Erica Marroquin
CEE 4350: Coastal Engineering
HW 4

1) A data record of the free surface elevation, $\eta(t)$, is taken by a wave gauge. Find estimates for:

i) The significant wave height, $H_{1/3}$
ii) The root mean square wave height, H_{rms}
iii) The significant wave period, $T_{1/3}$

The significant wave height and rms wave height were found using the "upward zero crossing" method. A script in MATLAB was written (attached in the Appendix) to find these using the entire given wave record data.

$$H_{1/3} = 4.7759\ cm$$
$$H_{rms} = 0.2990\ cm$$

A histogram showing the wave height is shown below in figure 1.

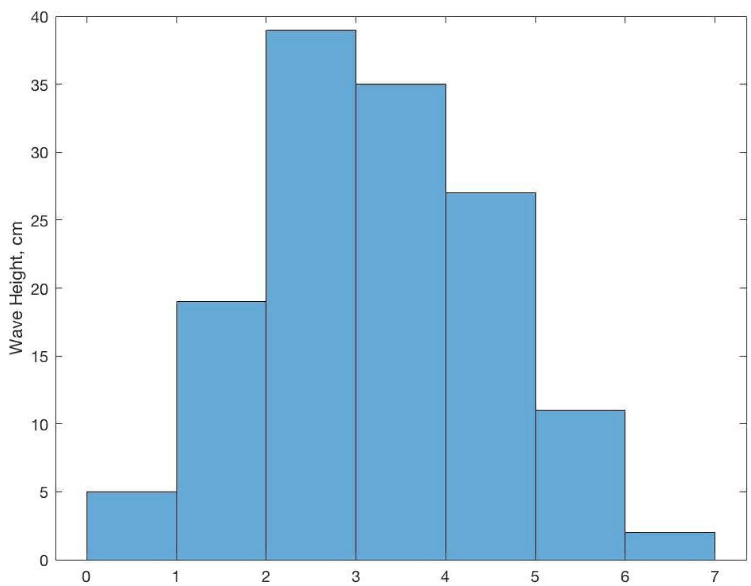

Figure 1: Histogram of wave height, in centimeters (cm), of the full set of given data.

The significant wave period was found using both the time between zero crossings, T_z, and the time between crests, T_c of the given data. This was found using a similar code to the one used to find $H_{1/3}$. This confirms the theory that $T_c < T_z$.

$$T_z = 1.0826\ s$$
$$T_c = 0.7304\ s$$

2) A simplified version of the Pierson and Moskowitz frequency (wave energy) spectrum can be written as:

$$S_{\eta\eta}(f) = 0.205 H_{1/3}^2 T_{1/3}^{-4} f^{-5} exp[-0.75(T_{1/3} f)^{-4}]\ (cm^2 s)$$

i) What is the frequency of the max energy? Find:

1. H_{rms}
2. $H_{1/3}$
3. $H_{1/10}$

ii) Consider a wave depth of $h = 10\ m$. What is the corresponding frequency spectrum for the dynamic pressure at the seafloor? Provide the analytical expression for $S_{pp}(f)$ and plot the results.

The period used for this analysis was $T_z = 1.0826\ s$. The frequency of the maximum energy is $f = 0.8165\ Hz$. This can be seen in the graph of the $S_{pp}(f)$ function below, in figure 2.

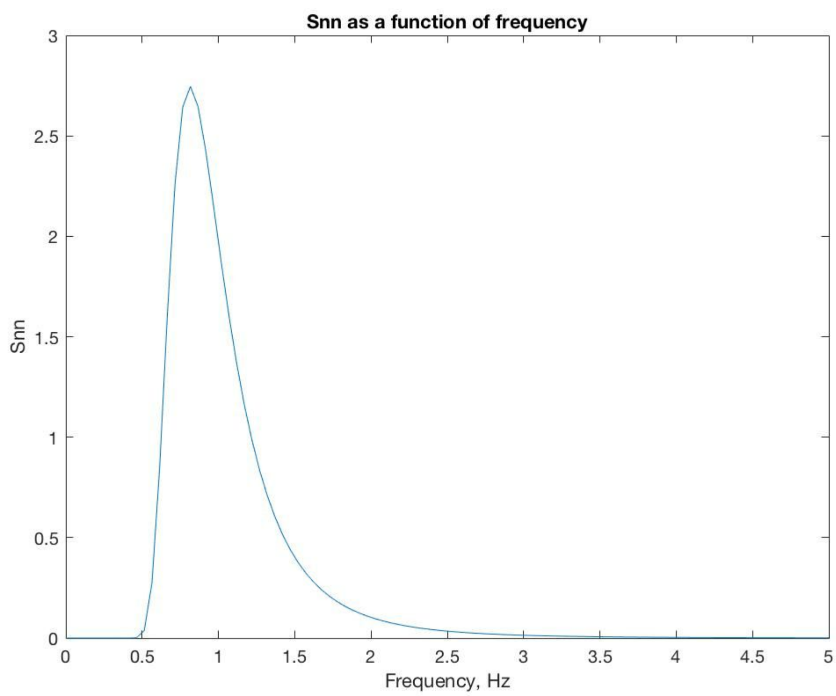

Figure 2: Plot showing $S_{\eta\eta}$ as a function of frequency.

