# Surface Wave Height Exceedance Probabilities on Two-Layer Fluids

by

Tom Redhead

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

BACHELOR OF SCIENCE

in the Department of Physics and Astronomy



University of Victoria

Surface Wave Height Exceedance Probabilities on Two-Layer Fluids

by

Tom Redhead

**Supervisors**

Dr. Johannes Gemmrich
Department of Physics and Astronomy

Dr. Adam Monahan
School of Earth and Ocean Sciences

# Abstract

Five sets of wave height spectra were used to generate surface profiles in a Monte Carlo simulation using a linear wave model. The sets of wave height spectra consisted of an initial spectrum, each with a different initial steepness, and a final spectrum after evolution on a two-layer fluid through class 3 resonant triads. Wave heights from the generated surface profiles were found to have reduced probabilities in four of five evolved wave height spectra when compared to wave heights generated by the initial spectra. The magnitude of the reductions in wave height exceedance probabilities were correlated, although not perfectly, to the initial steepness, and to the energy lost from the evolved spectrum relative to the initial spectrum. The exceedance probabilities of crest heights were found to not differ from Rayleigh distributed crest heights in the limit of large crest heights. Subsequently, the correlation between crest heights and trough depths was found to decrease with increasing initial steepness.

# Table of Contents

**Abstract** **i**

**Table of Contents** **ii**

**List of Figures** **iii**

**List of Tables** **v**

**Acknowledgements** **vi**

**1 Introduction** **1**
1.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

**2 Method and Results** **5**
2.1 Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

**3 Conclusion** **21**
3.1 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

**References** **23**

**Appendices** **24**

**A MATLAB Code for the Simulation** **25**
A.1 The Simulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
A.2 Exceedance Probabilities and Bootstrapping . . . . . . . . . . . . . . . . . . . 34

# List of Figures

1.1 Class 3 resonant triad. Two surface waves and an interfacial wave must propagate in the same direction. The wavenumbers of the two surface waves, $k_1$ and $k_2$, must be very close to each other in value, and the wavenumber of the interfacial wave, $k_3$ must be the difference of the two surface wave numbers. The density of the top layer, $\rho_1$ is less than the density of the lower layer, $\rho_2$ and the wave with wavenumber $k_3$ is the interfacial wave. . . . . . . . . . . . 3

2.1 Non-dimensional spectra before (blue) and after (pink) the two-layer evolution plotted in variance preserving form. The wavenumber, $k$, is on the horizontal axis scaled logarithmically, and the value of the spectrum multiplied by the wavenumber is on the vertical axis. These were the ten spectra used to generate the surface elevation profiles. The values $k_p$ and $A$ were the parameters of the spectra prior to the two-layer evolution. . . . . . . . . . . . . . . . . . 6

2.2 Energy lost from the surface wave field relative to the surface wave field energy of the initial spectrum, $\Gamma_s$ (Gemmrich & Monahan, 2021). . . . . . . . . . . 7

2.3 Exceedance probabilities of wave heights before (blue) and after (pink) the two-layer interaction. The wave height exceedance curve from Rayleigh distributed wave heights is shown in black. The error bars denote the range of uncertainties in the probability estimates determined by bootstrapping. The values of some of the exceedance probabilities, $P$, are shown on the right hand y-axis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

2.4 Relative change in wave height exceedance probabilities. Each curve was calculated by dividing the exceedance probabilities from the evolved spectrum ($P_2$) by the exceedance probabilities from the initial spectrum ($P_1$). The legend gives the values of $[k_p, A]$ and the colour indicates the initial steepness of the spectrum. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

2.5 Exceedance probabilities of crest heights before (blue) and after (pink) the two-layer interaction. The crest height exceedance curve from Rayleigh distributed crest heights is shown in black. The error bars denote the range of uncertainties in the probability estimates determined by bootstrapping. The values of some of the exceedance probabilities, $P$, are shown on the right hand y-axis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

2.6 Binned scatter plots of non-dimensional crest heights and trough depths before and after the two-layer interaction. Each subpanel also quotes the correlation coefficient, $r$, calculated from the corresponding crest height and trough depth distribution. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

# List of Tables

2.1 A comparison of the total number of waves generated from each case ($\mathrm{N} \times 10^6$), the largest wave height exceeded on the evolved surface (largest $\alpha$ exceeded), the corresponding probability from both the evolved and initial spectra ($P_2$ and $P_1$ respectively) and the initial spectral steepness. . . . . . . . . . . . . . . 14

# Acknowledgements

I would like to thank my supervisors, Dr. Johannes Gemmrich and Dr. Adam Monahan, for their support, suggestions and guidance over the course of this project.

# Chapter 1

# Introduction

## 1.1 Introduction

Large ocean waves can cause serious damage to ships, off-shore platforms, and smaller vessels. While predicting the exact time and place of a rogue wave is still not possible, understanding the occurrence frequency of large waves is important for maritime safety. Due to the unpredictable nature of individual wave heights, exceedance probabilities are the best way to assess the risk presented by an ocean state. A wave is defined as the profile of the surface elevation between two consecutive downward zero crossings, and a wave height is the distance from the peak (or crest) of the profile to the trough (Holthuijsen, 2010). A wave height exceedance probability is the probability that a given wave will exceed some value, often expressed as multiples of the significant wave height, $H_s$. The significant wave height is defined as the average of the largest third of the waves in the wave field (Holthuijsen, 2010). To make for a less cumbersome calculation, the significant wave height can also be calculated using an alternative definition

$$H_s = 4 \cdot \mathrm{std}\,(\eta) \tag{1.1}$$

with $\eta$ being the surface elevation of the wave field.

Rogue, or extreme, waves occur in the tail of a wave height probability distribution and are defined as having a wave height larger than $2.2H_s$. The crest height of a wave is the maximum point of the wave on the surface profile, and a trough depth is the minimum point of the wave on the surface profile. It is possible to have an extreme crest height, one which exceeds $1.25H_s$, that does not correspond to an extreme wave height. If the trough depth preceding an extreme crest was fairly shallow, the crest height would be measured as an extreme event, but the wave height may not be.

The occurrence rate of large waves depends on the shape of the wave height spectrum, with extreme waves becoming less likely as the bandwidth of the spectrum (the range of wavenumber values relevant to the spectrum) becomes larger (Gemmrich & Garrett, 2011). The wave height spectrum is the modulus squared of the Fourier transform of the surface elevation, and it contains all the statistics of the surface elevation assuming that the process is stationary and Gaussian (Holthuijsen, 2010). More specifically, the variance of the surface elevation is the integral of the wave height spectrum over its entire wavenumber domain.

In a two-layer fluid (for instance, freshwater on top of salty water) the density difference along the interface between the two layers can support internal waves. Interactions between waves on the surface and the density boundary between the two layers can change the shape of the wave height spectrum. A class 3 resonant triad (Figure 1.1), can allow energy to move from the surface to the interface between the two layers in the generation and maintenance of an interfacial wave. For this energy exchange to occur, two surface waves must propagate in the same direction with nearly equal wavenumbers to generate a third interfacial wave propagating in the same direction as the surface waves with wavenumber equal to the difference of the two surface wavenumbers. In addition to the wavenumber balance, the frequencies of the three wave must also balance (Alam, 2012). The energy exchange from the surface to the interface is a conservative process, but changes to the shape of the surface spectrum due to the interaction can enhance wave breaking which results in a net dissipation of surface wave energy (Gemmrich & Monahan, 2021). Additionally, due to the increased wave breaking and the two-layer effect, the affected spectrum can have the high frequency waves filtered out leading to a downshift in the spectral peak (Gemmrich & Monahan, 2021)

