OCEANLYZ

Release 1.5

Aug 05, 2020

Contents

OCEANLYZ, Ocean Wave Analyzing Toolbox

MATLAB / GNU Octave Toolbox

Developer Arash Karimpour

Website http://www.arashkarimpour.com

Download http://www.arashkarimpour.com/oceanlyz.html

Documentation https://oceanlyz.readthedocs.io

Tutorial Videos https://www.youtube.com/playlist?list=PLcrFHi9M_GZRTCshcgujlK7y5ZPim6afM

Source code https://github.com/akarimp/oceanlyz

Report issues https://github.com/akarimp/oceanlyz/issues

Version 1.5

OCEANLYZ, Ocean Wave Analyzing Toolbox, is a toolbox for analyzing the wave time series data collected by sensors in open body of water such as ocean, sea, and lake or in a laboratory.

CHAPTER 1

Introduction

OCEANLYZ, Ocean Wave Analyzing Toolbox, is a toolbox for analyzing the wave time series data collected by sensors in open body of water such as ocean, sea, and lake or in a laboratory.

This toolbox contains functions that each one is suitable for a particular purpose. Both spectral and zero-crossing methods are offered for wave analysis. This toolbox can calculate wave properties such as zero-moment wave height, significant wave height, mean wave height, peak wave period and mean period. This toolbox can correct and account for the pressure attention (pressure loss) in the water column for data collected by a pressure sensor. This toolbox can separate wind sea and swell energies and reports their properties.

1.1 Download

To download OCEANLYZ, visit http://www.arashkarimpour.com

1.2 Installation

To use this toolbox, download and unzip it in any location you choose such as "C:\".

1.3 Operating System

This code can be run on Windows, Mac and Linux. However, make sure any given path is compatible with a running operating system. In particular, "\" is used in Windows path, while "/" is used in Mac or Linux path. For example, if a path is "C:\" on Windows machine, it would be "C:/" on Mac or Linux.

1.4 Required Programing Language

This toolbox can be run by using MATLAB (https://www.mathworks.com) or GNU Octave (https://www.gnu.org/software/octave).

1.5 Required Package for MATLAB

MATLAB users need to install MATLAB Signal Processing Toolbox for running the OCEANLYZ spectral analysis. It gives OCEANLYZ access to MATLAB Welch's power spectral density calculation. However, MATLAB Signal Processing Toolbox it is not required for zero-crossing analysis.

1.6 Required Package for GNU Octave

GNU Octave users need to install/load GNU Octave Signal Package for running the OCEANLYZ spectral analysis. It gives OCEANLYZ access to GNU Octave Welch's power spectral density calculation. However, GNU Octave Signal Package it is not required for zero-crossing analysis.

GNU Octave Signal Package can be loaded inside GNU Octave by using a following command in a command window (This should be done every time GNU Octave is opened):

```
>> pkg load signal
```

If GNU Octave Signal Package is not already installed, it should be first installed from Octave Forge (octave.sourceforge.io), and then get loaded by using the following commands in a command window:

```
>> pkg install -forge signal
>> pkg load signal
```

1.7 Quick Start

- Open MATLAB or GNU Octave
- Change a current folder (current directory) to a folder that contains OCEANLYZ toolbox, for example "C:\oceanlyz", in MATLAB or GNU Octave.
- Open a file named "oceanlyzinput.m" in MATLAB or GNU Octave editor and modify it based on the properties of the collected dataset and required analysis.
- Run a file named "RunOceanlyz.m" in MATLAB or GNU Octave to start calculations.

For more information, refer to "Tutorial" section.

1.8 Recommended Book

For more details on coastal and ocean wave data analysis refer to:

Karimpour A., (2018), Ocean Wave Data Analysis: Introduction to Time Series Analysis, Signal Processing, and Wave Prediction, KDP.

Book link: https://www.amazon.com/dp/0692109978

1.9 Citation

Cite this toolbox as:

Karimpour, A., & Chen, Q. (2017). Wind Wave Analysis in Depth Limited Water Using OCEANLYZ, a MATLAB toolbox. Computers & Geosciences.

