

CEE 4350 Coastal Engineering

Lab 2

Zoe Maisel

Introduction:

An acoustic wave gauge and ADV sensor was used to measure wave amplitude and wave velocity in time. Multiple waves at different wave paddle strokes and frequencies were generated to measure wavelength. This experimental data was compared with theoretical analysis of expected dispersion and wave celerity relationships. The ADV measured water velocity at four different heights from the water surface and showed wave profile and velocity decay near bed. Again, experimental data was compared to theoretical analysis. In both cases, the experimental data fit the theoretical results well and were within the expected margin of experimental error.

```
from aide_design.play import*
import pandas as pd
g = pc.gravity
import os
import plotly.plotly as py
import plotly.graph_objs as go
```

```
cwd = os.getcwd() # Get the current working directory (cwd)
files = os.listdir(cwd) # Get all the files in that directory
```

Variable definition:

g : gravity

σ : dispersion

a : wave amplitude

A : velocity amplitude

f : frequency

f_s : sampling frequency

h : water depth

H : distance from wave crest to trough ($2a$)

T : wave period

λ : wavelength

k : wavenumber

c_p : celerity (wave phase speed)

u, w : x-velocity, z-velocity components

Governing Equations for Linear Wave Theory Analysis

$$\sigma = \left(\frac{2\pi}{T}\right)^2$$

$$c_p = \frac{gT}{2\pi} \tanh(kh)$$

$$\sigma^2 = gk * \tanh(kh)$$

$$T = \frac{1}{f}$$

$$\sigma = 2\pi f$$

$$k = \frac{2 * \pi}{\lambda}$$

$$u = a\sigma \frac{\cosh(k(h+z))}{\sinh(kh)} \cos(kx - \sigma t)$$

$$w = a\sigma \frac{\sinh(k(h+z))}{\sinh(kh)} \sin(kx - \sigma t)$$

For u ,

$$A = a\sigma \frac{\cosh(k(h+z))}{\sinh(kh)}$$

$$a = \frac{A}{\sigma \frac{\cosh(k(h+z))}{\sinh(kh)}}$$

For w ,

$$A = a\sigma \frac{\sinh(k(h+z))}{\sinh(kh)}$$

$$a = \frac{A}{\sigma \frac{\sinh(k(h+z))}{\sinh(kh)}}$$

The following conditions were given for the experimental analysis.

```
h = .20 # meters
freq = np.array([0.55, 1.00, 1.55]) # Hz
period = 1/freq # second
sigma = 2 * np.pi/period
```

Calculation of $\frac{\sigma^2 h}{g}$

Experimental

Using the wavemaker and flume, waves at the following experimental conditions were produced: | Stroke (mm) | Frequency (Hz) | | ————— | ————— | | 5 | 0.55 | | 10 | 0.55 | | 5 | 1.00 | | 10 | 1.00 | | 5 | 1.55 | | 10 | 1.55 |

Two wave gauge measurements were used to track wave progression in the flume and to determine wavelength. Wavelength was determined when the two wave gauges reported the same wave in phase, showing that the wave completed one wavelength to return to its original position. Wavenumber was then found using wavelength data.

$$k = \frac{2\pi}{\lambda}$$

```
lmbda_exp = np.array([2.42678, 1.22178, 0.62578])
```

```
k_exp = 2*np.pi/lmbda_exp
print(k_exp)
```

Stroke (mm)	Frequency (Hz)	Wavelength, λ (m)	Wavenumber, k
5	0.55	2.43	2.59
10	0.55	2.43	2.59
5	1.00	1.22	5.14
10	1.00	1.22	5.14
5	1.55	0.626	10.04
10	1.55	0.626	10.04

