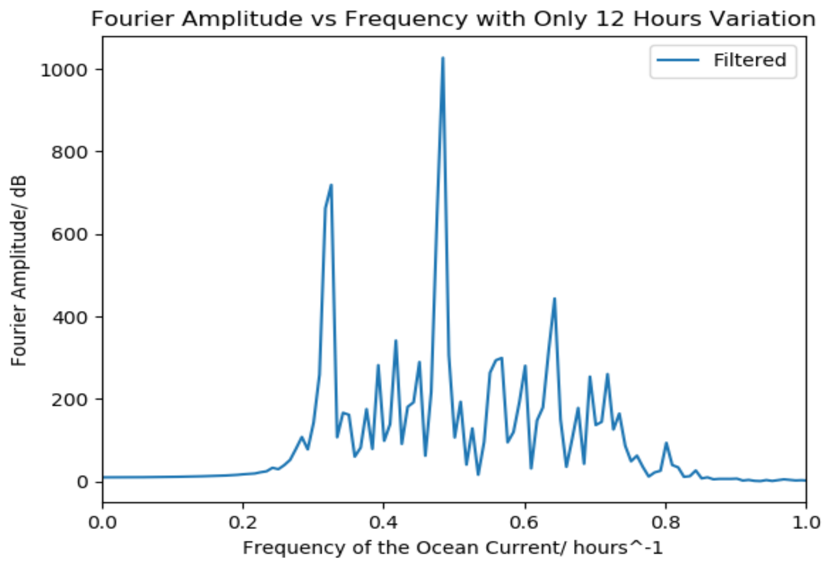

Figure 5. The graph shows the Fourier Amplitude only with the variation of 6.2 hours against the frequency of the ocean current.

Discussion of Results

From Figure 1, we can see that the velocity of the ocean current against time shows basically sinusoidal waves but with different amplitudes on different cycles, so we can deduce that the velocity of the ocean current is not only affected by one force resources. Therefore, we can use Fourier Transform to do further analysis. So we got Figure 2.
As I mentioned before in Theoretical Background, tide and wave are the main factors for causing the variation of velocity of the ocean current, while wave data is always with high frequency and tidal data is with low frequency. Because we only want the tidal data, the frequency after $1hours^{-1}$ can be ignored firstly, so we can get Figure 3. This graph has some significant peaks at the very start of the frequency, which has been already marked in the graph with vertical lines. The highest peak is the peak of 6.2 hours, which we will talk about later. The peak before it is the peak of 12 hours and 25 minutes, which is a type of tide, we called it *the semi-diurnal tide*[5].
To see the peaks of lower frequency more clearly and remove the interaction from this tide and data of higher frequency, we need to apply a low-pass filter on the data to remove any frequency after the semi-diurnal tide. The graph will be like Figure 4. From the graph, we can see that the 24 hours peak and 336 hours peak are still existed. Meanwhile, the peak of 336 hours is at about $5800dB$ and the peak of 24 hours is at about $1000dB$. Comparing with the amplitude before the removal, the amplitude of 335 hours increase (was about $4500dB$), and that of 24 hours decreases (was about $1350dB$), which means that the tide cycle of 24 hours would be destructed by the tide cycle of 12 hours and that of 14 days would be constructed. The peak of 24 hours is called the diurnal tide and the peak of 14 days is mixed, semi-diurnal tide[6].
Now, let us focus on the peak of 6.2 hours, which domains the trend of velocities of the ocean current. The graph is highest in the middle and decreases when the frequency moves to two sides, which are the properties of the Gaussian line. So we can fit the data with a Gauss function. Figure 6 is plotted. The fitting is generally fitting the data, which fits well especially on both sides. Thus, it is harmonic with the period of 6.2 hours, which is called the Shallow water overtides of principal lunar[7].