1.10 License Agreement and Disclaimer

CHAPTER 2

Getting Started

In order to use OCEANLYZ toolbox, first, one of the MATLAB or GNU Octave programming language should be installed with required packages (Refer to required packages for MATLAB/GNU Octave).

Then, the OCEANLYZ toolbox should be downloaded and unzip in a folder of your choice. For example, if the OCEANLYZ is unzipped on C: derive, then OCEANLYZ folder would be “C:\oceanlyz” on a windows machine.

Before using the OCEANLYZ, these steps should be followed first:

1. Setting values representing the properties of the dataset needs to be analyzed
2. Defining what type of analysis is required to be done
3. Preparing dataset in a format that the OCEANLYZ can use

These steps are explained in detail in following sections.

2.1 Set up Input, Output, and Calculation parameters

Values that define input, output, and calculation parameters are defined in a file named “oceanlyzinput.m”. To set up these parameters, first, open file “oceanlyzinput.m” by following these steps:

- Open MATLAB or GNU Octave
- Change a current folder (current directory) to a folder that contains OCEANLYZ toolbox in MATLAB or GNU Octave. For example, open “C:\oceanlyz” directory.
- Open a file named “oceanlyzinput.m” in MATLAB or GNU Octave editor and modify it based on the properties of the collected dataset and required analysis.

Parameters that can be defined in “oceanlyzinput.m” are described in following.

2.2 Input, Output, and Calculation Parameters

InputFileName It defines a name of the input data file

Notes: Enter name of the file contains water depth or surface elevation (Eta) time series.

Data should be in a single column vector file without any text, each burst of data should follow the previous burst.

Most file formats such as ‘.mat’, ‘.txt’, ‘.csv’ can be imported.

Example: InputFileName=’waterdepthsample.mat’;

InputFileFolder It defines a location of the input data file

Notes: Enter a location of the file contains water depth or surface elevation (Eta) time series.

For Linux or Mac, use ‘/’ instead of ‘’

Example: InputFileFolder=’C:\oceanlyz\Sample’;

SaveOutput It defines if to save output(s) and a location the output(s) to be saved

SaveOutput=’off’; :Does not save output(s)

SaveOutput=’on’; :Save output(s)

OutputFileFolder Location of the output file to be saved

OutputFileFolder=’C:\oceanlyz’;

Notes: For Linux or Mac, use ‘/’ instead of ‘’.

Output is a structure array

Name of output file is ‘wave.mat’

After wave.mat is loaded in Matlab, field(s) in structure array can be called by using ‘.’

Example: Output peak wave period is : “wave.Tp”

AnalysisMethod It defines analysis method

AnalysisMethod=’spectral’; :Use spectral analysis method / Fast Fourier Transform

AnalysisMethod=’zerocross’; :Use zero-crossing method

WaveParameterCalc It defines if to calculate wave parameters or not

WaveParameterCalc=’off’; :Does not calculate/report wave properties

WaveParameterCalc=’on’; :Calculates/reports wave properties

Notes: Use WaveParameterCalc=’off’ to only report corrected water level data measured by a pressure sensor.

If WaveParameterCalc=’off’, then pressureattenuation should be pressureattenuation=’on’ or ‘all’.

If WaveParameterCalc=’off’, it reports corrected water level data measured by a pressure sensor.

by accounting for pressure attenuation (pressure loss) in depth.

If WaveParameterCalc=’on’ and pressureattenuation=’on’ or ‘all’, it reports both waves and corrected water level data measured by a pressure sensor.

n_burst it defines number of burst(s) in the input file

n_burst=5; :Number of burst(s) in the input file

Notes: n_burst=total number of data points/(burst_duration*fs).