Theoretical

Theoretically, wavenumber can be found using the dispersion relationship.

```
def wavenumber(T, h):
    """Return the wavenumber of wave using period and water height from bed."""
    k = 10 # this is a guess to find what k is
    diff = (((2*np.pi)/T)**2)-(g.magnitude * k * np.tanh(k*h))
```

```

while diff<0:
    LHS = ((2*np.pi)/T)**2
    RHS = g.magnitude * k * np.tanh(k*h)
    diff = LHS - RHS
    k = k - 0.0001
return k

k_theor = np.array([wavenumber(period[0],h),
                    wavenumber(period[1],h), wavenumber(period[2],h)])
print(k_theor)

```

Comparison and Analysis of Experimental and Theoretical Results

$\frac{\sigma^2 h}{g}$ as a function of kh

As shown by both the experimental and theoretical data, wavelength was not impacted by stroke conditions; only frequency impacted the wavelength and wavenumber. The frequency at which waves are produced by the wavemaker dictates the wavelength because that is what is driving the fluid and energy. The changing stroke may impact the amplitude of the wave, due to conservation of mass principles, but the wavelength remains unimpacted. The equation for wavenumber, which is related to wavelength, does not even require paddle stroke or amplitude as an input. The table and figure below show that the experimental data very closely aligns with the theoretical data, confirming the dominating relationship between frequency and wavelength.

Frequency (Hz)	Experimental Wavenumber, k	Theoretical Wavenumber, k
0.55	2.59	2.57
1.00	5.14	5.18
1.55	10.04	10.0

```

sigh_g_exp = sigma**2 * h / g
kh_exp = k_exp * h
plt.plot(kh_exp, sigh_g_exp, 'bo')
plt.xlabel('kh', fontsize=18)
plt.ylabel('sigma^2*h/g', fontsize=16)
sigh_g_theor = sigma**2 * h / g
kh_theor = k_theor * h
plt.plot(kh_theor, sigh_g_theor)
plt.savefig('sigh_g.png')
plt.show()

```

The experimental and theoretical analysis for $\frac{\sigma^2 h}{g}$ as a function of kh . Only one plot is shown for both wave paddle amplitudes because the results are the same.

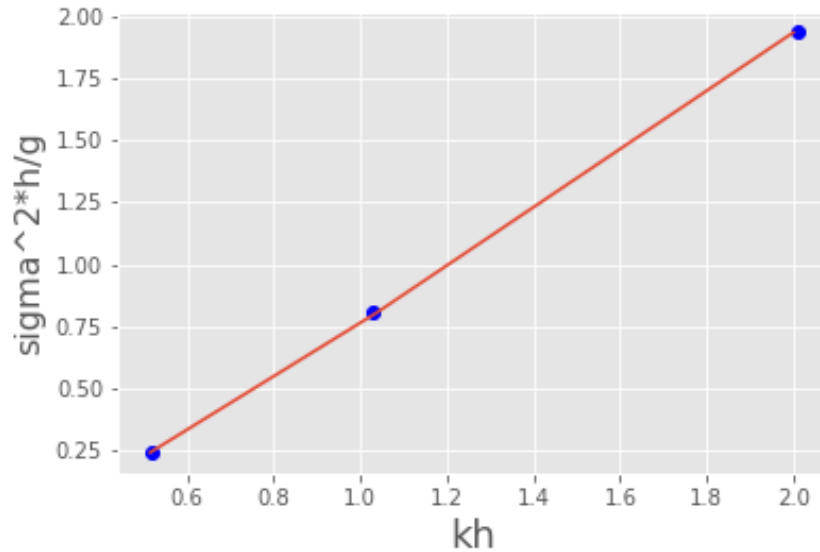


Figure 1: $\sigma^2 h / g$

Calculation of Wave Celerity

Wave celerity was calculated as an experimental and theoretical value. Celerity was found experimentally by $c_p = \frac{\lambda}{T}$.