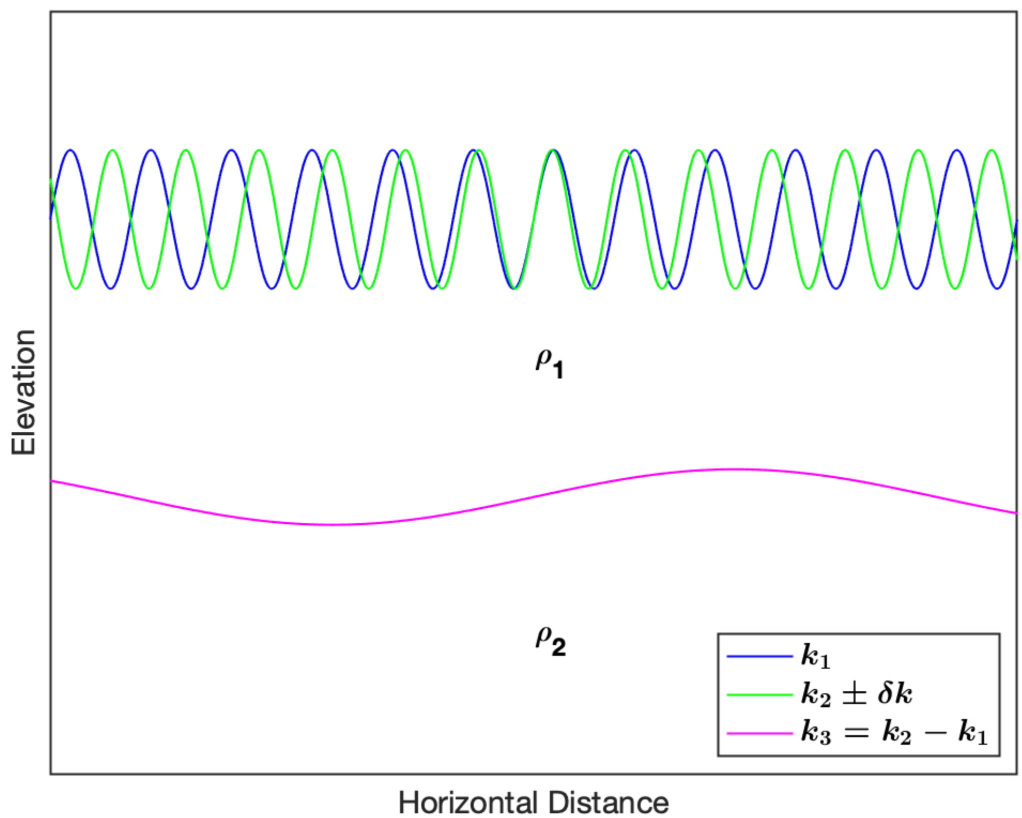


Figure 1.1: Class 3 resonant triad. Two surface waves and an interfacial wave must propagate in the same direction. The wavenumbers of the two surface waves, $k_1$ and $k_2$, must be very close to each other in value, and the wavenumber of the interfacial wave, $k_3$ must be the difference of the two surface wave numbers. The density of the top layer, $\rho_1$ is less than the density of the lower layer, $\rho_2$ and the wave with wavenumber $k_3$ is the interfacial wave.

Two-layer stratification is common in river estuaries due to fresh river water overlying salty ocean water, and in polar regions due to seasonal ice melt. Current significant wave heights in the Arctic Ocean are relatively small, with significant waveheights as large as 5 m now being observed (Thomson et al., 2018), the risk presented by extreme waves to large ships operating there is still low, but for local vessels wave heights exceeding 5 m could be very dangerous. In the Arctic Ocean, anthropogenic climate change is reducing the amount of sea ice present throughout the year. This means an increase in shipping through the region, and it also means that polar storms can generate larger significant wave heights through increased fetch due to the lack of ice covering the ocean surface.

In light of the spectral changes due to class 3 triads, the frequency of extreme waves on the surface of two-layer fluids may differ from those on homogeneous fluids. This research uses a linear wave model, spectra from Gemmrich and Monahan (2021), and Monte Carlo simulation to gain insight into the occurrence rates of large waves on two-layer fluids.

# Chapter 2

# Method and Results

## 2.1 Method

Monte Carlo simulation was the method used to determine if there are differences in wave height exceedance probabilities after spectral evolution on a two-layer fluid. One-dimensional surface profiles, generated from wave height spectra before and after the two-layer evolution, were used to compare wave height exceedance probabilities from the two cases. The initial wave height spectra used were Pierson-Moskowitz spectra represented by

$$S\left(k\right) = A\left(\frac{k}{k_p}\right)^{-3} \exp\left[-\frac{5}{4}\left(\frac{k}{k_p}\right)^{-2}\right] \tag{2.1}$$

where $A$ is a constant that determines the significant wave height, $k$ is the wavenumber, and $k_p$ is the peak wavenumber of the spectrum (Gemmrich & Monahan, 2021). The values of $k_p$ and $A$ determine the steepness of the initial spectrum, with larger values of $k_p$ and or $A$ corresponding to increasing steepness (Gemmrich & Monahan, 2021). Steepness is defined as the significant wave height multiplied by the average wavenumber of the bandwidth (Holthuijsen, 2010). Use of the Pierson-Moskowitz spectrum assumes a fully developed sea

(Pierson Jr & Moskowitz, 1964), meaning that the wind has blown steadily for a long time over a large area, and that the waves have reached a point of equilibrium with the wind.

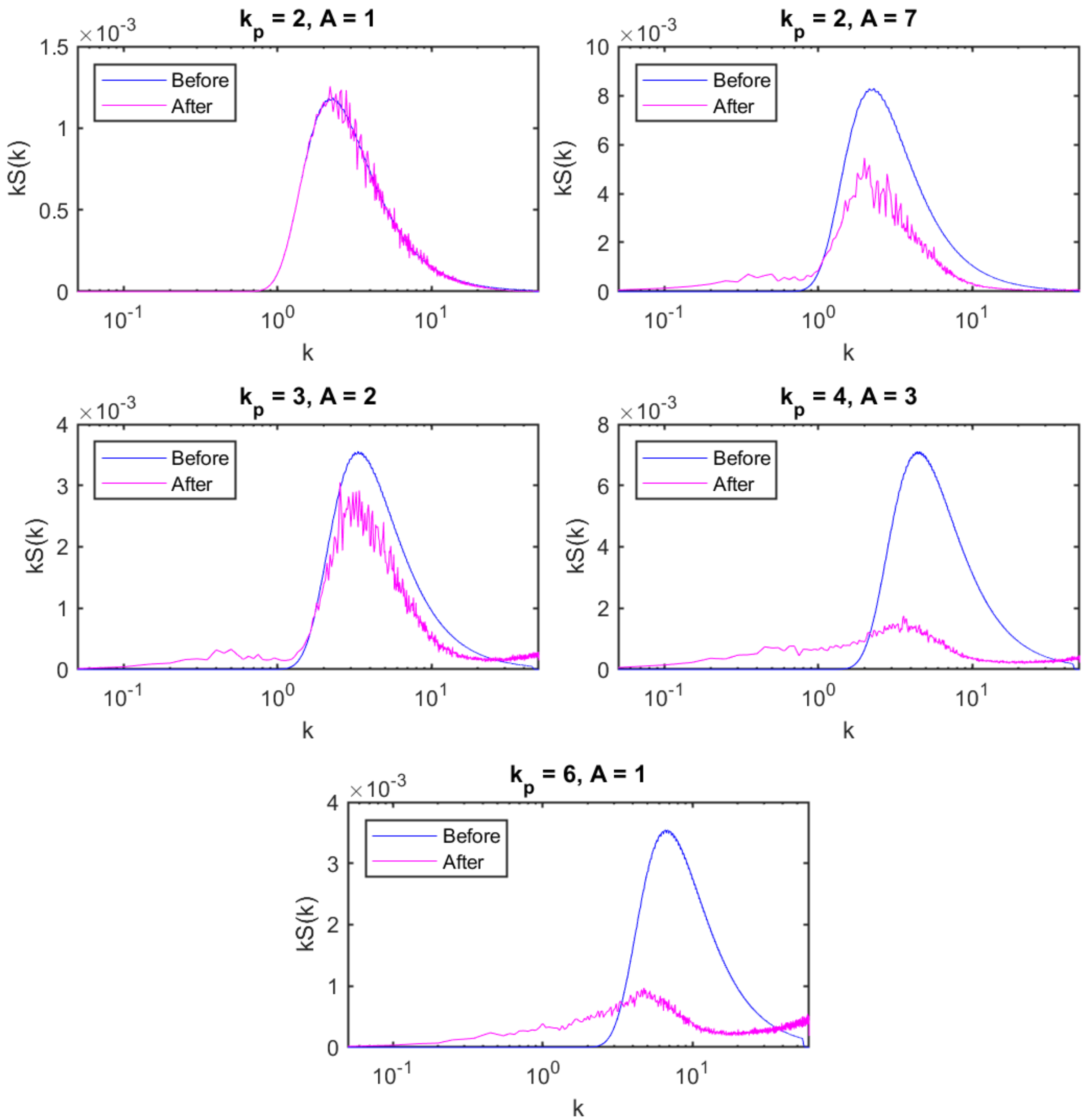


Figure 2.1: Non-dimensional spectra before (blue) and after (pink) the two-layer evolution plotted in variance preserving form. The wavenumber, $k$, is on the horizontal axis scaled logarithmically, and the value of the spectrum multiplied by the wavenumber is on the vertical axis. These were the ten spectra used to generate the surface elevation profiles. The values $k_p$ and $A$ were the parameters of the spectra prior to the two-layer evolution.

The ten spectra used (five pairs of initial and post-evolution spectra shown in Figure 2.1) were chosen as they each had lost different amounts energy relative to the initial surface wave field energy (Figure 2.2). The spectra used here had their evolution simulated prior to this simulation. The spectra were evolved on a two-layer fluid model with the density ratio, $R$, equal to 0.97, which is a realistic lower bound on density ratios in the ocean (Gemmrich & Monahan, 2021).