Example: For 12 bursts of data, which each burst has a duration of 30 minutes, and collected at sampling frequency of 10 Hz:

burst_duration=(30*60)

total number of data points = number of burst * burst_duration of each burst * sampling frequency = 12*(30*60)*10

burst_duration it defines duration time that data collected in each burst in (second)

burst_duration=1024; :Duration time that data collected in each burst in (second)

fs It defines sampling frequency that data are collected at in (Hz)

fs=10; :Sampling frequency that data are collected at in (Hz)

heightfrombed It defines pressure sensor height from a bed in (m)

heightfrombed=0.05; :Pressure sensor height from a bed in (m)

Notes: Leave heightfrombed=0.0; if data is not measured by a pressure sensor or if a sensor sits on the seabed.

nfft It defines NFFT for Fast Fourier Transform

nfft=2^10; :NFFT for Fast Fourier Transform

Notes: Results will be reported for frequency range of 0 <= f <= (fs/2) with (nfft/2+1) data points.

Example: If fs=4 Hz and nfft=1024, then output frequency has a range of 0 <= f <= 2 with 513 data points.

seaswellCalc it defines if to separate wind sea and swell waves or not

seaswellCalc='off'; :Does not separate wind sea and swell waves

seaswellCalc='on'; :Separates wind sea and swell waves

fminswell It defines minimum frequency that swell can have

fminswell=0.1; :Minimum frequency that swell can have (it is used for Tpswell calculation) in (Hz)

fmaxswell It defines maximum frequency that swell can have

fmaxswell=0.25; :Maximum frequency that swell can have (It is about 0.2 in Gulf of Mexico) in (Hz)

pressureattenuation It defines if to apply pressure attenuation factor or not

pressureattenuation='off'; :No pressure attenuation is applied

pressureattenuation='on'; :Pressure attenuation is applied without correction after fmaxpcorr

pressureattenuation='all'; :Pressure attenuation is applied with constant correction after fmaxpcorr

Notes: Pressure attenuation factor is used to account for pressure attenuation (pressure loss) in depth.

For pressureattenuation='on' or 'all', input data should be water depth.

autofmaxpcorr It defines if to calculate fmaxpcorr and ftailcorrection based on water depth or not

autofmaxpcorr='off': :Does not calculate fmaxpcorr and ftailcorrection based on water depth

autofmaxpcorr='on': :Calculate fmaxpcorr and ftailcorrection based on water depth

Notes: Code calculate a maximum frequency that a pressure attenuation factor should be applied up to that.

fminpcorr It defines minimum frequency that automated calculated fmaxpcorr can have if autofmaxpcorr='on' in (Hz)

fminpcorr=0.15; :Minimum frequency that automated calculated fmaxpcorr can have if autofmaxpcorr='on' in (Hz)

Notes: If autofmaxpcorr='on', then fmaxpcorr will be checked to be larger or equal to fminpcorr.

fmaxpcorr It defines maximum frequency for applying pressure attenuation factor in (Hz)

fmaxpcorr=0.55; :Maximum frequency for applying pressure attenuation factor in (Hz)

Notes: Pressure attenuation factor is not applied on frequency larger than fmaxpcorr.

mincutoff It defines if to cut off the spectrum below fmin, i.e. where f<fmin, or not

mincutoff='off'; : Does not cut off spectrum below fmin

mincutoff='on'; : Cuts off spectrum below fmin

fmin It defines minimum frequency to cut off the lower part of spectrum in (Hz)

fmin=0.04; :Minimum frequency to cut off the lower part of spectrum in (Hz)

Notes: If mincutoff='on', then results with frequency f<fmin will be removed from analysis.

It is a simple high pass filter.

maxcutoff It defines if to cut off the spectrum beyond fmax, i.e. where f>fmax, or not

maxcutoff='off'; : Does not cut off spectrum beyond fmax

maxcutoff='on'; : Cut off spectrum beyond fmax

fmax It defines maximum frequency to cut off the upper part of spectrum in (Hz)

fmax=1; :Maximum frequency to cut off the upper part of spectrum in (Hz)

Notes: If maxcutoff='on', then results with frequency f>fmax will be removed from analysis.