To find the different wave heights, the area under the $S_{\eta\eta}$ curve was used to find H_{rms}. From that, $H_{1/3}$ and $H_{1/10}$ were found.

$$H_{rms} = 3.9256\ cm$$

$$H_{1/10} = 20\sqrt{2} H_{rms} \frac{sin(\frac{\pi}{20})}{\pi} = 4.9712\ cm$$

$$H_{1/3} = 6\sqrt{2} H_{rms} \frac{sin(\frac{\pi}{6})}{\pi} = 4.7667\ cm$$

To find the frequency spectrum for the dynamic pressure at the seafloor, the equation for $S_{\eta\eta}$ was modified to be:

$$Spp(f) = \sqrt{\frac{S_{\eta\eta}gh}{cosh(kh)}}$$

The results for the dynamic pressure frequency spectrum are shown below, in figure 3:

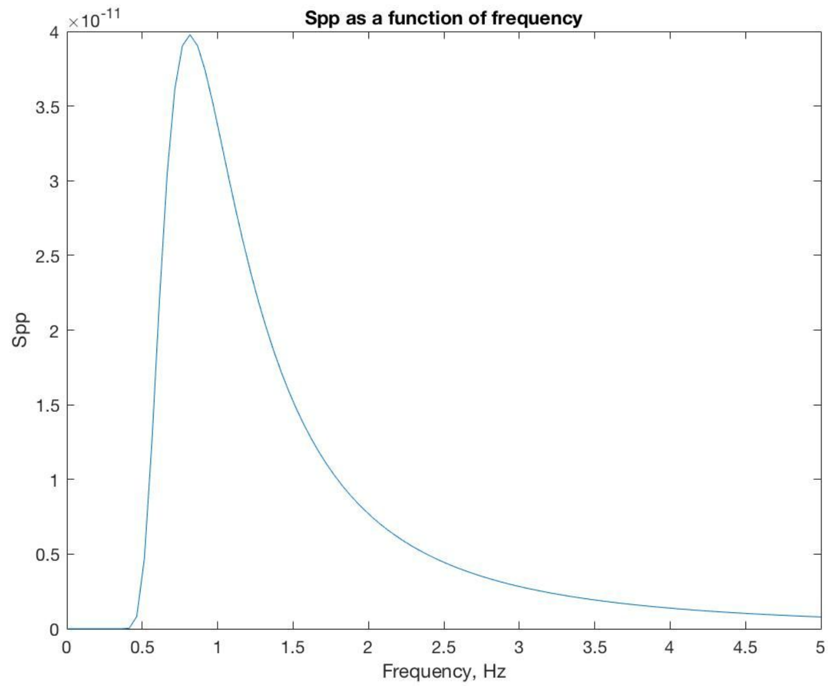

Figure 3: Plot showing S_{pp} as a function of frequency (f).

3) A train of small amplitude waves with wave period $T = 10\ s$ is propagating over a sloping bottom, which is uniform in the alongshore direction (y-direction). The angle of incidence is deep water is 30° and the wave amplitude is $a_0 = 1\ m$. The bathymetry can be expressed as:

$$h(x) = s\{x + \delta x_0 exp[-\frac{(x-x_0)^2}{\lambda}]\}$$

Consider the following situations:

i) $s = \frac{1}{50},\ \delta = 0$

ii) $s = \frac{1}{50},\ \delta = 0.5,\ x_0 = -1000\ m,\ \lambda = 5000\ m$

iii) $s = \frac{1}{50},\ \delta =- 0.5,\ x_0 = -1000\ m,\ \lambda = 5000\ m$

For each case:

i) Plot the bathymetry variation in the on-offshore direction, i.e., a plot of x vs. $h(x)$ (x on the horizontal axis).

ii) Trace (i.e., plot) a wave ray from deep water to the shoreline where the water depth is zero (this requires numerical integration).

iii) Plot the wave amplitude variation along the wave ray.

The results for problem 3 were found using data analysis methods in Python. The code for this can be found in the appendix. For the case where $\delta = 0$, the bathymetry variation is shown below, in figure 4.

