Zoe Maisel
Coastal Engineering CEE 4350
Lab 1: Wave Generation and Wave Gauge Calibration

Introduction:
An acoustic wave gauge was used to analyze multiple waves in time. The wave gauge uses voltage differences to measure the distance of the sensor to the wave water surface. By varying wavemaker paddle stroke sizes and wave frequency, the relationship between stroke, frequency, and wave amplitude is determinable. Wave amplitude varies with stroke at a fixed frequency because of mass conservation principles in accordance with linear wave theory.

Experimental Setup:
Sinusoidal waves were generated in a flume using a piston-type wavemaker. Acoustic wave gauge sensors were used at a constant position to determine wave height.

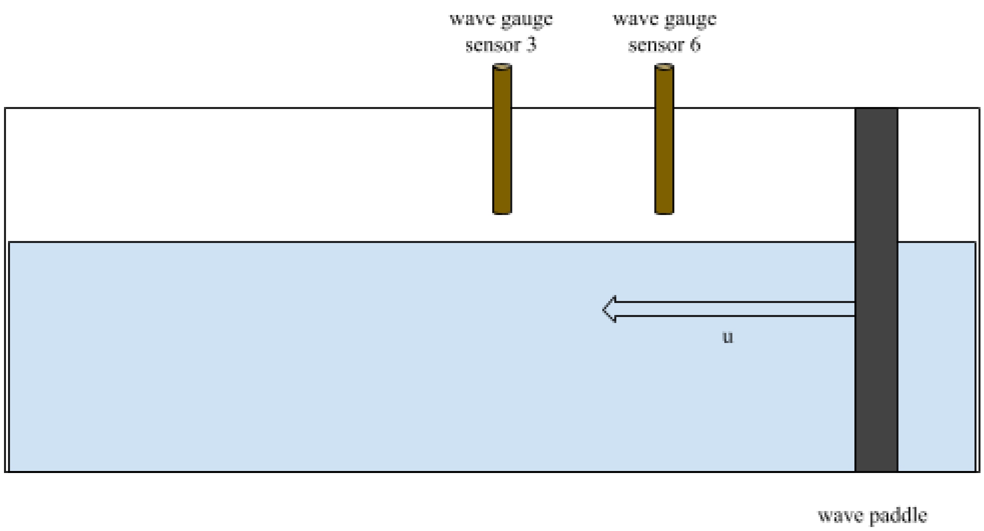

Figure 1: Experimental setup showing the hydraulic wave paddle, direction of wave propagation, and relative wave gauge sensor placements.

Five waves conditions were created using the wavemaker controller. The experimental conditions are detailed in Table 1.

Table 1: Experimental conditions for wave testing.

Naming Convention	Stroke (mm)	Frequency (Hz)
Data 1	3	2.0
Data 2	3	2.5
Data 3	3	3.0
Data 4	4	2.0
Data 5	5	2.0

R was used to analyze and plot experimental data.

Calibration:
The acoustic wave gauges record data as a voltage output using the time it takes for the sensor signal to leave the sensor, reach the water surface, and bounce back to the sensor. A wave gauge calibration can be done to determine the relationship between voltage and distance because the voltage output is a linear function of the distance from sensor to water surface. Calibration was determined by taking four voltage measurements in steps of 0.1 feet. Figure 2 shows the linear fit to the data points. The slope of the line was found to be 26.2 mm/V.