It is a simple low pass filter.

tailcorrection It defines if to apply diagnostic tail correction or not

tailcorrection='off'; :Does not apply diagnostic tail

tailcorrection='jonswap'; :Applies JONSWAP Spectrum tail

tailcorrection='tma'; :Applies TMA Spectrum tail

Notes: For tailcorrection='tma', input data should be water depth.

ftailcorrection It defines frequency that diagnostic tail applies after that in (Hz)

ftailcorrection=0.9; :Frequency that diagnostic tail applies after that in (Hz)

Notes: ftailcorrection is typically set at 2.5fm where fm=1/Tm01.

tailpower It defines power that a diagnostic tail will be applied based on that

tailpower=-5; :Power that a diagnostic tail will be applied based on that

Notes: Diagnostic tail will be proportional with (f^tailpower).

tailpower=-3 for shallow water, tailpower=-5 for deep water.

dispout It defines if to plot spectrum or not

dispout='off'; :Does not plot

dispout='on'; :Plot

2.3 Required Parameters for Spectral Analysis

All parameters mentioned in a previous section might be required for the spectral analysis (depending on which module is on or off). In other words, if AnalysisMethod='spectral'; then all mentioned parameters above might be required. If a parameter is not required, it is ignored by OCEANLYZ if defined.

2.4 Required Parameters for Zero-Crossing Method

Not all parameters mentioned in previous section are required for the zero-crossing method. If AnalysisMethod= 'zerocross'; then only following parameters are required. All other parameters, if defined, are ignored by OCEANLYZ.

```
InputFileName='waterdepthsample.mat';
InputFileFolder=pwd;
SaveOutput='on';
OutputFileFolder=pwd;
AnalysisMethod='spectral';
WaveParameterCalc='on';
n_burst=1;
burst_duration=1024;
fs=1;
heightfrombed=0.0;
pressureattenuation='off';
dispout='off';
```

2.5 Default Values for Input, Output, and Calculation Parameters

Default values are set as follow:

```
InputFileName='waterdepthsample.mat';
InputFileFolder=pwd;
SaveOutput='on';
OutputFileFolder=pwd;
AnalysisMethod='spectral';
WaveParameterCalc='on';
n_burst=1;
burst_duration=1024;
fs=1;
heightfrombed=0.0;
nfft=2^10;
seaswellCalc='off';
fminswell=0.1;
fmaxswell=0.25;
pressureattenuation='off';
autofmaxpcorr='off';
fminpcorr=0.15;
fmaxpcorr=0.55;
mincutoff='off';
fmin=0.05;
maxcutoff='off';
fmax=fs/2;
tailcorrection='off';
ftailcorrection=0.9;
tailpower=-5;
dispout='off';
```

2.6 Run OCEANLYZ

To run OCEANLYZ follow these steps:

- Open MATLAB or GNU Octave.
- Change a current folder (current directory) to a folder that contains OCEANLYZ toolbox inside MATLAB or GNU Octave. For example, open “C:\oceanlyz” directory.
- Run a file named “RunOceanlyz.m” in MATLAB or GNU Octave to start calculations.

2.7 Outputs

Output(s) of the wave properties are reported based on the selected parameters as a structure array named “wave”. Field(s) in the structure array “wave” can be called by using “.”. For example, an output a peak period is “wave.Tp”, an output for zero-moment wave height is “wave.Hm0”, and an output for a water surface elevation power spectral density is “wave.Syy”.

In general, output(s) for each time step (each burst) is reported in one row. For example, if an input file contains 5 bursts of data, then outputs has 5 rows, each row contains output for one burst. For this example, wave.Tp(1,1) or wave.Syy(1,:) are outputs for the first burst. Similarly, wave.Tp(5,1) or wave.Syy(5,:) are outputs for the fifth burst.

If SaveOutput=’on’, then the output(s) is saved in a file named “wave.mat” as a structure array in a folder defined by OutputFileFolder.