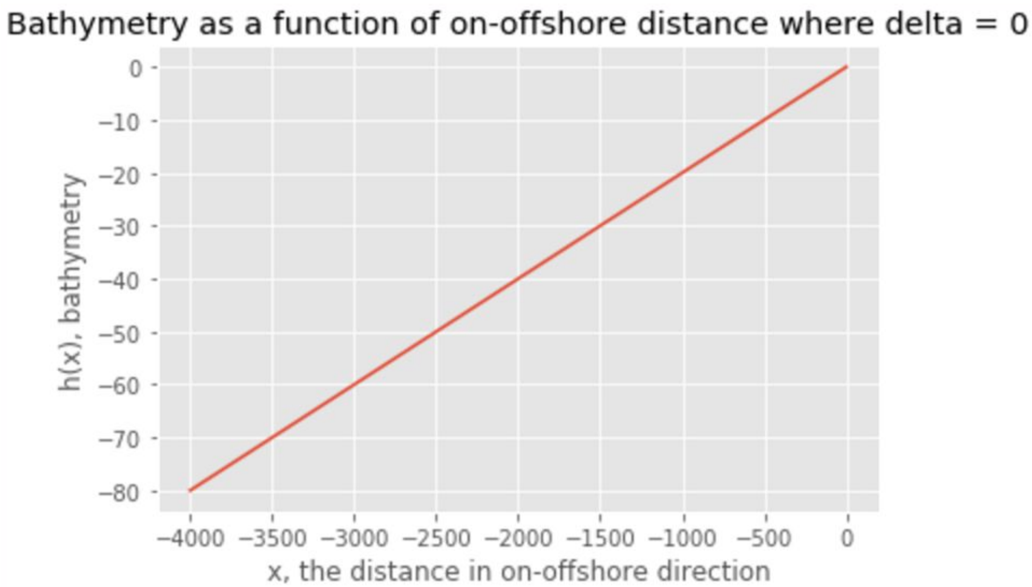

Figure 4: Plot showing the bathymetry as a function of on-offshore distance when $\delta = 0$.

The wave rays were found using a for-loop which performed numerical integration. The equation used to find the along-shore distance, or Δy, was:

$$\Delta y = \int_0^x \frac{\kappa}{\sqrt{k^2 - \kappa^2}} dx$$

Where:

$$\kappa = k_0 sin(\alpha_0)$$
$$\alpha_0 = 30°$$

The wave way when $\delta = 0$ is shown below in figure 5. The y-value starts at an arbitrary value. It can be assumed that the maximum value is essentially the “zero” y-value, as the wave ray rounds off as the wave hits the shore at x = 0.

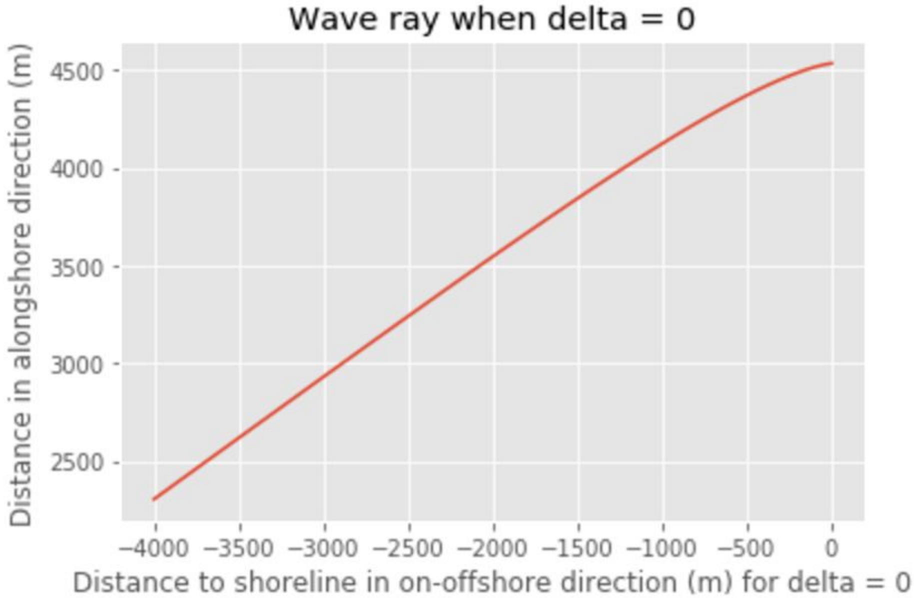

Figure 5: Plot showing the wave ray when $\delta = 0$.