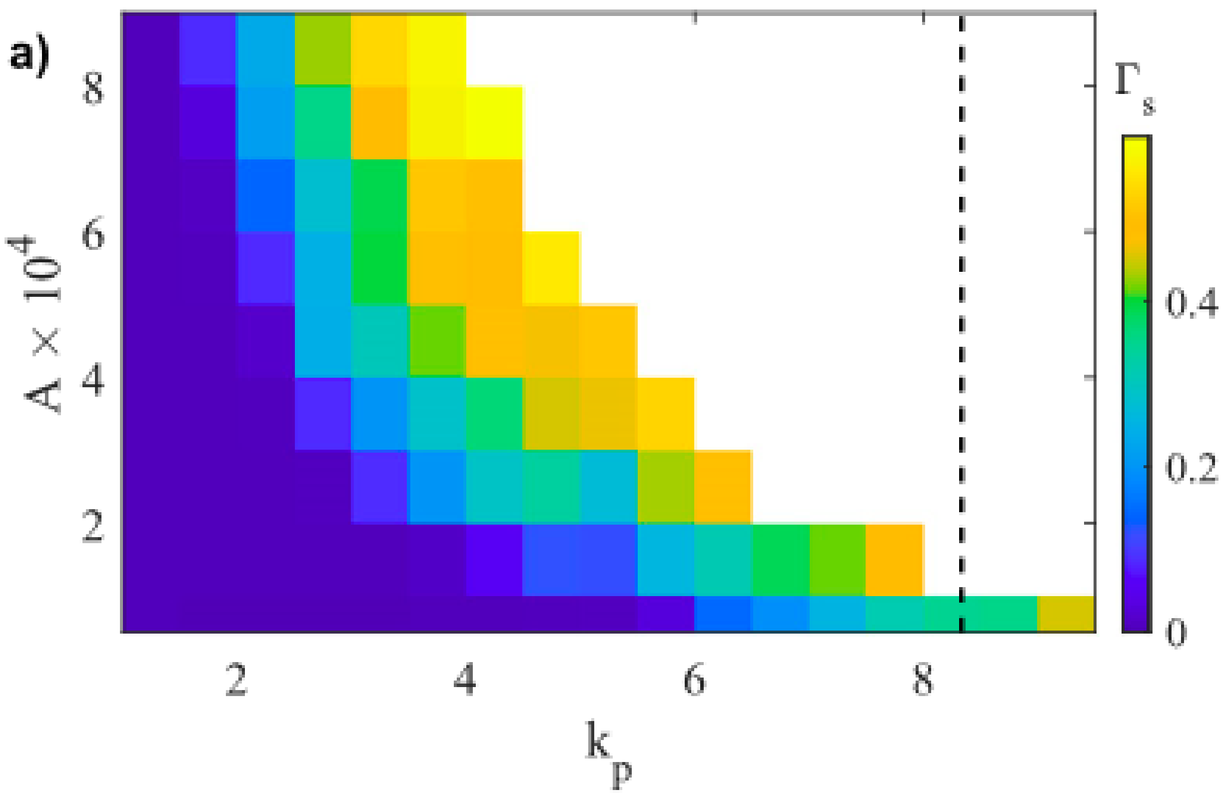


Figure 2.2: Energy lost from the surface wave field relative to the surface wave field energy of the initial spectrum, $\Gamma_s$ (Gemmrich & Monahan, 2021).

The model employed to simulate surface elevations was the random component model.

$$\eta(x) = \sum_{n=1}^{N_k} a_n \cos(k_n x) + b_n \sin(k_n x) \tag{2.2}$$

The coefficients, $a_n$ and $b_n$ were generated from MATLAB's normal random number generator `normrnd`, and values of $k_n = \frac{2\pi n}{L}$ were the discrete wavenumbers of the spectrum (with $L$ being the horizontal length of one surface profile realization). Each coefficient was generated with zero mean, and variance equal to the value of the spectrum at $k_n$. The same random seed was used for each spectrum to ensure that the simulation was reproducible.

The surface elevation needed to be Gaussian distributed so, by the central limit theorem, a relatively large number of coefficients was needed. However, it was desirable to have as small a number as possible for the coefficients to limit the computation time needed to generate each realization of the surface elevation. A value that was found to balance these two conditions was approximately $30,000$ values of each of $a_n$ and $b_n$. This number of coefficients was determined by generating a surface elevation with a specified number of coefficients, then using MATLAB's Anderson-Darling null-hypothesis test (`adtest`) to see if the generated surface elevation was consistent with a Gaussian distribution. The Andersion-Darling test determines if there is enough evidence to reject the null-hypothesis, in this case that the surface elevation was from a Gaussian distribution. The number of coefficients were adjusted until the approximate lowest number of coefficients that generated a distribution that could not be rejected as Gaussian was found.

With the coefficients generated, the surface elevation was then calculated for each realization. The generation of the surface elevation was the most computationally intensive part of the simulation. At each location of $x$ a total of 7 operations needed to be calculated. Thus, for a single set of coefficients, $a_i$, $b_i$, there would be $7N_x$ operations, where $N_x$ was the number of $x$ values. The number of $x$ values was set by the range of $x$ values and by the difference between each $x$ value, $\delta x$. In order to not miss information in each generated surface elevation, the value of $\delta x$ was set by Nyquist sampling theorem as $\delta x = \frac{\pi}{k_{max}}$ where $k_{max} = \frac{2\pi N_k}{L}$ was the Nyquist wavenumber, or maximum wavenumber, of the wave height spectrum (Bendat & Piersol, 2011). The range of $x$ values used in each simulation was $x \epsilon [0, 1000]$.

As extreme waves are very rare events (e.g. about one in every 10,000 waves is an extreme wave), one surface profile realization would not consistently generate enough large waves for statistically significant results in the extreme wave regime. As such, $10,000$ surface profile realizations were simulated from each of the 10 spectra (5 cases of before and after two-layer evolution).

To calculate wave height exceedance probabilities, wave heights needed to be extracted from the surface elevation. To find the wave heights, the zero crossings of the surface elevation needed to be found. Elevations at grid points on either side of a zero crossing were not necessarily close to zero, so interpolation between them needed to be done to find the position of the zero crossing. The interpolation was done globally with MATLAB's shape preserving interpolation function `pchip`. It should be noted that the interpolation reduced the variance of the surface elevation by a small amount. The analysis was not affected by the reduced variance as both cases of surface elevation (before the spectral evolution and after) were equally affected. The grid boxes containing the zero crossings were located by creating an array of the sign of the the surface elevation (-1 where the surface elevation was below zero and 1 where it was above zero), then adding to it the same array shifted to the right by one. The resulting array was populated by the values 0, 2, and $-2$. The grid boxes containing zero crossings were located at the indices populated by zeros. MATLAB's function `sign` was used to calculate the sign of the surface elevation.

To calculate the wave heights, the maximum and minimum values between successive downward zero crossings were found, with their difference being the wave height. Each wave height, crest height (maximum on the wave interval), and trough depth (minimum on the wave interval) were saved from each realization. The significant wave height was also saved from each realization as these were needed to calculate the exceedance probabilities. The exceedance probabilities were calculated relative to the significant wave height, $H_s$

$$P\left(H/H_s > \alpha\right) = \frac{\mathbf{N}\left(H/H_s > \alpha\right)}{\mathbf{N}(H)}. \tag{2.3}$$

In this calculation $H$ was the wave height, $\mathbf{N}(H/H_s > \alpha)$ was the number of wave heights relative to the significant wave height that were larger than $\alpha$, $\alpha \epsilon [0.5, 2.4]$, and $\mathbf{N}(H)$ was the total number of waves.

In order assess the significance of any difference between the exceedance probability curves, uncertainty estimates of the probabilities needed to be determined. In order to do this, a technique known as bootstrapping was employed. The set of wave heights from each spectrum was sampled with replacement to create new sets of surrogate wave height data from the original sets. In this way, 10,000 surrogate wave height data sets were created and exceedance probabilities were calculated from each of the new sets. The 75th and 25th percentiles of exceedance probability at each value of $\alpha$ were found. These values were used as the upper and lower bounds for the uncertainty range in exceedance probability comparisons.

The exceedance probability curves were plotted as $\ln(-\ln P)$ against $\ln\left(\frac{H}{H_s}\right)$, where $P = P(H/H_s > \alpha)$. Wave heights are randomly distributed according to a Rayliegh probability distribution provided that the random surface waves are distributed over a narrow wavenumber band (Longuet-Higgins, 1952). In this situation the exceedance probability of these wave heights is

$$P(H/H_s > \alpha) = \exp\left(-2\alpha^2\right) \tag{2.4}$$

(Gemmrich & Garrett, 2011). Taking the natural logarithm of equation 2.4 twice

$$\begin{aligned} \ln\left(-\ln P(H/H_s > \alpha)\right) &= \ln\left(2\alpha^2\right) \\ &= 2\ln(\alpha) + \ln(2) \end{aligned} \tag{2.5}$$

If the left hand side of equation 2.5 is plotted against $\ln(\alpha)$, the plot will appear as a straight line with a slope of 2 and an intercept of $\ln(2)$. Rayleigh distributed crest heights have an exceedance curve slightly modified from the Rayleigh wave height exceedance curve

$$P(\eta/H_s > \alpha) = \exp\left(-8\alpha^2\right) \tag{2.6}$$

Following the same line of reasoning as in the wave height case, the crest height exceedance curve would also have a slope of 2, but the intercept would now be $\ln(8)$. While the simulated wave heights and crest heights are not expected to be exactly Rayleigh distributed, looking at the exceedance curves in this way shows much more detail in the tail end of the exceedance curves and also provides an easy comparison to Rayleigh distributed wave heights and crest heights.

## 2.2 Results

For each of the different pairs of $k_p$ and $A$ considered, the wave height exceedance probabilities were different from a Rayleigh distributed exceedance curve before and after the two-layer interaction (Figure 2.3). The difference between the Rayleigh exceedance curve and those of the initial spectra was likely due to the wavenumber bandwidth being insufficiently narrow.