2.8 Notes

Note1: If data are collected in continuous mode you can choose n_burst and burst_duration as follow:

The burst_duration is equal to a period of time that you want data averaged over that. For example, if you need wave properties reported every 15 min, then the burst_duration would be 15*60 second

The n_burst is equal to the total length of the time series divided by the burst_duration. The n_burst should be a rounded number. So, if the total length of the time series divided by the burst_duration leads to a decimal number, then data should be shortened to avoid that.

Note2: In a calculation, NFFT value that is set in “oceanlyzinput.m” file will be used. However, a user can set NFFT to be calculated automatically. This should be done inside each function. In that case, NFFT will be set equal to the smallest power of two that is larger than or equal to the absolute value of the total number of data points in each burst. This should be done manually inside each function.

Note3: Welch spectrum is used to calculate a power spectral density. In all spectral calculation, a default window function with a default overlap window between segments is used. If any other values are required, it should be changed manually inside each function.

Note4: If autofmaxpcorr=’on’, then the package calculates fmaxpcorr and ftailcorrection based on water depth and a sensor height from seabed (refer to Applying Pressure Response Factor section). A maximum value for calculated fmaxpcorr and ftailcorrection will be limited to the ones user set in “oceanlyzinput.m” file.

CHAPTER 3

Prepare Input Data File

This section explains about a format of input data file that is being used in a calculation. An input data file contains a time series of water surface elevation or water depth. An input data file should be prepared as a single file, with only one column. An input data file should not contain any text. Most of file formats such as ".mat", ".txt", ".xlsx", and ".csv" can be imported by the program. The schematics of input data file for data recorded in a burst mode and a continuous mode are demonstrated in following sections.

3.1 Data Recorded in Burst Mode

If data are recorded in a burst mode, then an instrument records data for a specific duration, such as 20 minutes, and then it goes to sleep to save the battery for a predefined duration, such as 60 minutes, before it wakes up again and repeats the recording/sleeping procedure again. Assuming there are M bursts of data, and each burst contains N data points, then a schematic of input data file is:

Burst 1, data point 1	
Burst 1, data point 2	
Burst 1, data point 3	
.	
.	
.	
Burst 1, data point N-2	
Burst 1, data point N-1	
Burst 1, data point N	
Burst 2, data point 1	
Burst 2, data point 2	
Burst 2, data point 3	
.	
.	
.	

Continued on next page

Table 1 – continued from previous page

Burst 2, data point N-2	
Burst 2, data point N-1	
Burst 2, data point N	
.	
.	
.	
Burst M, data point 1	
Burst M, data point 2	
Burst M, data point 3	
.	
.	
.	
Burst M, data point N-2	
Burst M, data point N-1	
Burst M, data point N	

If data recorded with a sampling frequency of fs, then an input data file has a size of:

(fs * duration of each burst in seconds * total number of recorded bursts , 1)

In this mode, the total number of bursts is M. The total number of samples in each burst is:

N= fs * duration of each burst in seconds

3.2 Data Recorded in Continuous Mode:

If data are recorded in a burst mode, and the total number of recorded data points is equal to N, then a schematic of input data file is:

data point 1	
data point 2	
data point 3	
.	
.	
.	
data point N-2	
data point N-1	
data point N	

If data recorded with a sampling frequency of fs for a total duration equal to hr hours, then an input data file has a size of:

(fs * hr * 3600,1)

To define a number of bursts when data are collected in a continuous mode, a length (duration) of each burst should be defined. The total number of bursts is equal to the total duration that data are collected divided by a duration one burst. For example, consider data that were recorded continuously for 24 hours. Now, if 30 minutes of data is selected to represent one burst, then the total number of bursts is equal to the 24 hours divided by 0.5 hour (30 minutes), i.e. (24 / 0.5)=48. In this example, the total number of samples in each burst is (fs * 30 * 60).

