CEE 4350 Coastal Engineering

Lab 3

Zoe Maisel

Introduction:

Wave amplitude can be estimated by the shoaling formula until the point of wave break for mild slope and conservation of energy flux assumptions. Four waves were generated at different strokes and frequencies, and their heights were measured at different distances along the x-axis, on-shore direction. The initial wave heights and wave lengths were used to calculate the Iribarren number, which was then compared to observed wave breaker type. Understanding energy flux, slopes, and wave height are important in understanding coastal systems.

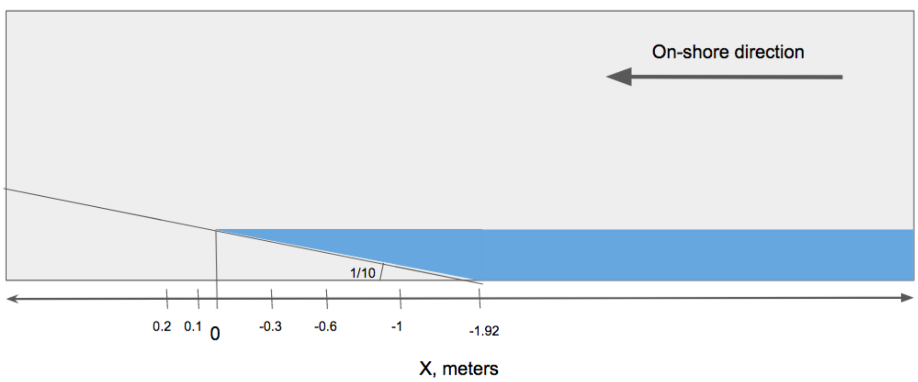


Figure 1:

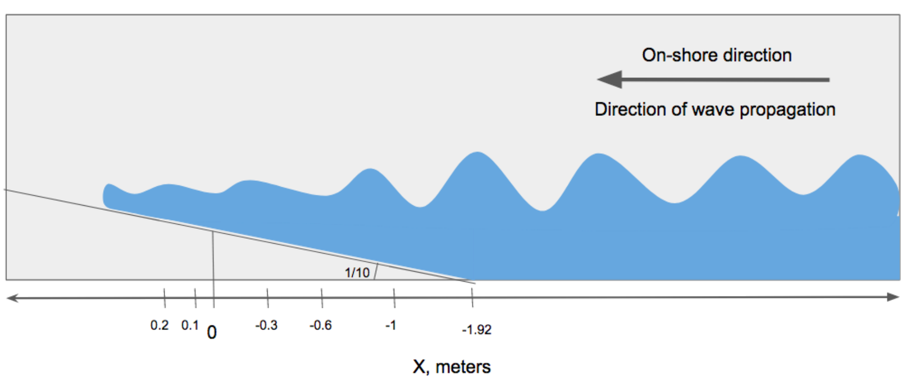


Figure 2:

Function definitions:

```
from aide_design.play import*
g = pc.gravity

def wavenumber(T, h):
  """Return the wavenumber of wave using period and water height from bed."""
  k = 800  # 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 - 0.5
  return k
```

Wave Amplitude:

The hydraulic-powered wave paddle in the DeFrees Lab was used to generate waves for the experiment. Experimental wave conditions generated for the lab are shown below.

Case	Stroke (mm)	Frequency (Hz)
W1	35	1.25
W2	40	0.85
W3	40	0.5
W4	30	0.5

slope	SWL (cm)	Distance from x=0 to flat bottom (m)
0.1	19.2	1.92

Experimental results for amplitude are shown below.

Case	H_0 (cm)	a (cm) @ x=-1.0 m	a (cm) @ x=-0.6 m	a (cm) @ x=-0.3 m	a (cm) @ x=0 m	a (cm) @ x=0.1 m	a (cm) @ x=0.2 m
W1	7.70	4.00	1.63	1.25	0.60	0	0
W2	7.10	4.25	1.90	1.55	1.10	0.85	0.3
W3	4.5	2.25	2.05	1.65	0.9	0.35	0.25
W4	2.9	1.5	1.7	1.1	0.75	0.5	0.25

Experimental data was compared with theoretical calculations using the shoaling formula. Constant width in the flume was assumed. The waves were analyzed in transitional water.