Experimental

Experimental analysis was done using experimentally measured λ values and varying frequencies.

```
celerity_exp = lambda_exp / period
print(celerity_exp)
```

Theoretical

Theoretical analysis to determine wave celerity was done using $c_p = \frac{\lambda}{T}$, from k values and λ values calculated from the dispersion relationship.

```
def celerity(T,h):
    """Return the celerity of wave using period and water height from bed."""
    k = wavenumber(T,h)
    lambda = 2 * np.pi / k
```

```

    celerity = lmbda / T
    return celerity

celerity_theor = np.array([celerity(period[0], h),
    celerity(period[1], h), celerity(period[2], h)])
print(celerity_theor)

```

Comparison and Analysis of Experimental and Theoretical Results

Frequency (Hz)	Experimental c_p (m/s)	Theoretical c_p (m/s)
0.55	1.334729	1.34350047
1.00	1.22178	1.21212773
1.55	0.969959	0.97389372

```

cp_gh_exp = celerity_exp / (g*h)
plt.plot(kh_exp, cp_gh_exp, "bo")
plt.xlabel('kh', fontsize=18)
plt.ylabel('cp/gh', fontsize=16)
cp_gh_theor = celerity_theor / (g*h)
plt.plot(kh_theor, cp_gh_theor)
plt.savefig('celerity.png')
plt.show()

```

The experimental and theoretical analysis for $\frac{c_p}{gh}$ as a function of kh . Only one plot is shown for both wave paddle amplitudes because the results are the same. Again, the experimental results match the theoretical results.

Water Particle Velocity

All of the data was collected in the DeFrees Fluids Lab using an ADV to collect velocity data at different z-elevations from the bed. f_s , the sampling frequency was 200 Hz. The wave paddle was set at an f of 1.00 Hz. The data was truncated to ensure that all the wave analysis begins at the wave crest. At wave crests, x and t are assumed to be zero, which simplifies the velocity equations to:

$$u = a\sigma \frac{\cosh(k(h+z))}{\sinh(kh)}$$

$$w = a\sigma \frac{\sinh(k(h+z))}{\sinh(kh)}$$

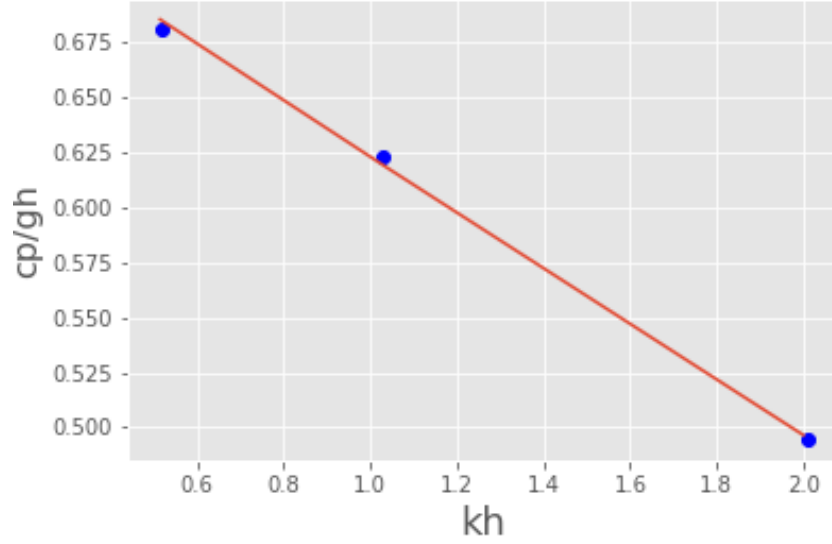


Figure 2: cp/gh

u_{max} and w_{max}

Experimental

First, the velocity amplitude A was found by sorting the absolute values of the data. The mean of the largest 10% of data was chosen. This velocity amplitude is the experimental maximum velocities found for each z-elevation analyzed.

```
z = np.array([-0.075, -0.1, -0.125, -0.175])
uA_amp_75 = 0.07482475
uA_amp_100 = 0.06905375
uA_amp_125 = 0.062269875
uA_amp_175 = 0.059395
u_vel_amp = np.array([uA_amp_75, uA_amp_100, uA_amp_125, uA_amp_175])

wA_amp_75 = 0.0380575
wA_amp_100 = 0.032196125
wA_amp_125 = 0.022905125
wA_amp_175 = 0.009311
w_vel_amp = np.array([wA_amp_75, wA_amp_100, wA_amp_125, wA_amp_175])
```