CHAPTER 4

Sample Files

With this package, two sample files are provided. The first file is named “input.mat”, and contains one burst of data recorded by a pressure sensor. It can be used as an input for any of function. The second file is named “waterdepthsample.mat”, and contains five bursts of data recorded by a pressure sensor. It can be used as an input file while using “RunOceanlyz.m” file.

Measurement properties for a file named “input.mat” is:

Properties	Value
File name	input.mat
Number of recorded burst (burst)	1
Sampling frequency (fs)	10 (Hz)
Recording duration (duration). This is duration of one burst.	1024 (second)
Pressure sensor height from bed (heightfrombed)	0.05 (m)
Mean water depth (h)	1.07 (m)

Measurement properties for a file named “waterdepthsample.mat” is:

Properties	Value
File name	waterdepthsample.mat
Number of recorded burst (burst)	5
Sampling frequency (fs)	10 (Hz)
Recording duration (duration). This is duration of one burst.	1024 (second)
Pressure sensor height from bed (heightfrombed)	0.05 (m)

CHAPTER 5

Apply Pressure Response Factor

Dynamic pressure from water wave will attenuate by moving from a water surface toward a seabed. Because of that, data collected by a pressure sensor have lower magnitudes compared to actual pressure values. To fix this issue and account for the pressure loss in the water column, pressure data collected by a pressure sensor should be divided by a pressure response factor. To do that, first, pressure data should be converted to water depth as:

$$h_{sensor} = \frac{P}{\rho g} + d_s$$

Then, data are corrected as:

$$h = \frac{h_sensor}{K_P}$$

$$K_P = \frac{\cosh(kd_s)}{\cosh(kh_m)}$$

where, h_sensor is water depth read by a pressure sensor, P is pressure, K_P is a pressure response factor, k is a wavenumber, h_m is a mean water depth, and d_s is a distance of the sensor location from a seabed.

One method to use these equations to correct the data is to use them on every single wave. To do that, a zero-crossing method is used to calculate K_P for every single wave. Then, K_P can be applied to each wave height or on water level time series corresponded to each wave.

Another method to use these equations to correct the data is to use Fast Fourier transform and convert a time series to the frequency domain, and then applying K_P on each frequency. In that case, calculated K_P in the frequency domain decreases from 1 to 0 as frequency, f, increases (Figure 1). It means that, a value of $1/K_P$ will be 1 for f=0, and increases toward infinity as a frequency increases. Because of that, if K_P applies to an entire frequency domain, it results in will have very large values in the high frequency region. To avoid that, lower and upper limits for frequencies should be considered, where K_P is not applied to raw data outside of that range.