Wave amplitude along the wave ray is expressed as:

$$a = a_0 \sqrt{\frac{Cg_0}{Cg}} \sqrt{\frac{b_0}{b}}$$

Where:

$$Cg = \sqrt{gh} \text{ and}$$
$$b = cos(\alpha)$$
$$\alpha = tan^{-1}(\frac{\Delta y}{\Delta x})$$

The wave amplitude when $\delta = 0$ is shown below in figure 6. Since the bathymetry along the bed is constant, as shown in figure 4, the wave amplitude exponentially increases as the wave approaches the shore.

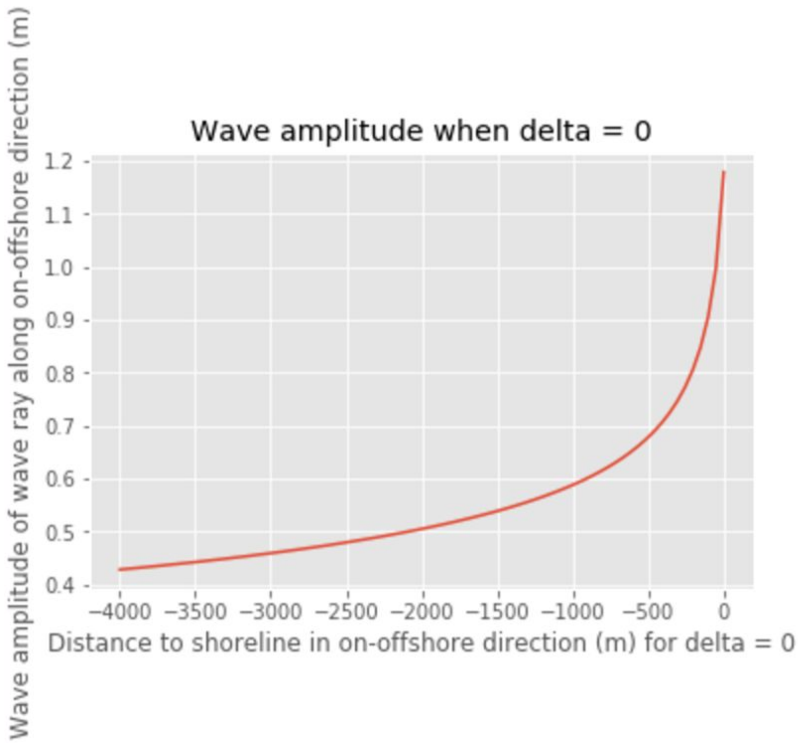

Figure 6: Wave amplitude when $\delta = 0$.

The same calculations were done for the conditions where $\delta = 0.5$ and $\delta = -0.5$. The plots for this are shown below. If there were any variations in bathymetry, that would be shown via an amplitude change because of conservation of energy (less space, higher amplitude to have energy remain constant, etc.). When the bathymetry changes at $x_0 = -1000\ m$ there is a decrease/increase in amplitude at that point, since the bathymetry increased/decreased. The wave ray is not a function of bathymetry, so it is not affected.

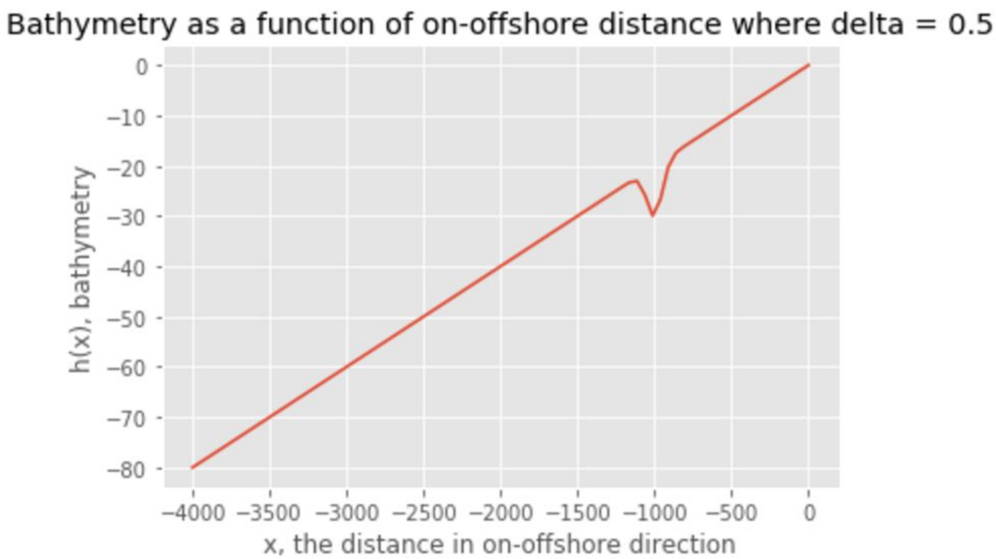

Figure 7: Plot showing the bathymetry as a function of on-offshore distance when $\delta = 0.5$