z (m)	u_{max} (m/s)	w_{max} (m/s)
-0.075	0.07482475	0.0380575

z (m)	u_{max} (m/s)	w_{max} (m/s)
-0.1	0.06905375	0.032196125
-0.125	0.062269875	0.022905125
-0.175	0.059395	0.009311

Theoretical

Theoretical analysis was done by using A values to calculate a , from the relationships given as:

for u ,

$$A = a\sigma \frac{\cosh(k(h+z))}{\sinh(kh)}$$

$$a = \frac{A}{\sigma \frac{\cosh(k(h+z))}{\sinh(kh)}}$$

for w ,

$$A = a\sigma \frac{\sinh(k(h+z))}{\sinh(kh)}$$

$$a = \frac{A}{\sigma \frac{\sinh(k(h+z))}{\sinh(kh)}}$$

```
f_wave = 1 # Hz
```

```
waveperiod_all = 1 # 1/s
```

```
k_all = wavenumber(waveperiod_all, h)
print(k_all)
```

```
sgma = 2 * np.pi * f_wave
print(sgma)
```

Eight a values were calculated, and the average of all a values were chosen. This was done because the wave amplitude should be constant across the wave, even at measurements taken at different wave depth.

```
ua_75_mm = uA_amp_75 / (sgma * np.cosh(k_all*(h+z[0])) / np.sinh(k_all*h))
ua_100_mm = uA_amp_100 / (sgma * np.cosh(k_all*(h+z[1])) / np.sinh(k_all*h))
ua_125_mm = uA_amp_125 / (sgma * np.cosh(k_all*(h+z[2])) / np.sinh(k_all*h))
ua_175_mm = uA_amp_175 / (sgma * np.cosh(k_all*(h+z[3])) / np.sinh(k_all*h))
```

```
wa_75_mm = wA_amp_75 / (sgma * np.sinh(k_all*(h+z[0])) / np.sinh(k_all*h))
wa_100_mm = wA_amp_100 / (sgma * np.sinh(k_all*(h+z[1])) / np.sinh(k_all*h))
wa_125_mm = wA_amp_125 / (sgma * np.sinh(k_all*(h+z[2])) / np.sinh(k_all*h))
```



```

wa_175_mm = wA_amp_175 / (sgma * np.sinh(k_all*(h+z[3])) / np.sinh(k_all*h))

ua_all = [ua_75_mm, ua_100_mm, ua_125_mm, ua_175_mm]
ua_mean = np.mean(ua_all)
wa_all = [wa_75_mm, wa_100_mm, wa_125_mm, wa_175_mm]
wa_mean = np.mean(wa_all)

a_all = [ua_all, wa_all]
a_mean = np.mean(a_all)
print(a_mean)

```

k	σ	mean a (m)
5.184	6.283	0.0118

Using the mean a wave, u and w can be found at varying z elevations by using the following equations:

$$u = a\sigma \frac{\cosh(k(h+z))}{\sinh(kh)}$$

$$w = a\sigma \frac{\sinh(k(h+z))}{\sinh(kh)}$$

```

z_theor_array = np.linspace(-0.075, -0.175, num = 50)
u_theor_array = a_mean * sgma * np.cosh(k_all*(h+z_theor_array)) / np.sinh(k_all*h)
print(u_theor_array)
w_theor_array = a_mean * sgma * np.sinh(k_all*(h+z_theor_array)) / np.sinh(k_all*h)
print(w_theor_array)

plt.plot(u_theor_array, z_theor_array)
plt.plot(u_vel_amp, z, "ro")
plt.xlabel('u-velocity (m/s)', fontsize=18)
plt.ylabel('z-elevations (m)', fontsize=16)
plt.savefig('uvel.png')
plt.show()

plt.plot(w_theor_array, z_theor_array)
plt.plot(w_vel_amp, z, "ro")
plt.xlabel('w-velocity (m/s)', fontsize=18)
plt.ylabel('z-elevations (m)', fontsize=16)
plt.savefig('wvel.png')
plt.show()