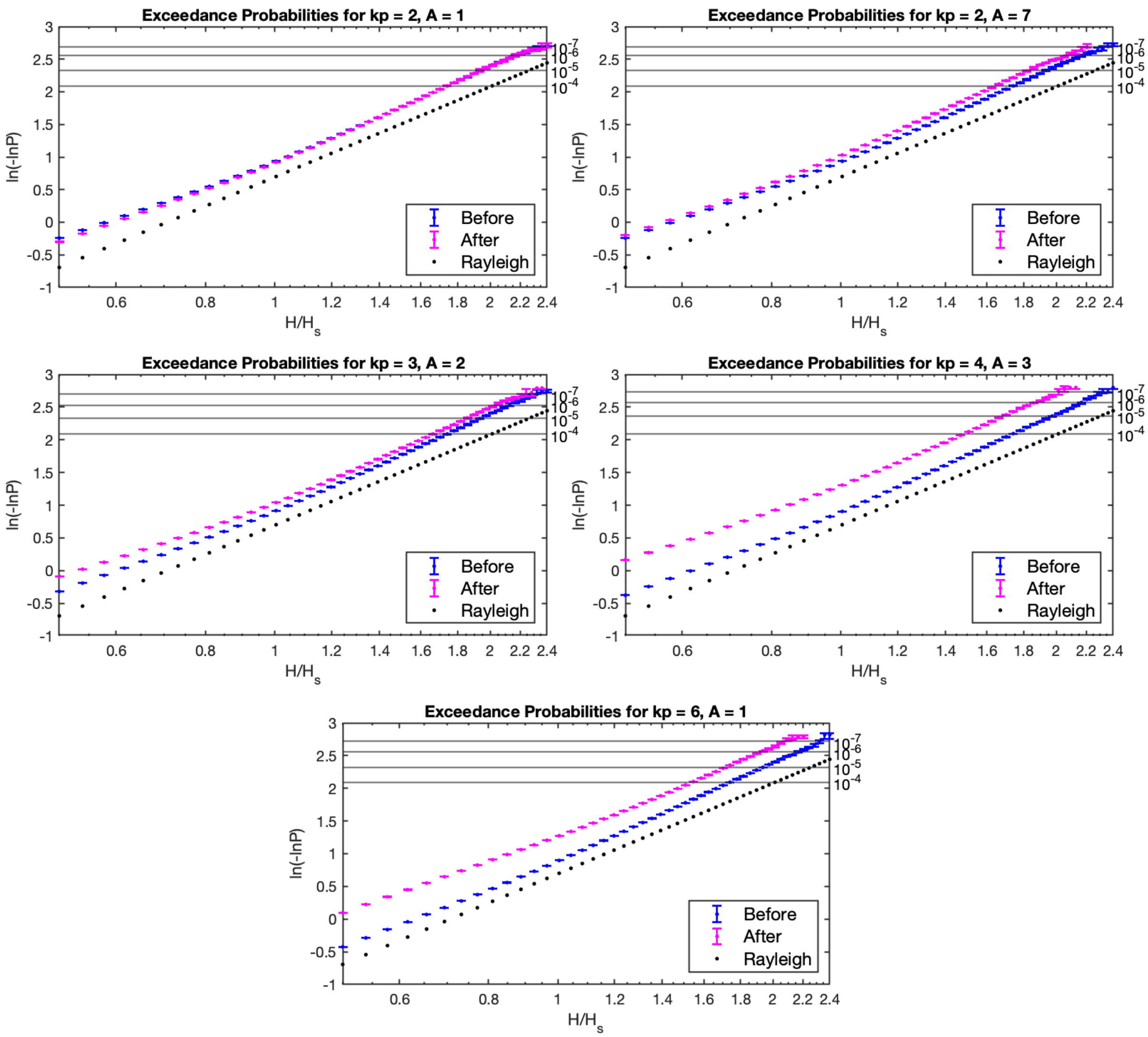


Figure 2.3: Exceedance probabilities of wave heights before (blue) and after (pink) the two-layer interaction. The wave height exceedance curve from Rayleigh distributed wave heights is shown in black. The error bars denote the range of uncertainties in the probability estimates determined by bootstrapping. The values of some of the exceedance probabilities, $P$, are shown on the right hand y-axis.

For each of the five $k_p$ and $A$ cases (individually denoted $[k_p, A]$), the exceedance curves both before and after two-layer evolution share a characteristic shape; the exceedance curves had slopes less than 2 for $\frac{H}{H_s} \lesssim 1.2$, and slopes greater than 2 while $\frac{H}{H_s} \gtrsim 1.2$.

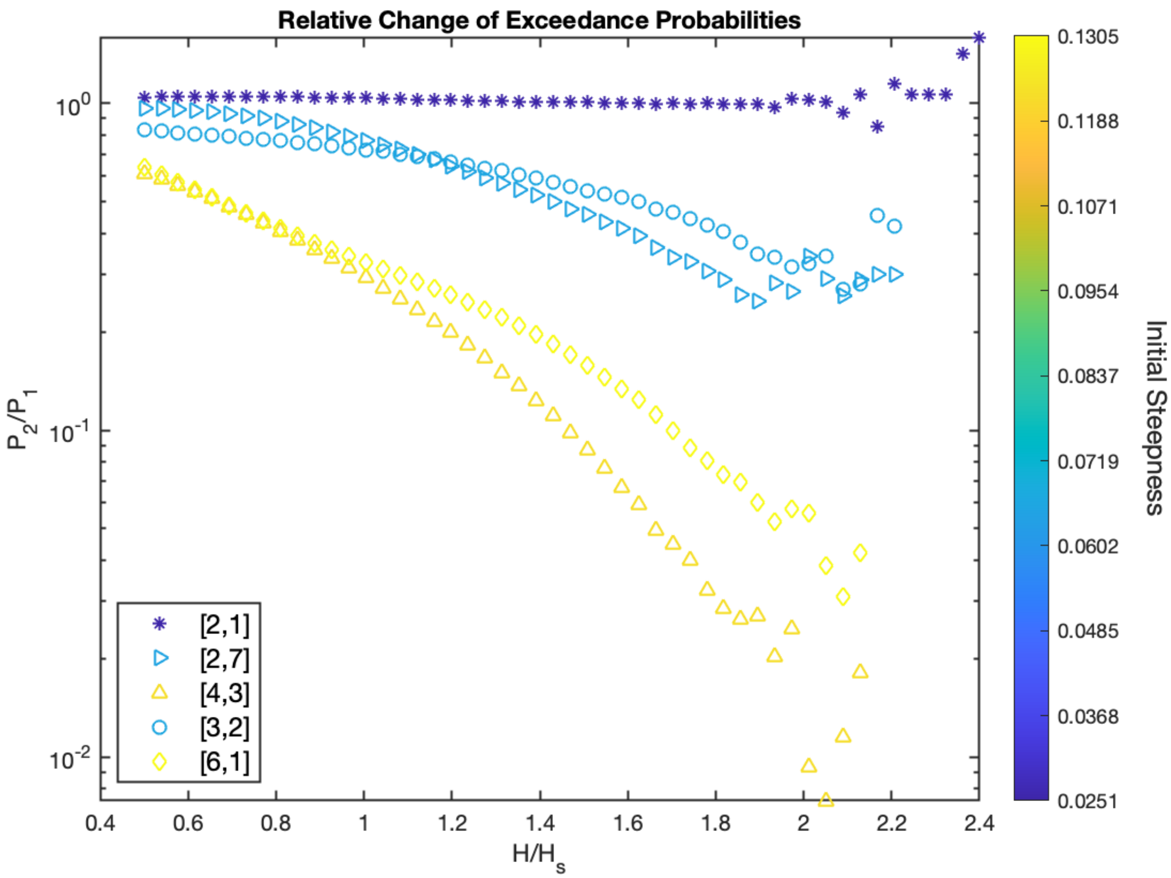


Figure 2.4: Relative change in wave height exceedance probabilities. Each curve was calculated by dividing the exceedance probabilities from the evolved spectrum ($P_2$) by the exceedance probabilities from the initial spectrum ($P_1$). The legend gives the values of $[k_p, A]$ and the colour indicates the initial steepness of the spectrum.