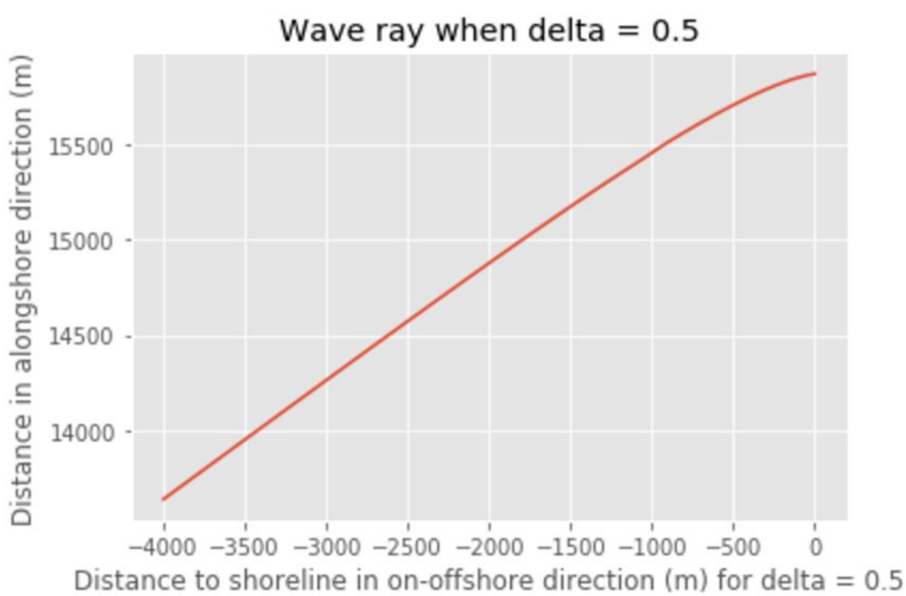

Figure 8: Plot showing the wave ray as a function of on-offshore distance when $\delta = 0.5$

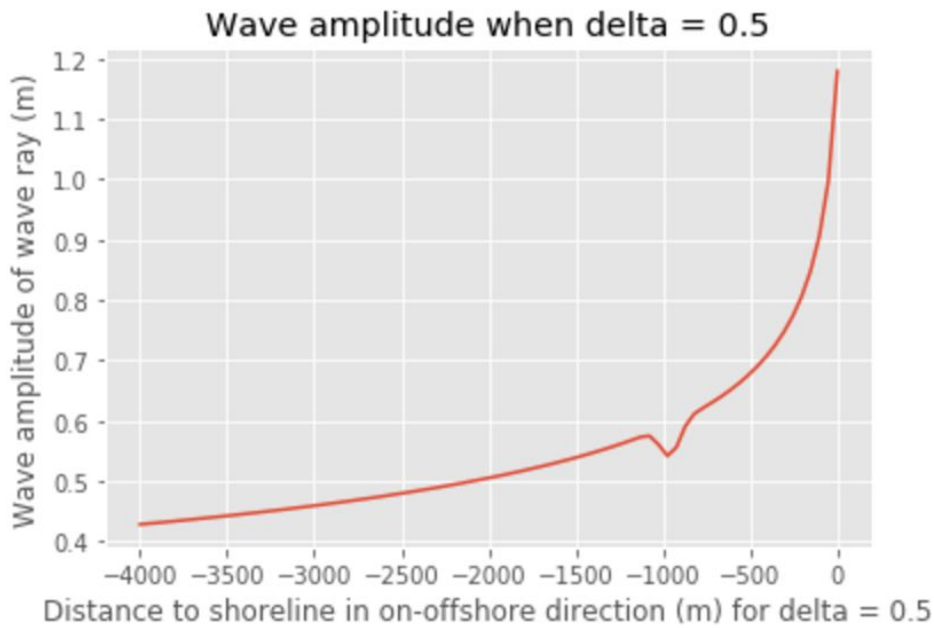

Figure 9: Plot showing the wave amplitude as a function of on-offshore distance when $\delta = 0.5$