Experimental Analysis

First, all the experimentally found values were recorded and converted to meters.

```
# Experimental conditions
slope = 1/10
SWL_cm = 19.2
H0_cm = np.array([7.7, 7.1, 4.5, 2.9])
a0_cm = H0_cm / 2

W1_amp_cm = np.array([4.00, 1.63, 1.25, 0.6, 0, 0])
W2_amp_cm = np.array([4.25, 1.90, 1.55, 1.10, 0.85, 0.3])
W3_amp_cm = np.array([2.25, 2.05, 1.65, 0.9, 0.35, 0.25])
W4_amp_cm = np.array([1.5, 1.7, 1.1, 0.75, 0.5, 0.25])

# Convert from cm to m
SWL_m = SWL_cm/100
H0_m = H0_cm/100
a0_m = a0_cm/100

W1_amp_m = W1_amp_cm/100
W2_amp_m = W2_amp_cm/100
W3_amp_m = W3_amp_cm/100
W4_amp_m = W4_amp_cm/100
```

Each wave had data collected at x = -1 m, x = -0.6 m, x = -0.3 m, x = 0 m, x = 0.1 m, and x = 0.2 m. These x-distances were then related to water depth by using the constant slope of the flume bed. The 1/10 slope

```
# Relate the x-distances to water depths, h
x_m_exp = np.array([-1, -0.6, -0.3, 0, 0.1, 0.2])
h_m_exp = x_m_exp/10
```

The experimental data was then nondimensionalized by dividing the height and wave data vectors at each x-distance by the initial, flat-bed condition.

$$h_{ND} = \frac{h(x)}{h_0}$$

$$a_{ND} = \frac{a(x)}{a_0}$$

```
# Nondimensionalize all parameters by deep water condition
h_nd_exp = h_m_exp/SWL_m
```

```
W1_amp_nd_exp = W1_amp_m / a0_m[0]
W2_amp_nd_exp = W2_amp_m / a0_m[1]
W3_amp_nd_exp = W3_amp_m / a0_m[2]
W4_amp_nd_exp = W4_amp_m / a0_m[3]
```

The four experimental wave conditions were then plotted, as shown below.

```
plt.plot(h_nd_exp , W1_amp_nd_exp, 'ro', label = "Wave 1")
plt.plot(h_nd_exp , W2_amp_nd_exp, 'bo', label = "Wave 2")
plt.plot(h_nd_exp , W3_amp_nd_exp, 'mo', label = "Wave 3")
plt.plot(h_nd_exp , W4_amp_nd_exp, 'go', label = "Wave 4")
plt.xlabel('Nondimensional Water Depth', fontsize=14)
plt.ylabel('Nondimensional Shoaling Formula', fontsize=14)
plt.legend(loc='upper left', borderaxespad=0.)
plt.suptitle('Experimental', fontsize=18)
plt.savefig('W_amp_nd_exp.png')
plt.show()
```