In the case of $[k_p, A] = [2, 1]$ (figure 2.3 top left), the wave height exceedance curves before and after the two-layer evolution differed by only small amounts. For small wave heights, or high probability events, the exceedance probabilities generated by the evolved spectrum were slightly higher than those generated by the initial spectrum. The results from case $[2, 1]$ were significant in the small wave regime, $H < H_s$, since the error bars representing the uncertainty range did not overlap here. The two exceedance curves converged at approximately $\frac{H}{H_s} = 1$, and apparent differences for $\frac{H}{H_s} > 1$ were within both of the curves uncertainty range. As such, the $[2, 1]$ case showed no significant difference in occurrence rates of large wave heights. Figure 2.4 shows that the ratio of the post evolution exceedance probabilities to the initial exceedance probabilities was close to 1 for the entire range of probabilities calculated. The variability in the ratios for large $\frac{H}{H_s}$ were due to sampling variability. This result provided further confirmation of the limited effect the spectral evolution had on the exceedance probabilities in the $[2, 1]$ case.

| | | Evolved | | | Initial | |
|---|---|---|---|---|---|---|
| Case | Steepness | Largest $\alpha$ Exceeded | $\mathrm{N} \cdot 10^6$ | $P_2$ | $\mathrm{N} \cdot 10^6$ | $P_1$ |
| $k_p = 2\ A = 1$ | 0.0251 | 2.4 | 7.41 | $4.05 \cdot 10^{-7}$ | 7.88 | $2.54 \cdot 10^{-7}$ |
| $k_p = 2\ A = 7$ | 0.0664 | 2.2 | 6.58 | $4.56 \cdot 10^{-7}$ | 7.88 | $1.52 \cdot 10^{-6}$ |
| $k_p = 3\ A = 2$ | 0.0652 | 2.4 | 11.9 | $8.38 \cdot 10^{-8}$ | 10.9 | $1.84 \cdot 10^{-7}$ |
| $k_p = 4\ A = 3$ | 0.1230 | 2.1 | 13.4 | $7.49 \cdot 10^{-8}$ | 13.6 | $4.13 \cdot 10^{-6}$ |
| $k_p = 6\ A = 1$ | 0.1305 | 2.2 | 22.0 | $9.11 \cdot 10^{-8}$ | 19.2 | $1.88 \cdot 10^{-6}$ |

Table 2.1: A comparison of the total number of waves generated from each case ($\mathrm{N} \times 10^6$), the largest wave height exceeded on the evolved surface (largest $\alpha$ exceeded), the corresponding probability from both the evolved and initial spectra ($P_2$ and $P_1$ respectively) and the initial spectral steepness.

The cases $[2, 7]$ and $[3, 2]$ were quite similar to each other. Both of them showed slightly reduced probabilities of exceeding any given wave height over the entire range of $\frac{H}{H_s}$ values (Figure 2.3). It is difficult to tell which of these two cases had a greater change in exceedance probabilities from Figure 2.3. Figure 2.4 showed that neither case was wholly more effected than the other. In the small wave regime the exceedance probabilities from case $[3, 2]$ experienced greater relative change than those from case $[2, 7]$. However this difference reversed after moving into the large wave regime, $H > H_s$, with the greater relative change belonging to the $[2, 7]$ case. In the $[3, 2]$ and $[2, 7]$ cases, the largest waves became less than half as common as in the initial spectrum (Figure 2.4).

Interestingly, the $[2,7]$ evolved spectrum only generated wave heights that exceeded $2.2H_s$ with a probability of $P = 4.56 \times 10^{-7}$, whereas the $[3,2]$ evolved spectrum generated some wave heights that exceeded $2.4H_s$ (Table 2.1). It could be that the simulation of the $[2,7]$ evolved spectrum did not generate enough waves to exceed $2.2H_s$. These extreme events are very rare, and most subject to sampling variability in any finite simulation. This variability is clear in Figure 2.4 with smooth behaviour in the small wave regime, and with increasingly random fluctuations moving into the extreme wave regime. Referencing Table 2.1, only 6.58 million waves were generated from this case whereas the $[2,1]$ evolved spectrum generated 7.41 million waves, while the $[3,2]$ evolved spectrum generate 11.9 million waves. In fact, of all the evolved spectra, the $[2,7]$ case generated the least waves. The number of waves generated by a spectrum depends on the peak wavenumber, the bandwidth, and the width of the domain. Spectra with wider bandwidths can generate a large number of relatively small waves; and smaller peak wavenumbers, which correspond with larger wavelengths due to their inverse relationship, generate fewer waves as fewer wavelengths fit into a given distance. Since an effect of the two-layer evolution was to downshift the spectral peak wavenumber, one might expect that the number of waves would decrease post spectral evolution in each of the cases. Referencing Table 2.1 shows that this did not always happen; cases $[6,1]$ and $[3,2]$ increased their wave counts post spectral evolution.

The exceedance curves generated from the $[4,3]$ and $[6,1]$ spectra showed much more dramatic changes in exceedance probabilities to those generated by the $[2,7]$, and $[3,2]$ spectra. The largest wave height exceeded by the $[4,3]$ evolved spectra was $2.1H_s$ with a probability of $P = 7.49 \times 10^{-8}$ from 13.4 million waves. The $[6,1]$ evolved spectra generated 22.0 million waves where the largest wave height exceeded was $2.2H_s$ with a probability of $P = 9.11 \times 10^{-8}$ (Table 2.1). Of these two cases, $[4,3]$ experienced the largest relative change in wave height exceedance probabilities, with extreme wave heights becoming less than a hundredth as common as those from the initial spectrum (Figure 2.4).

As seen in Figure 2.3 and Figure 2.4, all of the cases other than case $[2, 1]$ showed reduced exceedance probabilities after spectral evolution on a two-layer fluid. The most affected was the $[4, 3]$ case, with case $[6, 1]$ the second most affected. These two cases had by far the greatest initial steepness, while the second most affected, cases $[3, 2]$ and $[2, 7]$, also had larger initial steepness than the least affected (Table 2.1). Referring to Figure 2.4, the magnitude of the exceedance probability decrease could be grouped by initial steepness, where cases $[6, 1]$ and $[4, 3]$ shared very similar behaviour in both the large wave and small wave regimes. As shown in Table 2.1, these two cases differed in initial steepness by a ratio of 0.94. Cases $[3, 2]$ and $[2, 7]$ could also be grouped by initial steepness and relative change in exceedance probabilities. The relative change curves of these two cases were very close to each other in magnitude and, in terms of initial steepness, they were the closest to each other of all the cases simulated with the ratio of their steepness being 0.98. The mostly unaffected case, $[2, 1]$, had the smallest initial steepness, with the ratio between $[3, 2]$ steepness (the case with the second smallest initial steepness) and $[2, 1]$ steepness being 0.38. The minimal changes in exceedance probabilities in the $[2, 1]$ case could also be attributed to the small differences between the evolved spectrum and the initial spectrum in this case (Figure 2.1). Additionally, referring to Figures 2.2 and 2.4 ,the energy lost relative to the initial spectrum was correlated with the relative decrease in exceedance probabilities. Case $[4, 3]$ and $[6, 1]$ lost the most and second most energy relative to the initial spectrum respectively. Case $[2, 7]$ lost more energy relative to the initial spectrum than case $[3, 2]$ and, at least in the large wave regime, had larger relative reductions in exceedance probabilities.

The reduced probabilities of extreme wave heights after two-layer spectral evolution motivated investigation of the crest height and trough depth exceedance probabilities (Figure 2.5). In the small crest regime, $\eta < 0.5H_s$, the exceedance probabilities of the crest heights are similar to the corresponding wave height exceedance probabilities in each case. However, moving into the large crest regime, the exceedance probabilities calculated from both before and after the spectral evolution converge to the Rayleigh exceedance curve. The plots of trough depth exceedance probabilities (not shown) were indistinguishable from the crest height exceedance probabilities.