```

The experimental data matches the theoretical plots at different depths well. The u -velocity plot shows the expected velocity decay due to boundary condition effects of the flume closer to the bed. The w -velocity similarly follows expected

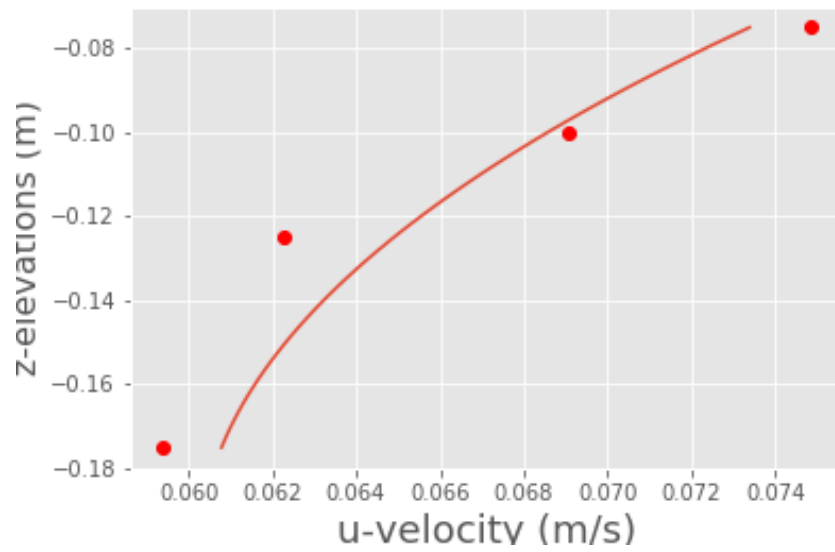


Figure 3: uvel

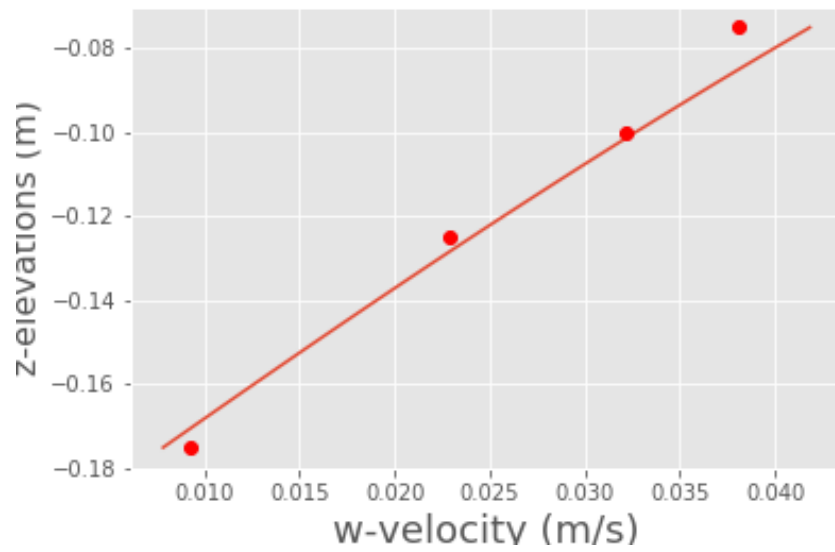


Figure 4: wvel

trends in which there is close to zero w-velocity close to the bed due to the impermeable surface, and there is some w-velocity away from the bed. The motion of water parcels in waves as decaying vortices is well modeled by the lab.

```
# To convert the document from markdown to pdf  
pandoc Lab_2.md -o Maisel_Lab_2.pdf
```