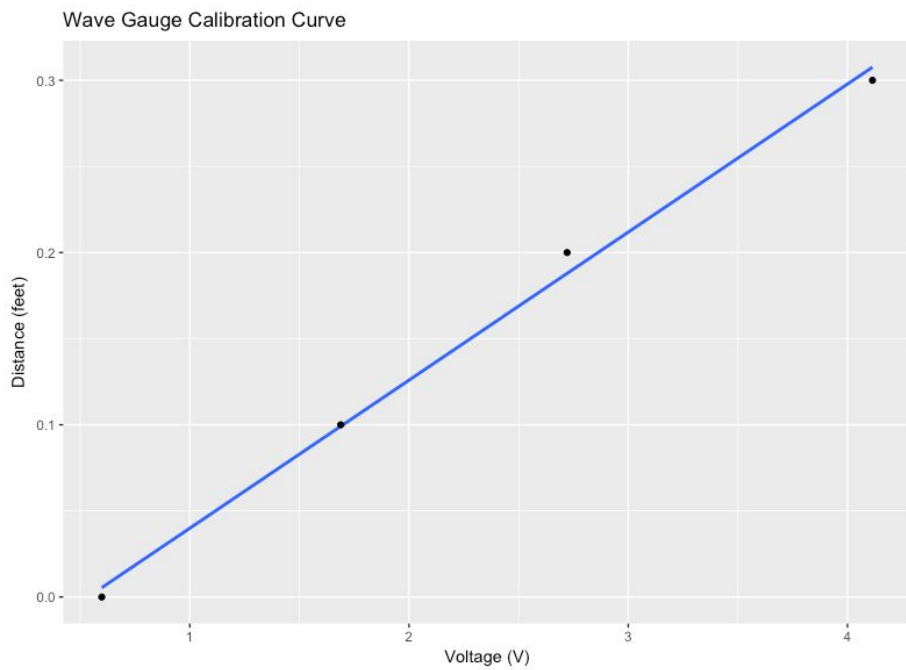

Figure 2: Wave gauge calibration to determine the relationship between voltage and distance.

Calibration was completed on a different sensor than the once used for wave data collection. The calibration for sensor 3 was given as 22.6 mm/V and for sensor 6 was given as 24.6 mm/V.

Wave Generation Results and Discussion:
Wave data collected during the experiment was loaded into R and analyzed. Data was recorded for 30 seconds at a sampling frequency of 100 Hz for both sensor 3 and 6. Sensor 3, the downstream sensor, was chosen to analyze the waves created at one horizontal distance.

To begin analysis, all voltage and time data were plotted in R. By visual inspection, Data 1, 2 and 4 all looked reasonable with no great outliers. Data 3 and 5 had large outliers and unexpected data that did not follow a sine wave pattern. Data 3 and 5 were both truncated to neglect the outlier data; after truncation there were still around 400 data points for analysis which was deemed sufficient. It is expected that the outlier data was a result of measurement and record error from the wavemaker, and from human interference (e.g. walking near sensor cables). After data was truncated, the given calibration value for sensor 3 as 22.6 mm/V was used to convert the data from volts to millimeters. Figure 3 shows the wave height as a function of time for 10.0 seconds of Data 4. A clear sine wave is evident which is expected for the wave.

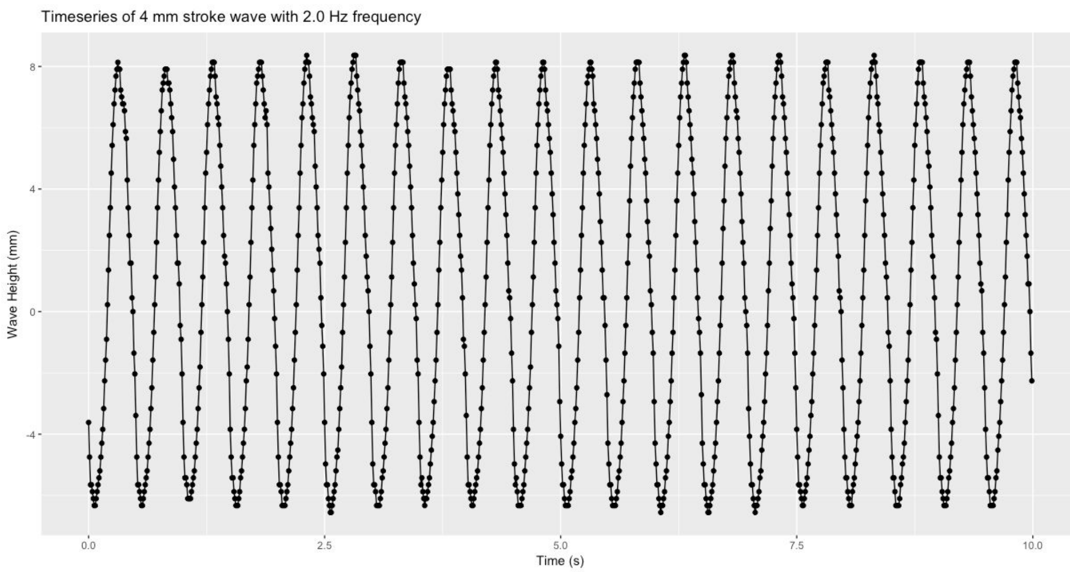

Figure 3: Wave height (mm) vs. time (s) for Data 4, with a paddle stroke of 4 mm at a 2.0 Hz frequency.

Wave amplitude was determined by taking the maximum wave height, minimum wave height, and then finding the difference. This difference is wave height (H) and divided by two gives wave amplitude. This analysis is a crude first pass because it does not take into account the fact that there may be outliers in maxima and minima that impact the calculation. However, because the data was already truncated to remove large errors, this amplitude calculation is sufficient to characterize the waves. The method likely overshoots the real amplitude because it takes the largest possible difference. Wave amplitude values are reported in Table 2.