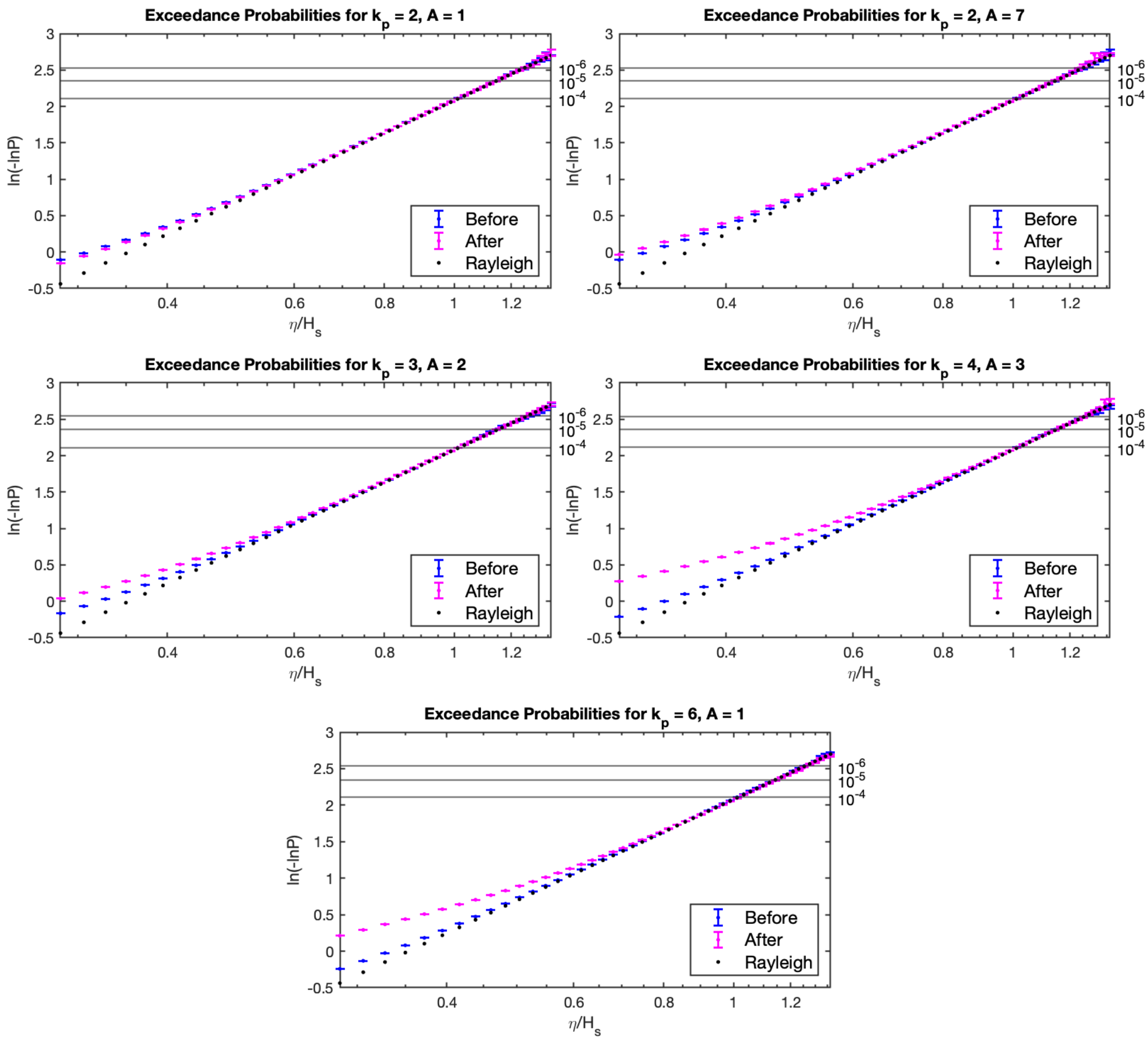


Figure 2.5: Exceedance probabilities of crest heights before (blue) and after (pink) the two-layer interaction. The crest height exceedance curve from Rayleigh distributed crest heights is shown in black. The error bars denote the range of uncertainties in the probability estimates determined by bootstrapping. The values of some of the exceedance probabilities, $P$, are shown on the right hand y-axis.

If the crest heights were strongly correlated with the trough depths a small crest height would also have a small trough depth in most cases. Since a wave height is just the sum of the crest height and the trough depth, a small crest height and small trough depth would equal a small wave height. As such, the only way for crest heights to show such different behaviour from the corresponding wave heights is if the crest heights became decoupled from the trough depths.

Figure 2.6 shows binned scatter plots of crest height and trough depth for the various cases considered. As seen in Figure 2.6, the correlation coefficient was approximately $r = 0.33$ for crests and troughs generated by the initial spectra. The probability of a large wave occurring, assuming a sufficiently narrow bandwidth, is larger for strongly correlated crest heights and trough depths (Casas-Prat & Holthuijsen, 2010). Additionally, the Rayleigh exceedance curve is calculated assuming that each wave height is twice the corresponding crest height (Casas-Prat & Holthuijsen, 2010). This assumption is equivalent to perfectly correlated crest heights and trough depths. The following correction to Rayleigh distributed exceedance curves, derived by Naess (1985), accounts for the correlation of crests and troughs assuming a narrow bandwidth.

$$P(H/H_s > \alpha) = \exp\left[-\frac{2\alpha^2}{\beta}\right] \tag{2.7}$$

Here $\beta = \frac{1}{2}(1 - \tilde{r})$ with $\tilde{r}$ being the crest-trough correlation. In the definition of $\tilde{r}$, trough depths are taken to be negative; for perfectly correlated crests and troughs $\tilde{r} = -1$, resulting in equation 2.4 (Naess, 1985). When $\alpha$ is plotted against $\ln(-\ln P)$, the slope of 2 remains the same, but the intercept becomes $\ln\left(\frac{2}{\beta}\right)$. As the crest-trough correlation decreases $\beta \to \frac{1}{2}$, and the intercept increases. The increase in intercepts seen in Figure 2.3 and the related decrease in crest-trough correlations are qualitatively consistent with the crest-trough correlation correction from Naess (1985).

The correlations of the crests and troughs from the initial spectra were not particularly strong; this result may provide further explanation as to how the exceedance probabilities generated from the initial spectra were reduced from the theoretical Rayleigh exceedance probabilities.

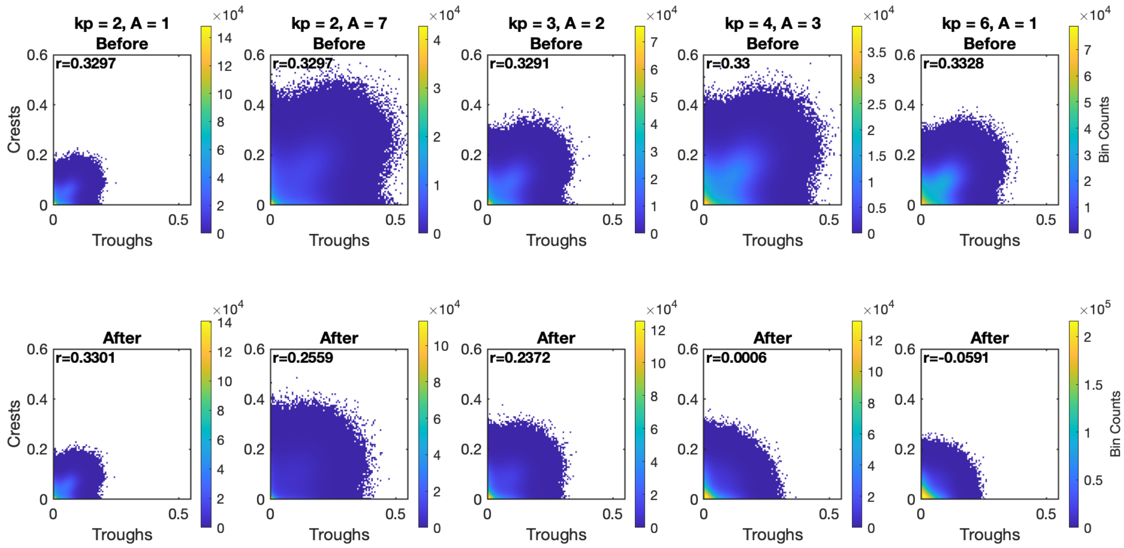


Figure 2.6: Binned scatter plots of non-dimensional crest heights and trough depths before and after the two-layer interaction. Each subpanel also quotes the correlation coefficient, $r$, calculated from the corresponding crest height and trough depth distribution.

After the two-layer spectral evolution each case, other than $[2, 1]$, showed a reduction in crest-trough correlation. The case with the smallest change in crest-trough correlation was $[2, 1]$, which also presented the smallest change in exceedance probabilities and exhibited the least relative energy loss. In this case the value of the correlation coefficient increased slightly. Recall that in the small wave regime case $[2, 1]$ showed slight, but significant, increases in wave height exceedance probabilities and that this case also had the smallest initial steepness. Cases $[2, 7]$ and $[3, 2]$ both showed larger decreases of crest-trough correlations, with $[2, 7]$ having a somewhat larger change. These two cases had very similar relative changes in wave height exceedance probabilities and both lost similar amounts of energy relative to their initial spectra with case $[2, 7]$ having lost slightly more. Recall that these cases had very similar initial steepness with case $[3, 2]$ having slightly smaller initial steepness of the two. This result is not fully consistent with the overall pattern of greater initial steepness being the sole predictor for decreases in crest-trough correlations. However, the difference in initial steepness between these two cases is very small, and as such the random nature of this simulation may be the cause of the lack of correlation between initial steepness and reduction in crest-trough correlations in these two cases. The smallest correlation, $r = 0.0006$, which was likely not meaningfully different from zero, came from the evolved $[4, 3]$ spectrum, which also had the greatest reduction in wave height exceedance probabilities and exhibited the greatest energy loss relative to the initial spectrum. The correlation coefficient from the evolved $[6, 1]$ spectrum was $r = -0.0591$ which was also likely not meaningfully different from zero. These two cases had the largest initial steepness and they both showed the most extreme reductions in crest-trough correlations. The reduction in exceedance probabilities after the two layer evolution is consistent, although not perfectly correlated, with the decrease in crest-trough correlations. Likewise, the reduction in crest-trough correlation is consistent, but not perfectly correlated, with initial steepness. As with the wave height exceedance probabilities, the energy lost relative to the initial spectrum (Figure 2.2) was correlated with the decrease in crest-trough correlations.

These results are also consistent with the findings of a recent analysis of billions of waves from historical ocean wave data which has shown that the single best predictor of extreme wave heights is actually crest-trough correlation (Häfner, Gemmrich, & Jochum, In Prep). The reduction of crest-trough correlation after spectral evolution on a two-layer fluid seen here seems to be the main reason for the decrease in exceedance probabilities generated from the evolved spectra.

# Chapter 3

# Conclusion

## 3.1 Conclusion

In four of the five cases investigated, the occurrence probability of large waves (measured by multiples of $H_s$) is reduced after evolution on a two-layer fluid. Overall the decrease was larger for initially steeper waves, although the steepest case had a slightly smaller reduction than the second steepest. Additionally, the magnitude of the relative reduction in wave height exceedance probabilities was well correlated with the energy lost relative to the initial spectrum. In the most affected case, no extreme wave heights ($H > 2.2H_s$) were generated, and the relative change in exceedance probabilities of wave heights larger than $H_s = 2$ became less common by a factor of approximately one hundredth.

The decrease in wave height exceedance probabilities was largest in the case of the initial spectrum $[4, 3]$ which did not correspond to the spectrum with the largest initial steepness, $[6, 1]$. Likewise, cases $[3, 2]$ and $[2, 7]$ had slightly different initial steepness (smaller and larger respectively) but the exceedance probabilities exhibited different behaviours in the high and low probability regimes, with their relative change curves crossing each other. More realizations from repeated simulations with different initial seeds in the $[3, 2]$ and $[2, 7]$ cases could provide clarity on the different behaviour exhibited by these exceedance curves in the small and large wave height regimes.