The variation in 24 hours can prove that the tide is affected by the Moon and the Sun. During only 10 days. the distance between the Earth and the Moon can be seen as constant. Nevertheless, the Earth makes a full rotation every 24 hours. Thus in the same place of the Earth, the distance between there and the Moon and the Sun changes, so the force from those two planets will change followed. As we know, tides are caused by gravitational force, so this type of tide is formed. The variation in 12 hours and 25 minutes is called semi-diurnal. It is also affected by the Moon. The cause of formation is also because the Earth make self-rotation, but at the same time, the Earth is also attracted by the Moon, so it also rotates to the Moon. The centre of mass of the Earth-Moon system is inside the Earth, so the centrifugal force always has the opposite direction to the gravitational forces. The schematic diagram is as Figure 7. The side closer to the Moon has not only a lower centrifugal force

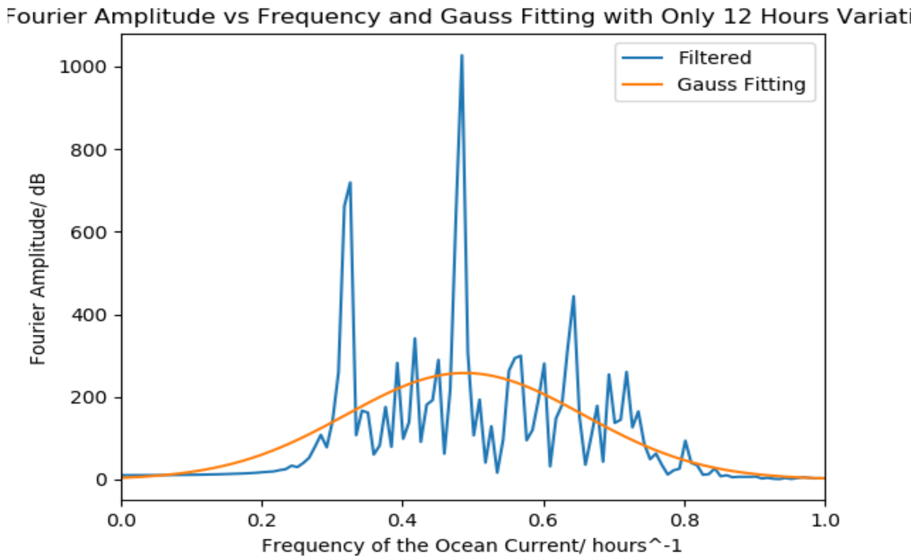

Figure 6. The graph shows the Fourier Amplitude only with the variation of 6.2 hours against the frequency of the ocean current and the fitting line to the data with Gauss function. The blue line is the data and the orange line is the Gauss fitting

because the distance between there and the mass centre of the Earth-Moon system is closer, but also a greater gravitational force. The direction of gravitational force is to the moon, so the sea will rise. Meanwhile, the side farthest to the Moon has a higher centrifugal force and a lower gravitational force. The direction of centrifugal force is to the outside of the Earth, so the sea rises again. Overall, for every rotation of the Earth, the sea rises twice, in the other words, there are two tides in about one day, so the period is 12 hours and 25 minutes. The difference between this period and half of a day (12 hours) is caused by the fact which is that the Earth and the Moon change their position during one rotation. The mixed tide of semi-diurnal and diurnal tides has the period of 14 days, which is half of the lunar month. During one lunar month, the lunar rotates the Earth for one cycle. So it is the Analogue of the semi-diurnal, but this time, the position of the Moon relative to the Earth does change. Again, the tides will appear twice in one lunar month, so one tide has the period of 14 days. The Shallow water overtides of principal lunar has a harmonic wave with higher frequency. The amplitude is fitting the Gauss function. Because it also affected by semi-diurnal and diurnal tides, but its range is around 6.2 hours.[9]

Conclusion

- The velocities of the ocean current contain the data from four types of tides, the mixed tide cycle - 14 days, the Diurnal tide cycle - 24 hours, the Semidiurnal tide cycle - 12 hours, and the Shallow water overtides of principal lunar - 6.2 days. The shallow water overtides of principal lunar affect the velocities most, and then mixed tide cycle. The diurnal tide and semi-diurnal tide affect the least.
- The tidal data is all with low frequency and affects the velocities of the ocean current most. The main variation in the data is the semi-diurnal tide. After we removed this tide from the data, the amplitude of the mixed tide increases a lot, so the mixed tide should destruct by the semi-diurnal tide. Meanwhile, the amplitude of the diurnal tides decreases to some extend, which means the semi-diurnal tide should construct with the mixed tide.
- The peak of 6.2 hours can fit by a Gaussian function, so it is a harmonic wave. It is also caused by the semi-diurnal and the diurnal tide with the time range of one cycle is 6.2 hours.