Table 2: Experimental conditions for wave testing.

Naming Convention	Stroke (mm)	Frequency (Hz)	Amplitude (mm)
Data 1	3	2.0	5.42
Data 2	3	2.5	5.65
Data 3	3	3.0	5.42
Data 4	4	2.0	7.57
Data 5	5	2.0	9.15

Plots were made to show wave amplitude as a function of wavemaker frequency for a fixed stroke, and for a plot showing wave amplitude as a function of wavemaker stroke for a fixed frequency. Figure 4 shows the plot for fixed stroke and Figure 5 shows the plot for fixed frequency.

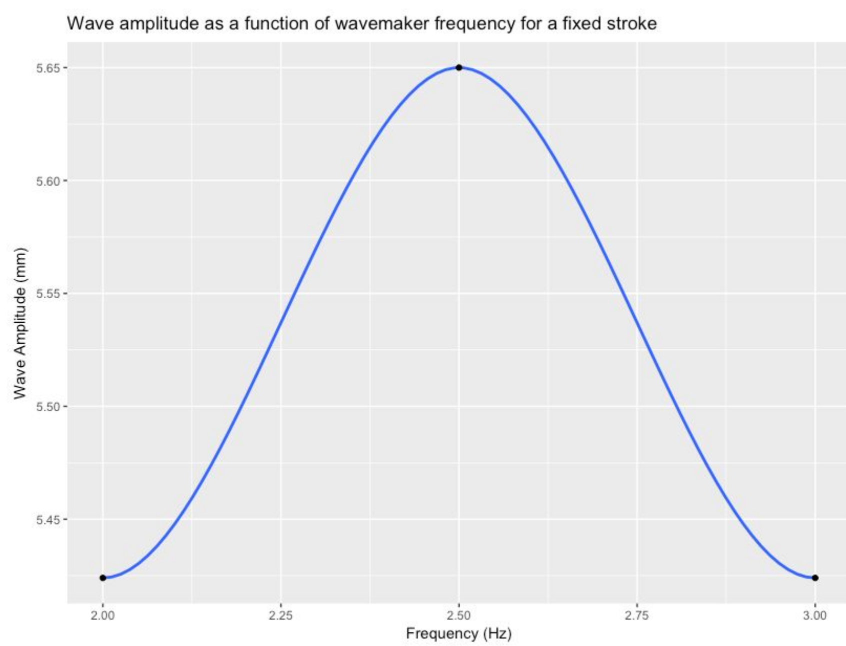

Figure 4: Wave amplitude as a function of wavemaker frequency for a fixed stroke

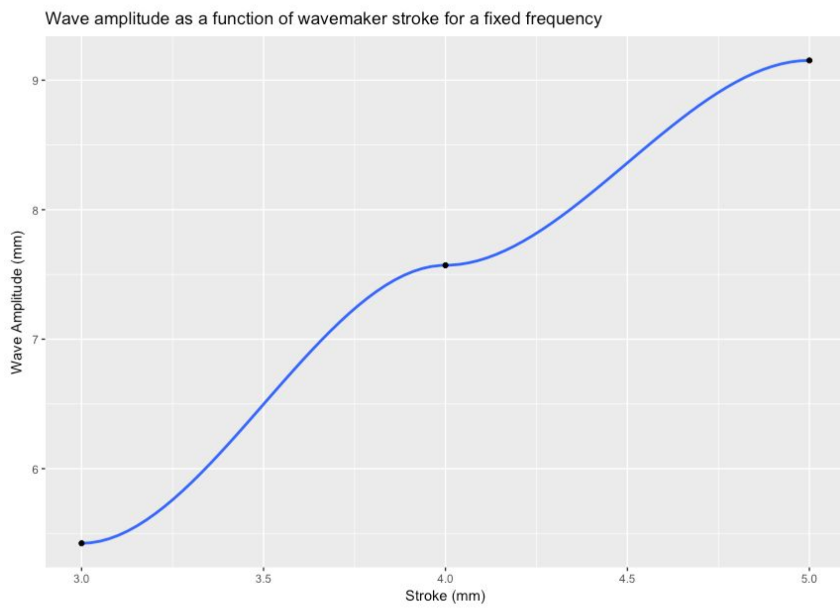

Figure 5: Wave amplitude as a function of wavemaker stroke for a fixed frequency

Wave amplitude as a function of wavemaker frequency for a fixed stroke appears to vary over different frequencies. However, the absolute difference between the amplitudes for the different cases are around 0.5 mm, which is not a large difference. Wave amplitude as a function of wavemaker stroke for a fixed frequency varies over different strokes over about a 4 mm range. This suggests that the relationship between amplitude and stroke dominates over the relationship between amplitude and frequency.