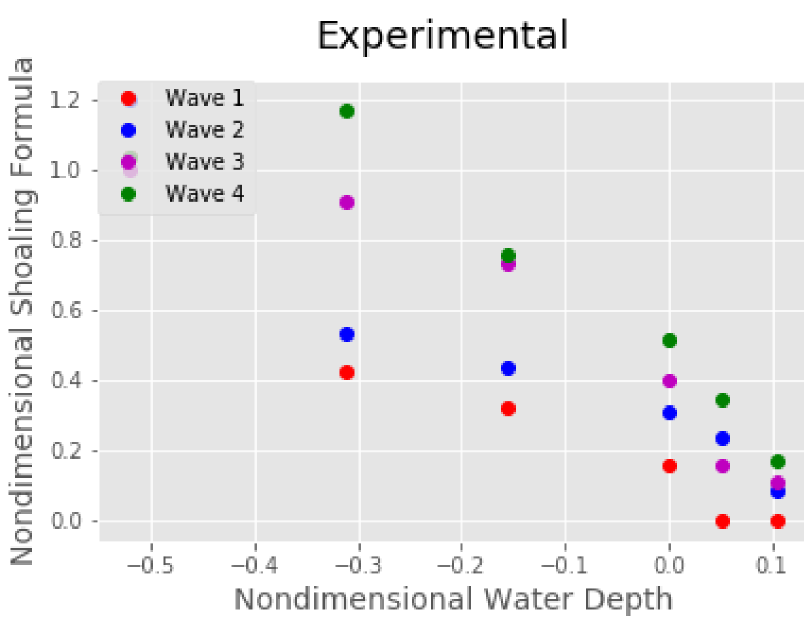


Figure 3:

Theoretical Analysis

Theoretical analysis was completed using the shoaling formula. First, experimentally found wavelengths and frequencies were used to determine C_{p0}. k_0 was found using the dispersion relationship, and was used to find n so that C_g could be determined.

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

$$C_{p0} = \frac{\lambda_0}{T}$$

$$n = \frac{1}{2}[1 + \frac{2kh}{sinh(2kh)}]$$

$$C_{g0} = n_0 C_{p0}$$

```
freq = np.array([1.25, 0.85, 0.5, 0.5])
T = 1/freq

W_lambda_cm = np.array([88.8, 148.75, 270.72, 270.72])
W_lambda_m = W_lambda_cm/100

Cp0_ms = W_lambda_m/T

k_theor_0 = np.array([wavenumber(T[0], SWL_m),
                      wavenumber(T[1], SWL_m),
                      wavenumber(T[2], SWL_m),
                      wavenumber(T[3], SWL_m)])

n_theor_0 = np.array([1/2 * (1 + (2 * k_theor_0[0] * SWL_m)/
                                    np.sinh(2 * k_theor_0[0] * SWL_m)),
                      1/2 * (1 + (2 * k_theor_0[1] * SWL_m)/
                                    np.sinh(2 * k_theor_0[1] * SWL_m)),
                      1/2 * (1 + (2 * k_theor_0[2] * SWL_m)/
                                    np.sinh(2 * k_theor_0[2] * SWL_m)),
                      1/2 * (1 + (2 * k_theor_0[3] * SWL_m)/
                                    np.sinh(2 * k_theor_0[3] * SWL_m))])

Cg0_ms = Cp0_ms * n_theor_0
```

All x-values from x = -1 m to x = 0.001 m was analyzed. x = 0 was not used because the mathematical analysis would have put a 0 in a denominator, so the approximation was made. X-distance was then converted to water depth using the same relationship of slope used for experimental analysis. k was calculated as a function of water depth over the x-distance range, which was used to calculate n at each depth.

$$n(x) = \frac{1}{2}[1 + \frac{2k(x)h(x)}{sinh(2k(x)h(x))}]$$

```
x_plot_m = np.linspace(-1, 0.001, num = 50)
h_m_theor = abs(x_plot_m) / 10
```

```
# Initialize a vector for k
k1_theor = np.ones(len(h_m_theor))
k2_theor = np.ones(len(h_m_theor))
k3_theor = np.ones(len(h_m_theor))
k4_theor = np.ones(len(h_m_theor))

# Calculate k for each x distance using water depth
for i in range(0, len(h_m_theor)):
  k1_theor[i] = wavenumber(T[0], h_m_theor[i])
  k2_theor[i] = wavenumber(T[1], h_m_theor[i])
  k3_theor[i] = wavenumber(T[2], h_m_theor[i])
  k4_theor[i] = wavenumber(T[3], h_m_theor[i])

# Calculate a new n value for each k and h term
n1_theor = 1/2 * (1 + (2 * k1_theor * h_m_theor)/
                        np.sinh(2 * k1_theor * h_m_theor))
n2_theor = 1/2 * (1 + (2 * k2_theor * h_m_theor)/
                        np.sinh(2 * k2_theor * h_m_theor))
n3_theor = 1/2 * (1 + (2 * k3_theor * h_m_theor)/
                        np.sinh(2 * k3_theor * h_m_theor))
n4_theor = 1/2 * (1 + (2 * k4_theor * h_m_theor)/
                        np.sinh(2 * k4_theor * h_m_theor))
```