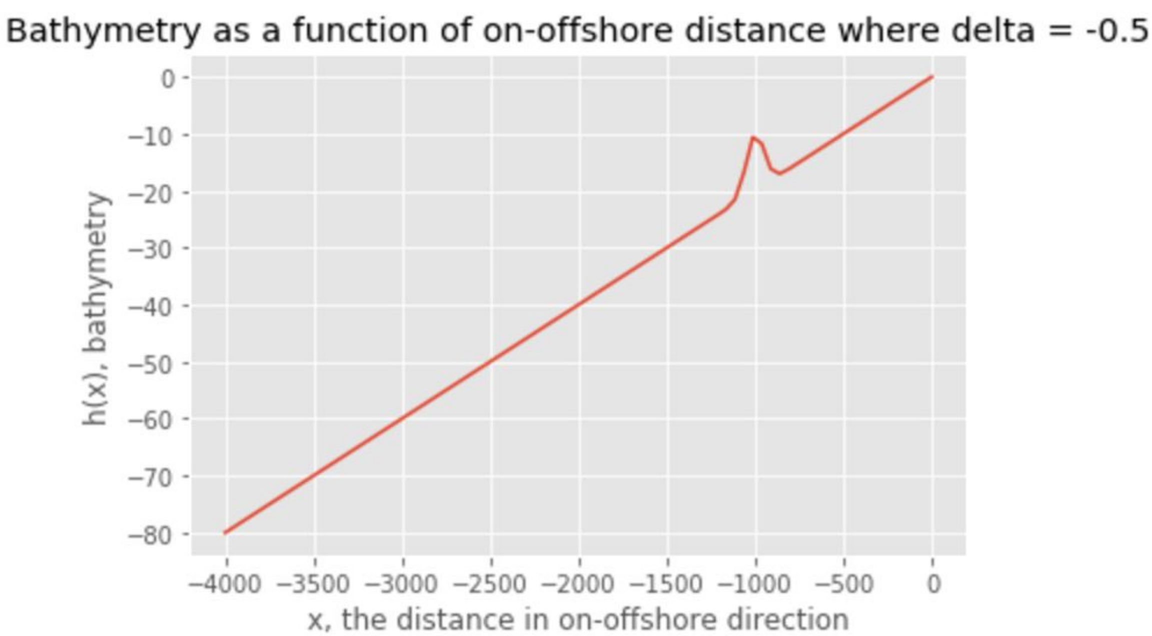

Figure 10: Plot showing the bathymetry as a function of on-offshore distance when $\delta = -0.5$

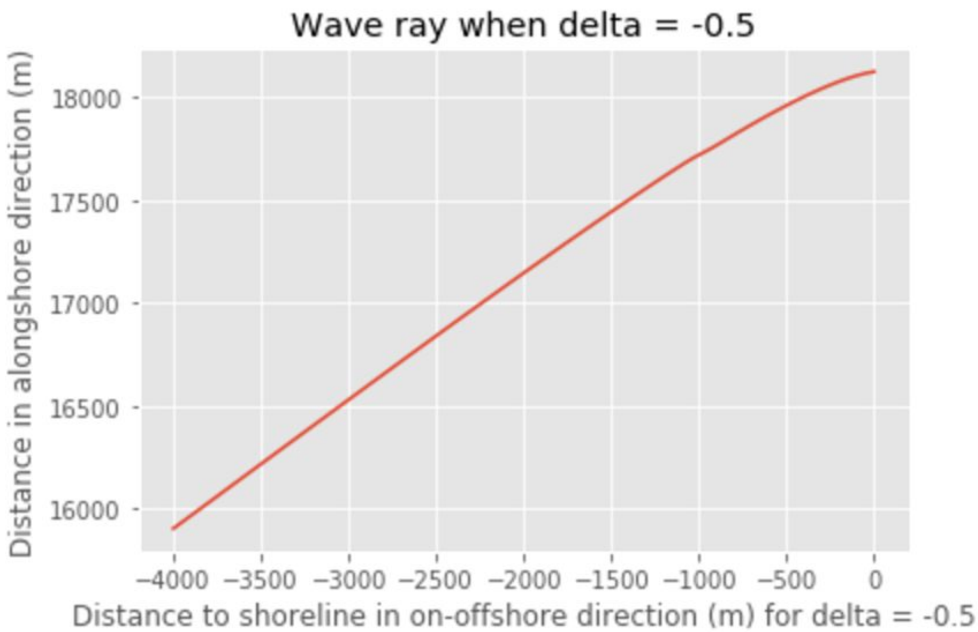

Figure 11: Plot showing the wave ray as a function of on-offshore distance when $\delta = -0.5$