Conversely, the exceedance probabilities of crest heights were found to not change from the initial spectra at lower probabilities. In light of the decrease in wave height exceedance probabilities, this result indicated that correlations between crest heights and trough depths must be reduced. These correlations were calculated, and it was found that correlations between crest height and trough depth did decrease after spectral evolution on a two-layer fluid. The magnitude of the decrease in correlation was also clearly linked with initial steepness. Like with the wave height exceedance probabilities, the magnitude of the changes in crest-trough correlations increased with increasing initial steepness. Additionally, the reductions in crest-trough correlations were similarly correlated with energy lost from the evolved spectrum relative to the initial spectrum.

Future work could explore in more detail both the relationship between initial steepness and relative change in exceedance probabilities and the effect of initial steepness on crest-trough correlations. For this analysis the simulations should be repeated for a larger set of $k_p$ and $A$ values.

# References


Alam, M.-R. (2012). A new triad resonance between co-propagating surface and interfacial waves. *Journal of Fluid Mechanics*, *691*, 267.

Bendat, J. S., & Piersol, A. G. (2011). *Random data: Analysis and measurement procedures* (Vol. 729). John Wiley & Sons.

Casas-Prat, M., & Holthuijsen, L. H. (2010). Short-term statistics of waves observed in deep water. *Journal of Geophysical Research: Oceans*, *115*(C9).

Gemmrich, J., & Garrett, C. (2011). Dynamical and statistical explanations of observed occurrence rates of rogue waves. *Natural Hazards and Earth System Sciences*, *11*(5), 1437–1446.

Gemmrich, J., & Monahan, A. (2021). Surface and interfacial waves in a strongly stratified upper ocean. *Journal of Physical Oceanography*, *51*(2), 269–278.

Häfner, D., Gemmrich, J., & Jochum, M. (In Prep). Real-world rogue wave probabilities.

Holthuijsen, L. H. (2010). *Waves in oceanic and coastal waters.* Cambridge University Press.

Longuet-Higgins, M. S. (1952). On the statistical distribution of the height of sea waves. *Journal of Marine Research*, *11*, 245–266.

Naess, A. (1985). On the distribution of crest to trough wave heights. *Ocean Engineering*, *12*(3), 221–234.

Pierson Jr, W. J., & Moskowitz, L. (1964). A proposed spectral form for fully developed wind seas based on the similarity theory of S.A. Kitaigorodskii. *Journal of Geophysical Research*, *69*(24), 5181–5190.

Thomson, J., Ackley, S., Girard-Ardhuin, F., Ardhuin, F., Babanin, A., Boutin, G., . . . others (2018). Overview of the Arctic sea state and boundary layer physics program. *Journal of Geophysical Research: Oceans*, *123*(12), 8674–8687.

# Appendices

# Appendix A

# MATLAB Code for the Simulation

## A.1 The Simulation

# Script: waveheight_sim

This script calls various functions to simulate wave heights from pre-made wave height spectrum files. This script loops over each file twice, to simulate waves from the initial spectrum and final spectrum from each file.

```
% This is the simulation

% Name all file handles
% These are the files containing the spectra used in the simulation.
handles_opn = {'ExpSpectra_nos_R9700_kp200_A100.mat'...
            'ExpSpectra_nos_R9700_kp200_A100.mat'...
            'ExpSpectra_nos_R9700_kp300_A200.mat'...
            'ExpSpectra_nos_R9700_kp300_A200.mat'...
            'ExpSpectra_nos_R9700_kp400_A300.mat'...
            'ExpSpectra_nos_R9700_kp400_A300.mat'...
            'ExpSpectra_nos_R9700_kp600_A100.mat'...
            'ExpSpectra_nos_R9700_kp600_A100.mat'};

% Save names for files

handles_sve = {'sim10e3_t1_kp2_A1.mat'...
            'sim10e3_t1500_kp2_A1.mat'...
            'sim10e3_t1_kp3_A2.mat'...
            'sim10e3_t1500_kp3_A2.mat'...
            'sim10e3_t1_kp4_A3.mat'...
            'sim10e3_t1500_kp4_A3.mat'...
            'sim10e3_t1_kp6_A1.mat'...
            'sim10e3_t1500_kp6_A1.mat'};
```

Run for each spectrum

```
times = [1,4];
for jj = 1:length(handles_opn)

t1 = clock; % this code would take upwards of 20 hours to run a
            % simulation so it was nice to have some idea when things
            % started and stopped.
time =  datestr(t1,'dd-mmm-yyyy HH:MM:SS');

% open the mat file containing the spectrum
file = open(handles_opn{jj});

% print file currently running
fprintf('Now running from %s\n', sprintf(handles_opn{jj}))
fprintf('Run started at %s.\n', sprintf(time))

Spec_surf = file.Spec_surf; % spectrum at four different times
waveNumber = file.wave_number;

dk = mean(diff(waveNumber)); % these wave numbers are not evenly
                             % spaced. However, they are nearly evenly
                             % spaced so I took the mean of the
```

```
                                       % differences as dk.


dk = dk/30; % this will set the number of coefficients.

kn = min(waveNumber):dk:max(waveNumber); % kn values for making
 coeffs.

% This if statement determines if the spectrum used should be
% the evolved spectrum or the initial spectrum
if rem(jj,2) == 0

    tt = 4;
    spect = interp1(waveNumber,Spec_surf(tt,:),kn); % spectrum

else

    tt = 1;
    spect = interp1(waveNumber,Spec_surf(tt,:),kn);

end

x_start = 0;
x_end = 1000;

num_runs = 10e3; % number of runs

sig_heights = zeros(num_runs,1); % this is where significant wave
                                 % heights are stored.

heights_container = cell(1,num_runs); % this is a cell that contains
                                      % vectors with each wave height.

crests_container = cell(1,num_runs); % cell that contains vectors with
                                     % each crest height

periods_container = cell(1,num_runs); % same but for periods

troughs_container = cell(1,num_runs); % troughs

% start the simulation

rng 'default' % At the beginning of each simulation set the seed to be
              % the same.

for ii = 1:num_runs

    [an,bn] = coefficients(spect,dk); % coefficients

    [x,state] = sea_state(an,bn,kn,x_start,x_end,1); % sea state and
                                                     % distance
                                                     % vector

    % dx for interpolation required for finding zero crossings
```

```
    dx = mean(diff(x))/100;

    % interpolated state and x with zero crossing locations crss
    [x_new,state_new,crss] = find_zero_crss(state,x,dx);

    Hs = 4*std(state_new); % significant wave height

    sig_heights(ii) = Hs;

    % heights and periods (periods are spatial)
    [heights,crests,troughs,periods] = wave_heights2...
                                       (state_new, x_new,crss);

    % stored heights from run
    heights_container{ii} = heights;

    % store crests from run
    crests_container{ii} = crests;

    % store troughs from run
    troughs_container{ii} = troughs;

    % stored periods from run
    periods_container{ii} = periods;

end

% These are the fields to be saved in the data file
H = heights_container;
C = crests_container;
TR = troughs_container;
Hs = sig_heights;
T = periods_container;


t2 = clock;
time =  datestr(t2,'dd-mmm-yyyy HH:MM:SS');
elapsed = etime(t2,t1)/3600;

% save the data
save(handles_sve{jj},'H','Hs','C','T','TR')
fprintf('Finished running %s\n', sprintf(handles_opn{jj}))
fprintf('File Saved as %s\n', sprintf(handles_sve{jj}))
fprintf('Run finished at %s.\n', sprintf(time))
fprintf('The run took %.2f hours.\n', elapsed)
fprintf('----------------------------------------------------------
\n')

end

fprintf('All files run \n')
```

*Published with MATLAB® R2020b*

# Function: Coefficients

This function takes a wave spectrum and either a difference in wavenumber or a difference in frequency. Either will work. The function returns two vectors of coefficients that can be used to determine the surface profile associated with the spectrum.

```
function [an,bn] = coefficients(spectrum,dk)

    spectrum = spectrum.*dk; % due to discrete bandwidth of spectrum

    an = normrnd(0,sqrt(spectrum));
    bn = normrnd(0,sqrt(spectrum));

end
```

*Published with MATLAB® R2020b*

# Function: sea_state

Takes a wavenumber vector, and two vectors of coefficients. Requires spatial or time series start and end values. Returns distance vector and state vector. If using time xort = 0, if using space xort = 1.

```
function [x,state] = sea_state(an,bn,kn,x0,x_end,xort)

% make sure that an, bn, and kn are all column vectors
    [col_a,row_a] = size(an);
    [col_b,row_b] = size(bn);
    [col_k,row_k] = size(kn);

    if col_a == 1

        an = an';

    end

    if col_b == 1

        bn = bn';

    end

    if col_k == 1

        kn = kn';

    end

    k_max = max(kn); % find max k to determine dx

    if xort == 0

        dx = 1/(2*k_max); % calculates dt from frequency

    else

        dx = pi/k_max; % calculates dx from Nyquist wavenumber

    end

    x = x0:dx:x_end;

    waves = an.*cos(kn.*x) + bn.*sin(kn.*x);

    state = sum(waves);

end
```