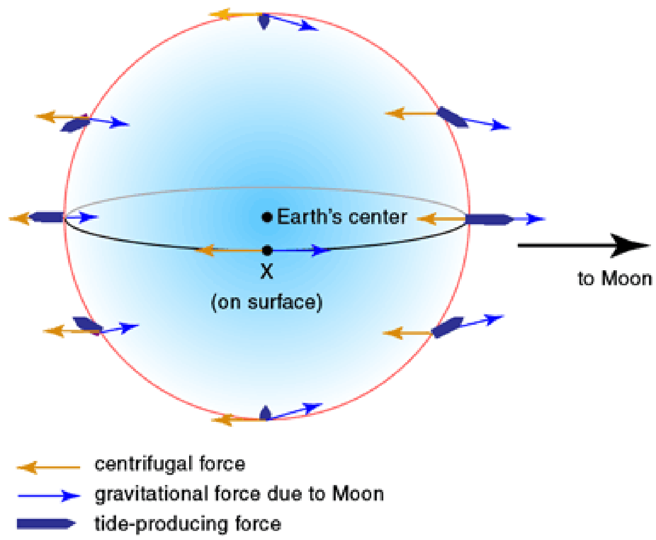

Figure 7. This graph shows the main forces acting on the water on the surface of the Earth, and the direction of the resulting force is the direction of the tides.

References

1. Ross, D.A.
   Introduction to Oceanography (1995)
   New York, NY: HarperCollins. pp. 236-242.
2. National Ocean Services
   Tides and Water Levels
   https://oceanservice.noaa.gov/education/tutorial_tides/tides11_newmeasure.html
3. Thurman, H.V.
   Oceanography,Introductory Oceanography, seventh edition (1994) New York, NY: Macmillan. pp. 252-276
4. Smith, Steven W.
   "Chapter 8: The Discrete Fourier Transform".
   The Scientist and Engineer's Guide to Digital Signal Processing (Second ed.)(1999)
   San Diego, Calif.: California Technical Publishing.
   ISBN 978-0-9660176-3-2.
5. Tidal Cycles on Earth - Diurnal Tides and Semidiurnal Tides (2020) https://beltoforion.de/en/tides/tidal_cycles.php
6. Donald E. Simanek
   "Tidal Misconceptions" (2003)
   Webseite
7. HåvardV indenes, Kjell ArildOrvik, Henrik Søiland, Henning Wehde
   Analysis of tidal currents in the North Sea from shipboard acoustic Doppler current profiler data (2018)
   https://doi.org/10.1016/j.csr.2018.04.001
8. Josep L. Pelegrí. Researcher at the Physical Oceanography Department. Institute of Marine Sciences, CSIC, Barcelona
   Why Are Tides 6 Hours Long if the Sun and Moon Have 24-Hour Cycles? (2014)
   https://metode.org/metodes-whys-and-wherefores/why-are-tides-6-hours-long-if-the-sun-and-html
9. NASA
   Ocean Motion and Surface Currents - Types of Tides Background (2005)
   Webseite