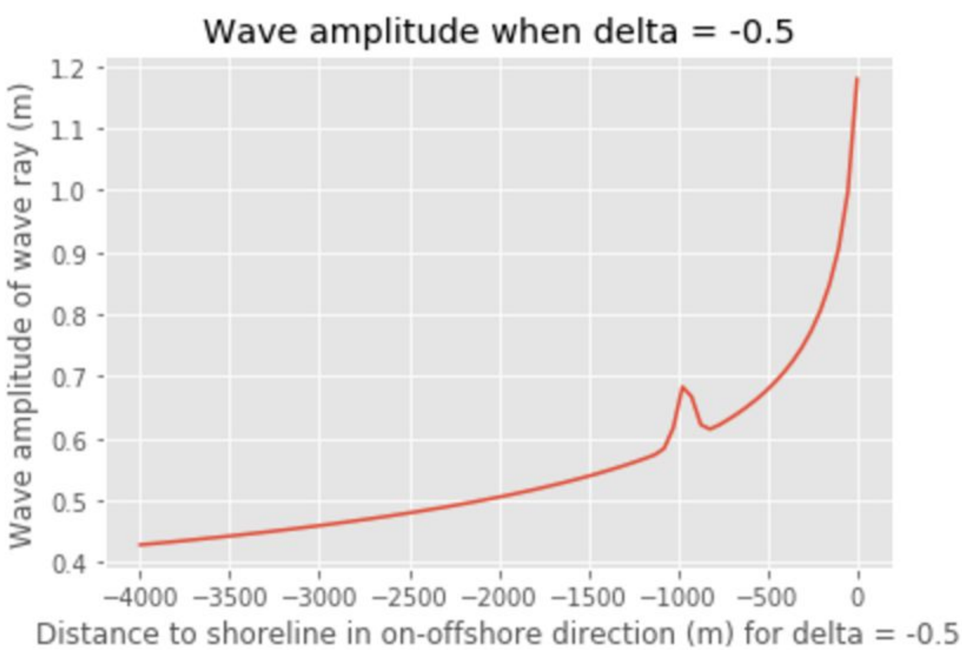

Figure 12: Plot showing the wave amplitude as a function of on-offshore distance when $\delta = -0.5$

Appendix

```
%%%%%%%%%%%% PROBLEM 1
file_name = 'waverecord_HW4.dat';
HW4_data = importdata(file_name);
time = HW4_data(:,1);
eta = HW4_data(:,2);

j = 0;
for i = 1:length(eta)
    if eta(i) < 0 && eta(i+1) > 0
        j = j + 1; % finds the number of zero crossings
        up(j) = i; % finds the index of each upward zero crossing
    end
end

Nz = j; % number of zero crossings
N = Nz - 1;

Hj = zeros(1,j-1);
for i = 1:j-1
    Hj(i) = max(eta(up(i):up(i+1))) - min(eta(up(i):up(i+1))); %finds H, wave heights between crests and throughs
end

Hi = sort(Hj,'descend'); % sorts the wave heights in descending order
sig_wave_height = 3/N * sum(Hi(1:(N/3))); % H(1/3)

histogram(eta)
ylabel('Wave Height, cm')

Hj_sq = Hj.^2;
rms_wave_height = 1/N * sqrt(sum(Hj_sq(1:N)));
```

```matlab
[pks,locs] = findpeaks(eta); % finding the index of local maxima of eta
time_btwn_crests = zeros(1,length(locs)-1);
for i = 1:length(locs)-1
  time_btwn_crests(i) = time(locs(i+1)) - time(locs(i)); %finding the time between each crest
end

time_btwn_crests_i = sort(time_btwn_crests, 'descend'); % sorting the time into descending order
sig_wave_period_Tc = 3/N * sum(time_btwn_crests_i(1:(N/3)));

time_of_zero_cross = time(up); % finding the time of each zero crossing
time_btwn_zeros = zeros(1,length(time_of_zero_cross)-1);
for i = 1:length(time_of_zero_cross)-1
   time_btwn_zeros(i) = time_of_zero_cross(i+1) - time_of_zero_cross(i); %finding the time between each zero crossing
end

time_btwn_crests_z = sort(time_btwn_zeros, 'descend'); % sorting the time into descending order
sig_wave_period_Tz = 3/N * sum(time_btwn_crests_z(1:(N/3)));

%%%% Problem 2
H_third = 4.7759;
T_third = 1.0826;
f = linspace(0.01,5); % creating an evenly spaced array from (almost) 0 to 5. spectra does not like the handle 0 values.

Snn = 0.205*H_third^2*T_third^(-4)*f.^-5.*exp(-0.75*(T_third*f).^(-4));

[max_a,max_f] = findpeaks(Snn); % finds the index of the maximum value of energy
peak_f = f(max_f); % finds the frequency of the max energy value

plot(f,Saa)
xlabel('Frequency, Hz')
ylabel('Snn')
title('Snn as a function of frequency')

df = f(2) - f(1);
area = sum(Snn*df); % calculates the area under the Snn curve
a_rms = sqrt(2*area);
H_rms = 2*a_rms;
H_max = sqrt(2)*H_rms;
H_onetenth = 20*sqrt(2)*H_rms*sin(pi/20)/pi;
H_onethird = 6*sqrt(2)*H_rms*sin(pi/6)/pi;


%%% PART 2
sigma_sq = ((2*pi)/(T_third))^2;
g = 9.81; % gravity
h = 10; % meters, given
rho = 1000; %kg/m^3

LHS = sigma_sq/g;
k = 0;
while k*tanh(k*h) <  LHS
   k = k + 0.0001;
end

pressure_f = sqrt(Snn)*rho*g/(cosh(k*h));

plot(f, pressure_f)
xlabel('Frequency, Hz')
ylabel('Spp')
title('Spp as a function of frequency')
```