The nondimensional form is calculated by

$$h_{ND} = \frac{h(x)}{h_0}$$

$$C_{p1} = 2\sqrt{(gh(x))}$$

$$C_{g1} = n_1 C_{p1}$$

$$a_{ND} = \sqrt{(\frac{C_{g0}}{C_{g1}})}$$

$$a_{ND} = \sqrt{(\frac{C_{g0}}{2n_1\sqrt{(gh(x))}})}$$

```
h_nd_theor = h_m_theor/SWL_m

W1_amp_nd_theor = np.sqrt(Cg0_ms[0])/
                      ((np.sqrt(2 * n1_theor))*(g.magnitude * h_m_theor)**(1/4))
W2_amp_nd_theor = np.sqrt(Cg0_ms[1])/
                      ((np.sqrt(2 * n2_theor))*(g.magnitude * h_m_theor)**(1/4))
W3_amp_nd_theor = np.sqrt(Cg0_ms[2])/
                      ((np.sqrt(2 * n2_theor))*(g.magnitude * h_m_theor)**(1/4))
W4_amp_nd_theor = np.sqrt(Cg0_ms[3])/
                      ((np.sqrt(2 * n2_theor))*(g.magnitude * h_m_theor)**(1/4))
```

The four theoretical conditons were then plotted, as shown below.

```
plt.plot(-h_nd_theor, W1_amp_nd_theor, label = "Wave 1")
plt.plot(-h_nd_theor, W2_amp_nd_theor, label = "Wave 2")
plt.plot(-h_nd_theor, W3_amp_nd_theor, label = "Wave 3")
plt.plot(-h_nd_theor, W4_amp_nd_theor, label = "Wave 4")
plt.xlabel('Nondimensional Water Depth', fontsize=14)
plt.ylabel('Nondimensional Shoaling Formula', fontsize=14)
plt.legend(loc='upper left', borderaxespad=0.)
plt.suptitle('Theoretical', fontsize=18)
plt.savefig('W_amp_nd_theor.png')
plt.show()
```

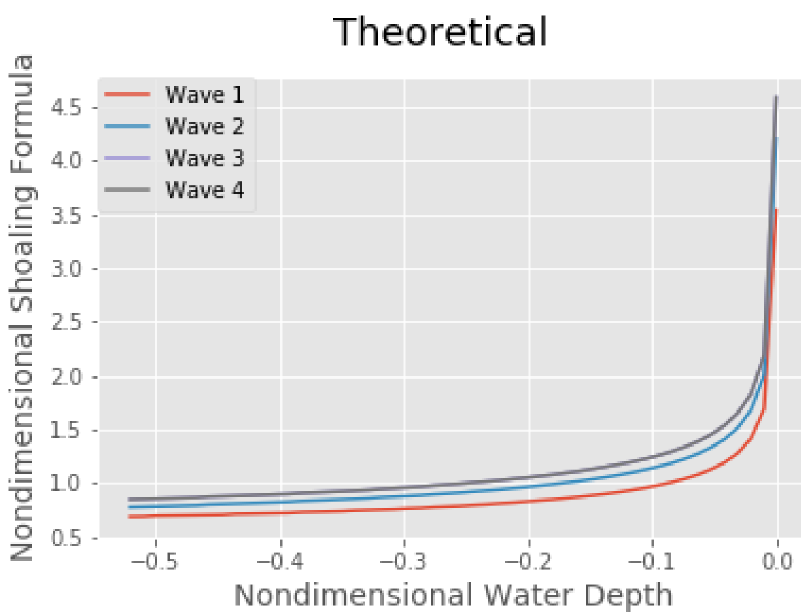


Figure 4:

Comparison of Experimental and Theoretical Results

The experimental data shows wave amplitude decreasing as the bed gets shallower (as x becomes positive). From the theoretical analysis, we would expect that amplitude increases in shallow water because as water depth decreases, amplitude increases and wavelength decreases. However, the shoaling formula does not hold when waves break, which happened during the experiment. The shoaling formula relies on the assumption that energy within the wave is conserved, but wave breaking dissipated energy, allowing the amplitude after wave break to decrease.

```python
# Only plot experimental data up to x = 0 m
h_nd_exp_comp = np.array([h_nd_exp[0], h_nd_exp[1], h_nd_exp[2], h_nd_exp[3]])
W1_amp_nd_exp_comp = np.array([W1_amp_nd_exp[0], W1_amp_nd_exp[1],
                              W1_amp_nd_exp[2], W1_amp_nd_exp[3]])
W2_amp_nd_exp_comp = np.array([W2_amp_nd_exp[0], W2_amp_nd_exp[1],
                              W2_amp_nd_exp[2], W2_amp_nd_exp[3]])
W3_amp_nd_exp_comp = np.array([W3_amp_nd_exp[0], W3_amp_nd_exp[1],
                              W3_amp_nd_exp[2], W3_amp_nd_exp[3]])
W4_amp_nd_exp_comp = np.array([W4_amp_nd_exp[0], W4_amp_nd_exp[1],
                              W4_amp_nd_exp[2], W4_amp_nd_exp[3]])

plt.plot(-h_nd_theor, W1_amp_nd_theor, label = "Wave 1")
plt.plot(-h_nd_theor, W2_amp_nd_theor, label = "Wave 2")
plt.plot(-h_nd_theor, W3_amp_nd_theor, label = "Wave 3")
plt.plot(-h_nd_theor, W4_amp_nd_theor, label = "Wave 4")
plt.plot(h_nd_exp_comp , W1_amp_nd_exp_comp, 'ro', label = "Wave 1")
plt.plot(h_nd_exp_comp , W2_amp_nd_exp_comp, 'bo', label = "Wave 2")
plt.plot(h_nd_exp_comp , W3_amp_nd_exp_comp, 'mo', label = "Wave 3")
plt.plot(h_nd_exp_comp , W4_amp_nd_exp_comp, 'go', label = "Wave 4")
plt.xlabel('Nondimensional Water Depth', fontsize=14)
plt.ylabel('Nondimensional Shoaling Formula', fontsize=14)
plt.legend(loc='upper left', borderaxespad=0.)
plt.suptitle('Comparison', fontsize=18)
plt.savefig('W_amp_nd_comp.png')
plt.show()
```