Appendix

```python
import numpy as np
from matplotlib import pyplot as plt
from scipy import signal, optimize

"""
READ ME

This program did the analysis on the velocities of the ocean current in the 'ocean.dat' file.

Please put the 'ocean.dat' file on the same folder with this script file
"""


# Function Part
# Functional functions
def read_data():
    """
    Read the velocity data from 'ocean.dat'.
    :return: list of data with floats
    """
    # find 'ocean.dat' in current path
    data_path = r'./ocean.dat'

    # open data as only-read and save the data with a list called data_str, split by lines.
    with open(data_path, 'r') as d:
        data_str = d.read().splitlines()
        d.close()

    # because the data involves in list is strings, we need to convent them to floats.
    data = []
    for single_data_str in data_str:
        data.append(float(single_data_str))
    return data


def dft(data):
    """
    Discrete Fourier Transform the data
    :param data: the data waiting for fourier transform
    :return: the data after fourier transform
    """
    # need be absolute values, because we only focus the magnitude of the frequency.
    return np.abs(np.fft.rfft(data))


def remove_12_filter():
    """
    Create a filter to remove the variation of 12 hours and 25 minutes.
    :return: Filter coordinates
    """
    # use irrnotch function, w0 is the frequency needed to be removed.
    notch_b, notch_a = signal.iirnotch(w0=1 / (12 + 25 / 60), Q=0.23, fs=fs)
    return notch_b, notch_a
```

```python
def filter_twice(b, a, data):
    """
    Filter the data twice, once forward and once backward.
    :param b: Filter coordinates b
    :param a: Filter coordinates a
    :param data: the raw data waiting to be filtered
    :return: filtered data
    """
    return signal.filtfilt(b, a, data)


def remove_peak_of_12(data):
    """
    The main function to create a filter and filter the data to remove the variation
    of 12 hours and 25 minutes.
    :param data:the raw data waiting to be filtered
    :return:filtered data
    """
    notch_b, notch_a = remove_12_filter()
    return filter_twice(notch_b, notch_a, data)


def keep_6_filter():
    """
    Create a filter contains high-pass and low-pass, which low-pass is to reduce
    the noise and high-pass is
    to filter out the low frequency
    :return:the filter coordinates
    """
    Wn = np.array([0.07, 0.18])
    return signal.butter(N=5, Wn=Wn, btype='bandpass', analog=False)


def keep_peak_of_6(data):
    """
    The main function to create a filter and filter the data to only keep the variation
    of 6.2 hours.
    :param data: raw data of velocity
    :return: data after filtered
    """
    filter_6_b, filter_6_a = keep_6_filter()
    return filter_twice(filter_6_b, filter_6_a, data)


def gauss_function(x, a, x0, sigma):
    """
    Gauss function.
    :param x: x input
    :param a: real constant, height of the curve's peak
    :param x0: real constant, position of the centre of the peak
    :param sigma:non-zero constant, standard deviation
    :return:Gauss function
    """
    return a * np.exp(-(x - x0) ** 2 / (2 * sigma ** 2))
```

```python
# Plotting Part
def plot_v_vs_t(time_range, data):
    """
    Plot the graph with velocity of the ocean current against the time.
    :param data:raw data of velocity
    :return:True if successive
    """
    time_range = np.arange(0, len(data) * 14, 14)
    plt.plot(time_range, data, '.', label='velocity')
    plt.legend()
    plt.title('Velocity of the Ocean Current vs Time')
    plt.xlabel('Time/min')
    plt.ylabel('Velocity of the Ocean Current/ m*s^(-1)')
    plt.show()
    return True


def plot_fourier_vs_f(frequency, data_fft):
    """
    Plot the graph with frequency of the ocean current against fourier data.
    :param frequency: frequency of the current
    :param data_fft: data after fourier transform
    :return: True if successive
    """
    plt.plot(frequency, data_fft, label='Fourier')
    plt.title("Fourier Amplitude vs Frequency of the Ocean Current")
    plt.xlabel('Frequency of the Ocean Current/ hours^-1')
    plt.ylabel('Fourier Amplitude/ dB')
    plt.legend()
    plt.show()
    return True


def plot_fourier_zoom_in(frequency, data_fft):
    """
    Plot the graph with frequency of the ocean current against fourier data within 0 to 1.
    :param frequency: frequency of the current
    :param data_fft: data after fourier transform
    :return: True if successive
    """
    plt.plot(frequency, data_fft, label='Fourier')
    plt.title("Fourier Amplitude vs Frequency with X-axis Range Between 0 and 1")
    # vertical lines for different peaks