*Published with MATLAB® R2020b*

# Function: find_zero_crss

find_zero_crss finds the zero crossings in data. It interpolates to a higher resolution set by input dx and returns the zero crossing locations in the new interpolated time series. It also returns the new time series and the new x range for plotting purposes.

```
function [x_interp,state_interp,crss] = find_zero_crss(state, x, dx)

% first interpolate to higher resolution
x_interp = x(1):dx:x(end);

state_interp = pchip(x,state,x_interp);


% Find the zero crossings in the new series
sgn = sign(state_interp);
sgn_shift = [4, sgn(1:end-1)];

dff = sgn + sgn_shift;

% zero crossing point is crss
[~, crss] = find(dff == 0);

end
```

*Published with MATLAB® R2020b*

# Function: wave_heights2

This function takes as vectors the surface profile (state), the x coordinates (x), and the index of the zero crossings of the vector state (crossing_index). It returns the wave heights, crest heights, trough depths, and wave periods.

```
function [heights,crests,troughs,periods] = ...
                                    wave_heights2(state, x,
 crossing_index)

    terminal = length(crossing_index); % last value of crss

    heights = zeros(1,terminal); % vector for filling with wave
                                 % heights
    crests = heights;            % same but for crests
    periods = heights;           % same but for periods
    troughs = heights;           % troughs

    % if the first crossing location has an index of 1, the code will
    % crash. Remove the opertunity for failure.

    if crossing_index(1) == 1

        crossing_index(1) = [];
        state(1) = [];
        x(1) = [];

    end

    if state(crossing_index(1) - 1) > 0 % downward zero crossing

        ii = 1; % the count

        while ii + 2 < terminal

            jj = ii + 2;

            begg = crossing_index(ii); % beginning of interval
            endi = crossing_index(jj); % end of interval

            % fill the heights
            heights(ii) = max(state(begg:endi)) - ...
                          min(state(begg:endi));

            % fill the crests
            crests(ii) = max(state(begg:endi));

            % fill troughs
            troughs(ii) = min(state(begg:endi));

            % fill periods
            periods(ii) = x(endi) - x(begg);
```

```
            % A wave is between two successive zero crossings so
            % to get to the next wave, skip a zero crossing by adding
            % two to the count
            ii = ii + 2;

        end

    else

        ii = 2; % first is upward zero crossing

        while ii + 2 < terminal

            jj = ii + 2;

            begg = crossing_index(ii); % beginning of interval
            endi = crossing_index(jj); % end of interval

            heights(ii-1) = max(state(begg:endi)) - ...
                          min(state(begg:endi));

            crests(ii) = max(state(begg:endi));

            troughs(ii) = min(state(begg:endi));

            periods(ii-1) = x(endi) - x(begg);

            ii = ii + 2;

        end

    end

    % get rid of the zero values

    nope_H = find(heights == 0);
    nope_T = find(periods == 0);
    nope_C = find(crests == 0);
    nope_TR = find(troughs == 0);

    heights(nope_H) = [];
    periods(nope_T) = [];
    crests(nope_C) = [];
    troughs(nope_TR) = [];

end
```

*Published with MATLAB® R2020b*

## A.2 Exceedance Probabilities and Bootstrapping

# Script: bootstrap_all

This script calculates the exceedance probabilities from each set of wave heights. It also performs the bootstrapping calculation to determine uncertainty ranges.

```
% files that need to be opened

file_handles = {'sim10e3_t1_kp2_A1.mat'...
               'sim10e3_t1_kp2_A7.mat'...
               'sim10e3_t1_kp3_A2.mat'...
               'sim10e3_t1_kp4_A3.mat'...
               'sim10e3_t1_kp6_A1.mat'...
               'sim10e3_t1500_kp2_A1.mat'...
               'sim10e3_t1500_kp2_A7.mat'...
               'sim10e3_t1500_kp3_A2.mat'...
               'sim10e3_t1500_kp4_A3.mat'...
               'sim10e3_t1500_kp6_A1.mat'};

% file names that are going to be made

file_sve =  {'sim10e3_t1_kp2_A1_H_boot.mat'...
             'sim10e3_t1_kp2_A7_H_boot.mat'...
             'sim10e3_t1_kp3_A2_H_boot.mat'...
             'sim10e3_t1_kp4_A3_H_boot.mat'...
             'sim10e3_t1_kp6_A1_H_boot.mat'...
             'sim10e3_t1500_kp2_A1_H_boot.mat'...
             'sim10e3_t1500_kp2_A7_H_boot.mat'...
             'sim10e3_t1500_kp3_A2_H_boot.mat'...
             'sim10e3_t1500_kp4_A3_H_boot.mat'...
             'sim10e3_t1500_kp6_A1_H_boot.mat'};


% alpha = linspace(0.5,2.4,50)./1.76; %% use this alpha if using
```

# crest heights

```
alpha = linspace(0.5,2.4,50);

for file = 1:length(file_handles)

    % This can take a long time. It's nice to know when a file started
    t1 = clock;
    time =  datestr(t1,'dd-mmm-yyyy HH:MM:SS');

    fprintf('Running from %s.\n', sprintf(file_handles{file}))
    fprintf('Run started at %s.\n', sprintf(time))

    % open the data
    data = open(file_handles{file});


    HH = data.H; % wave heights
```

```
    Hs = data.Hs; % significant wave height

    % calculate the exceedance probabilities
    exceeds = exceedance(HH,Hs,alpha);

    % bootstrap the data
    [err_neg,err_pos,probs] = bootstrapper(HH,Hs,alpha);

    % these are the fields to be saved in the new file
    neg = err_neg;
    pos = err_pos;
    probMat = probs;
    prob = exceeds;

    % save the file
    save(file_sve{file},'neg','pos','probMat','prob')

    t2 = clock;
    time =  datestr(t2,'dd-mmm-yyyy HH:MM:SS');
    elapsed = etime(t2,t1)/3600;

    fprintf('Finished running %s.\n', sprintf(file_handles{file}))
    fprintf('File Saved as %s.\n', sprintf(file_sve{file}))
    fprintf('Run finished at %s.\n', sprintf(time))
    fprintf('The run took %.2f hours.\n', elapsed)

 fprintf('-----------------------------------------------------------
\n')


end

fprintf('All files run.\n')
fprintf('-----------------------------------------------------------
\n')
fprintf('-----------------------------------------------------------
\n')
```

*Published with MATLAB® R2020b*

# Function: exceedance

This function takes, as parameters, a cell containing vectors of waveheights, Hs is a vector containing the significant waveheight from each run, and alpha is a vector containing exceedence ratios of H/Hs.

```
function exceeded = exceedance(heights,Hs,alpha)

    num_waves = length([heights{:}]); % total number of waves

    runs = length(Hs); % number of runs as indicated by length of
                       % Hs vector

    ratios = cell(1,runs); % make an empty cell to  be filled
                           % with ratios

    for run = 1:runs
        % calculate ratios of H/Hs
        ratios{1,run} = heights{1,run}./Hs(run);

    end

    ratios = [ratios{:}];

    exceeded = zeros(length(alpha),1);

    for ii = 1:length(alpha)

    exceed_index = find(ratios > alpha(ii));

    % exceedence probability
    exceeded(ii) = length(exceed_index)/num_waves;


    end

end
```

*Published with MATLAB® R2020b*

# Function: bootstrapper

This function takes data as a cell (H), and as a vector (Hs), and a vector of exceedance values (alpha). The data, H, is bootstrapped. The function returns the 75th and 25th percentiles of the data based on the values of the surrogate data.

```
function [err_neg, err_pos, probs] = bootstrapper(H,Hs,alpha)

    run_num = length(Hs); % number of different runs

    H = [H{:}]; % vector of all wave heights
    Hs = mean(Hs); % mean of Hs

    samp_size = length(H); % determines how many samples to take

    alpha_size = length(alpha); % determines the number of columns of
 the
                                % probability matrix

    samp_frac = round(0.75*samp_size); % number of samples to be taken
                                       % from data

    probs = zeros(run_num,alpha_size); % initialize probability matrix

    % fill the probability matrix
    for jj = 1:run_num

        H_jj = datasample(H,samp_frac); % datasample

        exceeded = exceedance_boot(H_jj,Hs,alpha); % exeedance probs

        probs(jj,:) = exceeded; % each row of probs is filled with the
                                % exceedance probabilities calculated
 in
                                % the line above.

    end

    % These are returned. They are the top and bottom of the
    % error bars.
    % The true value should be somewhere between the 25th and 75th
    % percentile.
    err_pos = prctile(probs,75,1);
    err_neg = prctile(probs,25,1);
```

*Published with MATLAB® R2020b*

# Function: exceedance_boot

This function is identical to exceedance, but it takes H as a vector instead of a cell and Hs as a single value.

```
function exceeded = exceedance_boot(heights,Hs,alpha)

    num_waves = length(heights); % total number of waves

    ratios = heights./Hs;

    exceeded = zeros(1,length(alpha));

    for ii = 1:length(alpha)

    exceed_index = find(ratios > alpha(ii));

    %exceedence probability
    exceeded(ii) = length(exceed_index)/num_waves;

    end


end
```

*Published with MATLAB® R2020b*