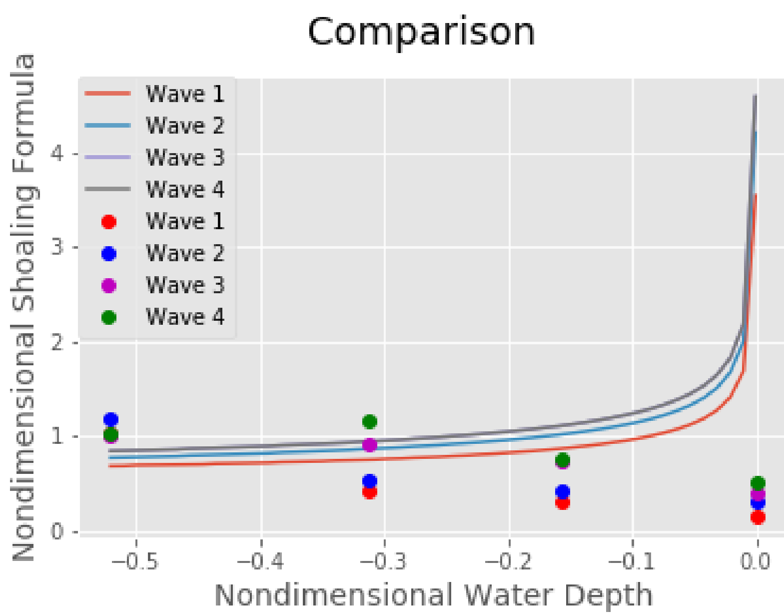
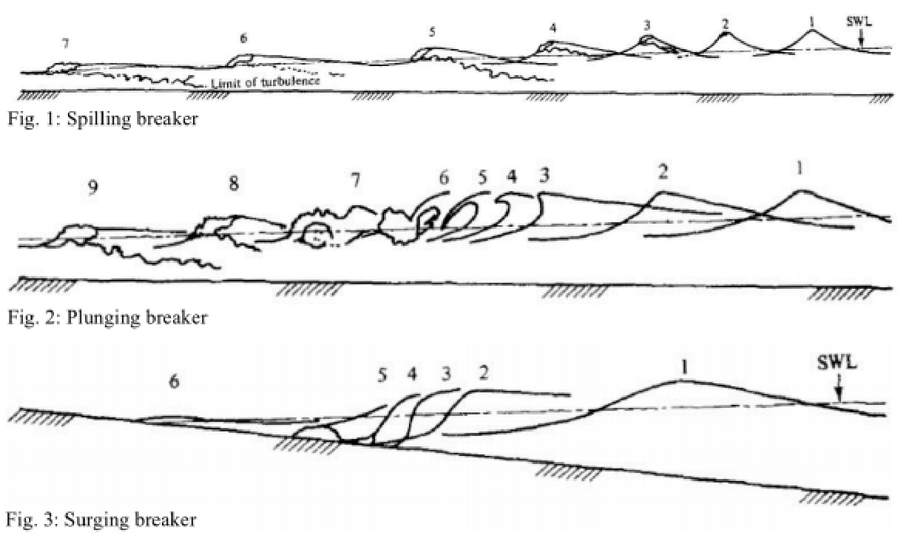


Figure 5:

Wave Breaker Types

Wave breaks can occur in three ways, as shown by the following diagram.



The Iribarren number is used to theoretically predict the type of wave break. Theoretical demarcations of the Iribarren number, ξ, are:

ξ	Breaker Type
$\xi < 0.5$	Spilling
$0.5 < \xi < 3.3$	Plunging
$\xi > 3.3$	Surging

The Iribarren number can be calculated using experimental data by

$$\xi = \frac{s}{\sqrt{(H_0/\lambda_0)}}$$

```
iribarren = slope/np.sqrt(H0_cm/W_lambda_cm)
```

Experimental results for breaker type are shown below. λ was measured experimentally and ξ was calculated using the above equation.

Case	λ_0 (cm)	ξ	Predicted Breaker Type	Observed breaker Type
W1	88.80	0.339	Spilling	Spilling
W2	148.75	0.458	Spilling	Plunging
W3	270.72	0.776	Plunging	Spilling
W4	270.72	0.966	Plunging	Surging

Experimental results for breaker location are shown below.

Case	h_b (cm)	H_b (cm)	$\frac{H_b}{h_b}$
W1	7.5	4.9	0.653
W2	6.4	5.7	0.891
W3	4.9	4	0.816
W4	3.5	3.5	1

Iribarren Number Analysis

Under the given lab conditions, it was unlikely that we would see surging waves because the scale was too small. Therefore, most of the waves that were observed were spilling and plunging, instead of surging. There was one observed surging wave, Wave 4, but during experimental testing it was unclear if it truly was surging or not. The Iribarren number suggested that the wave should be plunging instead of surging.

The Iribarren number increases with decreasing frequency. This can be seen across all waves. The Iribarren number is not related to the wave stroke.

```
# To convert the document from markdown to pdf
pandoc Lab_3.md -o Maisel_Lab_3.pdf
```