    plt.axvline(0.16, label='6.2 hours', color='aqua')
    plt.axvline(1 / (12 + 60 / 25), label='12 hours and 25 mins', color='violet')
    plt.axvline(0.0416, label='24 hours', color='coral')
    plt.axvline(0.00297, label='336 hours', color='lime')
    plt.xlabel('Frequency of the Ocean Current/ hours^-1')
    plt.ylabel('Fourier Amplitude/ dB')
    plt.xlim(0, 1)
    plt.legend()
    plt.show()
    return True
```

```python
def plot_fourier_removed_12hours(frequency, data_filtered):
    """
    Plot the graph with frequency of the ocean current against fourier data within 0 to 1
    when the variation of 12 hours has been removed .
    :param frequency: frequency of the current
    :param data_filtered: filtered data after removal of the variation
    :return: True if successive
    """
    plt.plot(frequency, data_filtered, label='Filtered')
    plt.title("Fourier Amplitude vs Frequency Removed 12 Hours Variation")
    plt.xlabel('Frequency of the Ocean Current/ hours^-1')
    plt.ylabel('Fourier Amplitude/ dB')
    plt.axvline(0.0416, label='24 hours', color='coral')
    plt.axvline(0.00297, label='336 hours', color='lime')
    plt.xlim(0, 1)
    plt.legend()
    plt.show()
    return True


def plot_fourier_6(frequency_range, data_fft_6):
    """
    Plot the graph with frequency of the ocean current, which only contains
    the variation of 6.2 hours,
    against fourier data within 0 to 1
    :param frequency_range: frequency of the current
    :param data_fft_6: filtered data with only variation of 6 hours
    :return: True if succesive
    """
    plt.plot(frequency_range, data_fft_6, label='Filtered')
    plt.title("Fourier Amplitude vs Frequency with Only 12 Hours Variation")
    plt.xlabel('Frequency of the Ocean Current/ hours^-1')
    plt.ylabel('Fourier Amplitude/ dB')
    plt.xlim(0, 1)
    plt.legend()
    plt.show()
    return True


def plot_fourier_6_with_gauss(frequency_range, data_fft_6, fitting_x, gauss_line):
    """
    Plot the graph with frequency of the ocean current, which only contains
    the variation of 6.2 hours, against fourier data within 0 to 1, and
    the graph of fitting line is also plotted.
    :param frequency_range: frequency of the current
    :param data_fft_6: filtered data with only variation of 6 hours
    :param fitting_x: the x values with range between 0 to 1
    :param gauss_line: fitting gauss line
    :return: True if successive
    """
    plt.plot(frequency_range, data_fft_6, label='Filtered')
    plt.plot(fitting_x, gauss_line, label='Gauss Fitting')
    plt.title("Fourier Amplitude vs Frequency and Gauss Fitting
```

```
            with Only 12 Hours Variation")
    plt.xlabel('Frequency of the Ocean Current/ hours^-1')
    plt.ylabel('Fourier Amplitude/ dB')
    plt.xlim(0, 1)
    plt.legend()
    plt.show()
    return True


# Main body
if __name__ == '__main__':
    # read the data first
    data = read_data()

    # because the data is recorded every 14 minutes
    # the sampling frequency is 60/14 then, with unit of hours
    global fs
    fs = 60 / 14  # in hours

    # plot the raw data between velocity and time
    time_range = np.arange(0, len(data)) * 14 / 60
    plot_v_vs_t(time_range, data)

    # do discrete fourier transform with the raw data
    data_fft = dft(data)

    # plot the fourier amplitude with the frequency
    frequency_range = np.arange(0, len(data_fft)) * fs / len(data_fft)
    plot_fourier_vs_f(frequency_range, data_fft)

    # plot the fourier amplitude with the frequency in the range of 0<frequency<1
    plot_fourier_zoom_in(frequency_range, data_fft)

    # Remove variation on 12 hours and 25 minutes
    filtered_12 = remove_peak_of_12(data)  # get filter data of velocity
    data_fft_12 = dft(filtered_12)

    # plot the fourier amplitude with the frequency when the variable of 12 hours
    # has been removed
    plot_fourier_removed_12hours(frequency_range, data_fft_12)

    # Only keep the variation of the period of 6.2 hours
    filtered_6 = keep_peak_of_6(data)
    data_fft_6 = dft(filtered_6)

    # plot the fourier amplitude with the frequency only with the variable of 6.2 hours
    plot_fourier_6(frequency_range, data_fft_6)

    # create a fitting line with gauss function to fit the fourier amplitude
    popt, pcov = optimize.curve_fit(f=gauss_function, xdata=frequency_range,
                                    ydata=data_fft_6)
    fitting_x = np.linspace(0, 1, 1000)
    gauss_line = gauss_function(fitting_x, *popt)

    # plot the fourier amplitude with the frequency only with the variable of 6.2 hours
```

```
# and its fitting line
plot_fourier_6_with_gauss(frequency_range, data_fft_6, fitting_x, gauss_line)
exit()
```