For linear wave theory, the equation for water surface is given below as:

$$\eta = a\, cos(kx) cos(\sigma t)$$

where η is the free water surface, a is wave amplitude, k is wavenumber, x is horizontal location, σ is the dispersion number, and t is time. The equation shows that the observed behavior of amplitude varying with stroke but not frequency is expected. Frequency only shows up inside the cosine terms for η, and because cosine only varies between 0 and 1, the influence is small. It physically makes sense that amplitude varies most significantly with stroke because at a fixed frequency, the volume of water displaced is greater for greater strokes; because that volume of water needs to be moved somewhere, the amplitude must increase.

Appendix:

R-code used for data analysis:

```
# Zoe Maisel
# Coastal Engineering
# Lab 1

setwd("~/github/CoastalEngineering")

# Load in gdata library to allow for xls read in
library(gdata)
library(ggplot2)

# Load in calibration data, remove unnecessary columns and rename columns
calib1 <- read.xls("calib1.xlsx")
calib1 <- calib1[-c(1), (-c(3,4))]
colnames(calib1) <- c("time_s","calibration_sensor_volts")

calib2 <- read.xls("calib2.xlsx")
calib2 <- calib2[-c(1), (-c(3,4))]
colnames(calib2) <- c("time_s","calibration_sensor_volts")

calib3 <- read.xls("calib3.xlsx")
calib3 <- calib3[-c(1), (-c(3,4))]
colnames(calib3) <- c("time_s","calibration_sensor_volts")

calib4 <- read.xls("calib4.xlsx")
calib4 <- calib4[-c(1), (-c(3,4))]
colnames(calib4) <- c("time_s","calibration_sensor_volts")

# Find all the means for the calibration data and fill it in a datafram
mean_calibs = data.frame(matrix(nrow = 4, ncol = 0))
mean_calibs$distance_ft = c(0.0, 0.1, 0.2, 0.3)
mean_calibs$volts = c(mean(as.numeric(as.character(calib4$calibration_sensor_volts))),
                    mean(as.numeric(as.character(calib3$calibration_sensor_volts))),
                    mean(as.numeric(as.character(calib2$calibration_sensor_volts))),
                    mean(as.numeric(as.character(calib1$calibration_sensor_volts))))

# Plot the calibration data
qplot(mean_calibs$volts, mean_calibs$distance_ft, xlab = "Voltage (V)", ylab = "Distance (feet)", main = "Wave
Gauge Calilbration Curve")
ggplot(mean_calibs, aes(x = mean_calibs$volts, y = mean_calibs$distance_ft)) +
geom_smooth(method='lm',formula=y~x, se = FALSE) + geom_point() +
  labs(title = 'Wave Gauge Calibration Curve', x = 'Voltage (V)', y = 'Distance (feet)')

calib_slope = coef(lm(mean_calibs$distance_ft ~ mean_calibs$volts,data=mean_calibs))[2]
calib_slope = -0.0859569  # feet/V
```

```
calib_slope = calib_slope*304.8 #mm/V

# Load in sensor data, delete extraneous points. Truncate data to have reasonable plots. Choose data from sensor 3 (downstream sensor)
# If all of the data is plotted, the curve is not the expected wave sine curve. The data was truncated to choose ranges that had reasonable sines
# Determine wave amplitude from voltage data
# Reset time to start at 0.0 s
# Use the given calibration of Sensor 3 which is 22.6 mm/V

calib_s3 = 22.6 # mm/V

# Keep all data rows
data1 <- read.csv("data1.csv")
data1 <- data1[-c(1), (-c(2,4))]
colnames(data1) <- c("time_s","sensor3_volts")
data1$wave_mm = data1$sensor3_volts * calib_s3
data1$time_s = data1$time_s - data1$time_s[1]
ggplot(data1, aes(x = time_s, y = wave_mm)) + geom_line() + geom_point() +
   labs(title = 'Timeseries of 3 mm stroke wave with 2.0 Hz frequency', x = 'Time (s)', y = 'Wave Height (mm)')