Problem 3 Part i
```python
g = pc.gravity.magnitude

def h_x(delta, x):
  bathy = s*(x + delta * x_0 * np.exp(-(x-x_0)**2/(wavelength)))
  return bathy

s = 1/50
T = 10 #s
x_0 = -1000 #m
wavelength = 5000 #m
delta_x = 50 #m taking a measurement every 50 m to cut down number of iterations that need to occur
array_elements = 4000/delta_x
array_elements_loop = int(array_elements - 1)
x_array = np.linspace(-4000, 0, num = 80)

#plotting bathymetry in the on-offshore direction (i)
#for delta = 0 condition
plt.plot(x_array, h_x(0, x_array))
plt.xlabel('x, the distance in on-offshore direction')
plt.ylabel('h(x), bathymetry')
plt.title('Bathymetry as a function of on-offshore distance where delta = 0')
plt.show()

#for delta = 0.5 condition
plt.plot(x_array, h_x(0.5, x_array))
plt.xlabel('x, the distance in on-offshore direction')
plt.ylabel('h(x), bathymetry')
plt.title('Bathymetry as a function of on-offshore distance where delta = 0.5')
plt.show()

#for delta = -0.5 condition
plt.plot(x_array, h_x(-0.5, x_array))
plt.xlabel('x, the distance in on-offshore direction')
plt.ylabel('h(x), bathymetry')
plt.title('Bathymetry as a function of on-offshore distance where delta = -0.5')
plt.show()
```

Part ii
```python
```

```
#making different arrays to fill with a loop for numerical integration
#if the arrays were expanded each time the loop iterated, the run time would
#be SO LONG. like in MATLAB, predefined arrays are best.

delta_y = np.ones(array_elements_loop)
delta_y[0] = 0
y = np.ones(array_elements_loop)
y[0] = 0
alpha = np.zeros(array_elements_loop)
alpha[0] = 30 #given angle of incidence
h = np.ones(array_elements_loop)
h[0] = abs(h_x(0, x_array[0]))
k = np.ones(array_elements_loop)
k[0] = CF.wavenumber(T, h[0])
Cg = np.zeros(array_elements_loop)
Cg[0] = np.sqrt(g*h[0])
h_0 = h[0]
Cg_0 = Cg[0]
b = np.zeros(array_elements_loop)
b_0 = np.cos(alpha[0])
a = np.zeros(array_elements_loop)
a[0] = 1
a_0 = a[0]

KAPPA = k[0]*np.sin(a_0) #a constant

def wave_rays(lil_delta):
  for i in range(0, array_elements_loop):
    k[i] = CF.wavenumber(T, abs(h_x(lil_delta, x_array[i])))  #finding a new k for each height
    delta_y[i] = KAPPA/(np.sqrt(k[i]**2 + KAPPA**2))*delta_x  #finding the new step change
    y[i] = y[i-1] + delta_y[i]  #moving over by previous delta y to get a new height
    alpha[i] = np.arctan(delta_y[i]/delta_x)  #finding a new angle of incidence
    Cg[i] = np.sqrt(g*abs(h_x(lil_delta, x_array[i])))  #calculates new group velocity
    b[i] = np.cos(alpha[i])
    a[i] = a_0*np.sqrt(Cg_0/Cg[i])*np.sqrt(b_0/b[i])   #new amplitude
  return k, delta_y, y, alpha, Cg, b, a
```

Graphs for Part ii and iii
```python
x_graphing = np.linspace(-4000, 1, num = 79) #to make the vectors the same length (delta_y is 79 elements in the array_elements_loop)
```

```
#code was changed each time there were different conditions for amplitude or wave ray
plt.plot(x_graphing, wave_rays(-0.5)[6]) #wave_rays returns an array, and y values are the second index of the arrays, amplitudes are sixth index
plt.xlabel('Distance to shoreline in on-offshore direction (m) for delta = -0.5')
plt.ylabel('Wave amplitude of wave ray (m)')
plt.title('Wave amplitude when delta = -0.5')
plt.show()
```