# Keep all data rows
data2 <- read.csv("data2.csv")
data2 <- data2[-c(1), (-c(2,4))]
colnames(data2) <- c("time_s","sensor3_volts")
data2$wave_mm = data2$sensor3_volts * calib_s3
data2$time_s = data2$time_s - data2$time_s[1]
ggplot(data2, aes(x = time_s, y = wave_mm))+geom_line()+geom_point()+
  labs(title = 'Timeseries of 3 mm stroke wave with 2.5 Hz frequency', x = 'Time (s)', y = 'Wave Height (mm)')

# Keep data rows 181 to 547
data3 <- read.csv("data3.csv")
data3 <- data3[-c(1), (-c(2,4))]
data3 <- data3[(181:547), ]
colnames(data3) <- c("time_s","sensor3_volts")
data3$wave_mm = data3$sensor3_volts * calib_s3
data3$time_s = data3$time_s - data3$time_s[1]
ggplot(data3, aes(x = time_s, y = wave_mm))+geom_line()+geom_point()+
  labs(title = 'Timeseries of 3 mm stroke wave with 3.0 Hz frequency', x = 'Time (s)', y = 'Wave Height (mm)')

# Keep all data rows
data4 <- read.csv("data4.csv")
data4 <- data4[-c(1), (-c(2,4))]
colnames(data4) <- c("time_s","sensor3_volts")
data4$wave_mm = data4$sensor3_volts * calib_s3
data4$time_s = data4$time_s - data4$time_s[1]
# Create a new dataframe
datashort = data.frame(matrix(nrow = 3, ncol = 0))
```

```
datashort = data4[(1:1000),]
ggplot(datashort, aes(x = time_s, y = wave_mm))+geom_line()+geom_point()+
  labs(title = 'Timeseries of 4 mm stroke wave with 2.0 Hz frequency', x = 'Time (s)', y = 'Wave Height (mm)')

# Keep data rows 1 to 1204
data5 <- read.csv("data5.csv")
data5 <- data5[-c(1), (-c(2,4))]
data5 <- data5[(1:1204), ]
colnames(data5) <- c("time_s","sensor3_volts")
data5$wave_mm = data5$sensor3_volts * calib_s3
data5$time_s = data5$time_s - data5$time_s[1]
ggplot(data5, aes(x = time_s, y = wave_mm))+geom_line()+geom_point()+
  labs(title = 'Timeseries of 5 mm stroke wave with 2.0 Hz frequency', x = 'Time (s)', y = 'Wave Height (mm)')

# Find the amplitude of each wave
wave1_amp_mm = (max(data1$wave_mm) + abs(min(data1$wave_mm))) / 2
wave2_amp_mm = (max(data2$wave_mm) + abs(min(data2$wave_mm))) / 2
wave3_amp_mm = (max(data3$wave_mm) + abs(min(data3$wave_mm))) / 2
wave4_amp_mm = (max(data4$wave_mm) + abs(min(data4$wave_mm))) / 2
wave5_amp_mm = (max(data5$wave_mm) + abs(min(data5$wave_mm))) / 2

# Create a dataframe with stroke (mm), frequency (Hz), and amplitude (mm) for each wave
exp_conditions = data.frame(matrix(nrow = 5, ncol = 0))
exp_conditions$stroke_mm = c(3, 3, 3, 4, 5)
exp_conditions$freq_Hz = c(2.0, 2.5, 3.0, 2.0, 2.0)
exp_conditions$amp_mm = c(wave1_amp_mm, wave2_amp_mm, wave3_amp_mm, wave4_amp_mm,
wave5_amp_mm)

# Create a dataframe for fixed stroke, using stroke = 3 mm
fixed_stroke = data.frame(matrix(nrow = 3, ncol = 0))
fixed_stroke$stroke_mm = c(3, 3, 3)
fixed_stroke$freq_Hz = c(2.0, 2.5, 3.0)
fixed_stroke$amp_mm = c(wave1_amp_mm, wave2_amp_mm, wave3_amp_mm)
ggplot(fixed_stroke, aes(x = freq_Hz, y = amp_mm))+geom_smooth()+geom_point()+
  labs(title = 'Wave amplitude as a function of wavemaker frequency for a fixed stroke', x = 'Frequency (Hz)', y =
'Wave Amplitude (mm)')

# Create a dataframe for fixed frequency, using frequency = 2.0 Hz
fixed_freq = data.frame(matrix(nrow = 3, ncol = 0))
fixed_freq$stroke_mm = c(3, 4, 5)
fixed_freq$freq_Hz = c(2.0, 2.0, 2.0)
fixed_freq$amp_mm = c(wave1_amp_mm, wave4_amp_mm, wave5_amp_mm)
ggplot(fixed_freq, aes(x = stroke_mm, y = amp_mm))+geom_smooth()+geom_point()+
  labs(title = 'Wave amplitude as a function of wavemaker stroke for a fixed frequency', x = 'Stroke (mm)', y = 'Wave
Amplitude (mm)', color = "black")
```