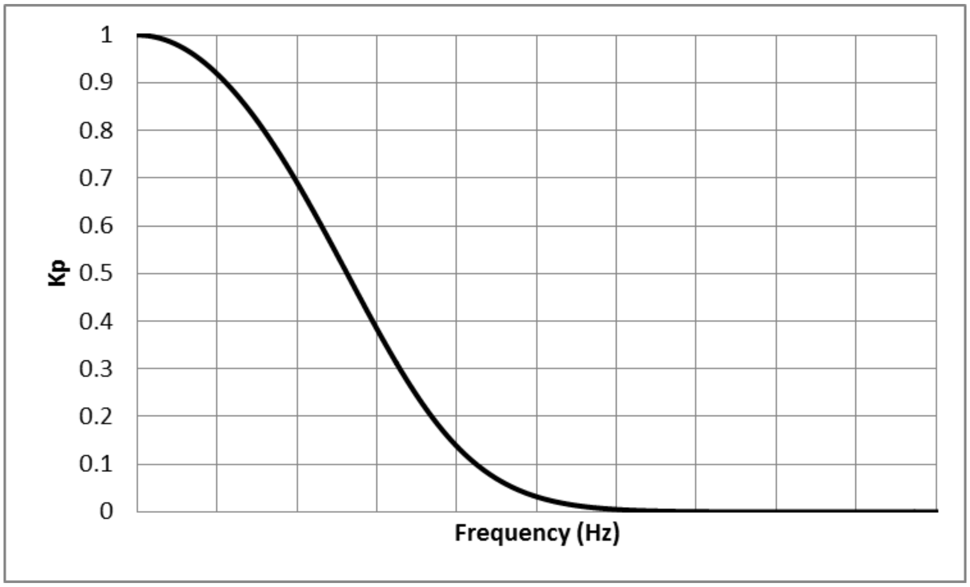

Fig. 1: Figure 1: Schematic trend of K_P versus f

A lower-frequency limit for applying K_P is the smallest frequency that is available in a time series such as f=0.04 to f=0.05 Hz. But an upper-frequency limit is depending on the deployment situations. As it was pointed out, wave pressure attenuates from a water surface toward a seabed (Figure 2). As a wave gets smaller (its frequency increases) it damps in water depth sooner. Because of that, there is a limit that a wave smaller than that (wave with larger frequency than that limit) cannot be sensed by a sensor, as their effect is damped completely before reaching a sensor depth. In that case, K_P should only apply to a range of frequencies that their effects reach a sensor location. So, for any case, depending on deployment situations, a range of frequencies that are large enough to reach a sensor should be calculated, and K_P only applies to that range.

To calculate a maximum frequency that K_P should be applied, deep-water condition can be considered as $h = \frac{L}{2}$ or $kh = \pi$. Considering a sensor setup as described in Figure 2, it can be written:

$$h - d_s = \frac{L}{2} \, or \, k(h - d_s) = \pi \, so \, k = \frac{\pi}{(h - d_s)}$$

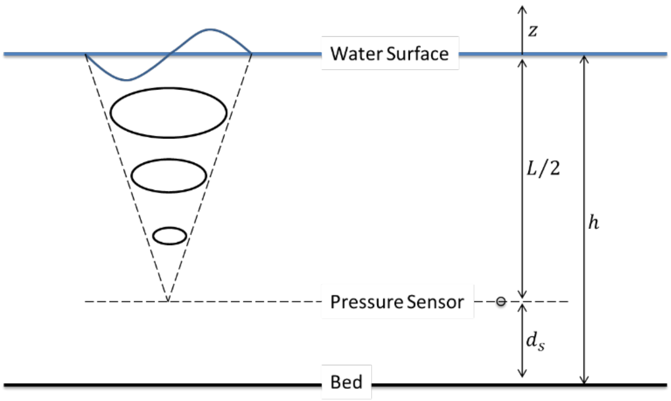

Fig. 2: Figure 2: Schematic of sensor deployment setup

$$K_{P-min}(h,d_s) = \frac{\cosh(k(h+z))}{\cosh(kh)} = \frac{\cosh(kd_s)}{\cosh(kh)} = \frac{\cosh(\frac{d_s}{(h-d_s)}\pi)}{\cosh(\frac{h}{(h-d_s)}\pi)}$$

Then, a maximum frequency that K_P should apply to data up to that frequency is:

$$\omega^2 = gk\tanh(kh)$$

$$(2\pi f)^2 = gk\tanh(kh)$$

$$f = \frac{\sqrt{gk\tanh(kh)}}{2\pi}$$

$$f_{max(for\ pressure\ correction)} = \frac{\sqrt{g\frac{\pi}{(h-d_s)}\tanh(\frac{h}{(h-d_s)}\pi)}}{2\pi}$$

For the case that a pressure sensor sits on a seabed (i.e. $d_s = 0$ and $kh = \pi$), the K_{P-min} and $f_{max(for\ pressure\ correction)}$ are:

$$K_{P-min}(h,0) = \frac{\cosh(0)}{\cosh(\pi)} = 0.0863$$

$$f_{max(for\ pressure\ correction)} = \frac{\sqrt{g\frac{\pi}{h}\tanh(\pi)}}{2\pi}$$

Figures 3 and 4 show the maximum frequency that K_P should not apply beyond that frequency.

For more details on this topic refer to Karimpour and Chen (2017) and Karimpour (2018).

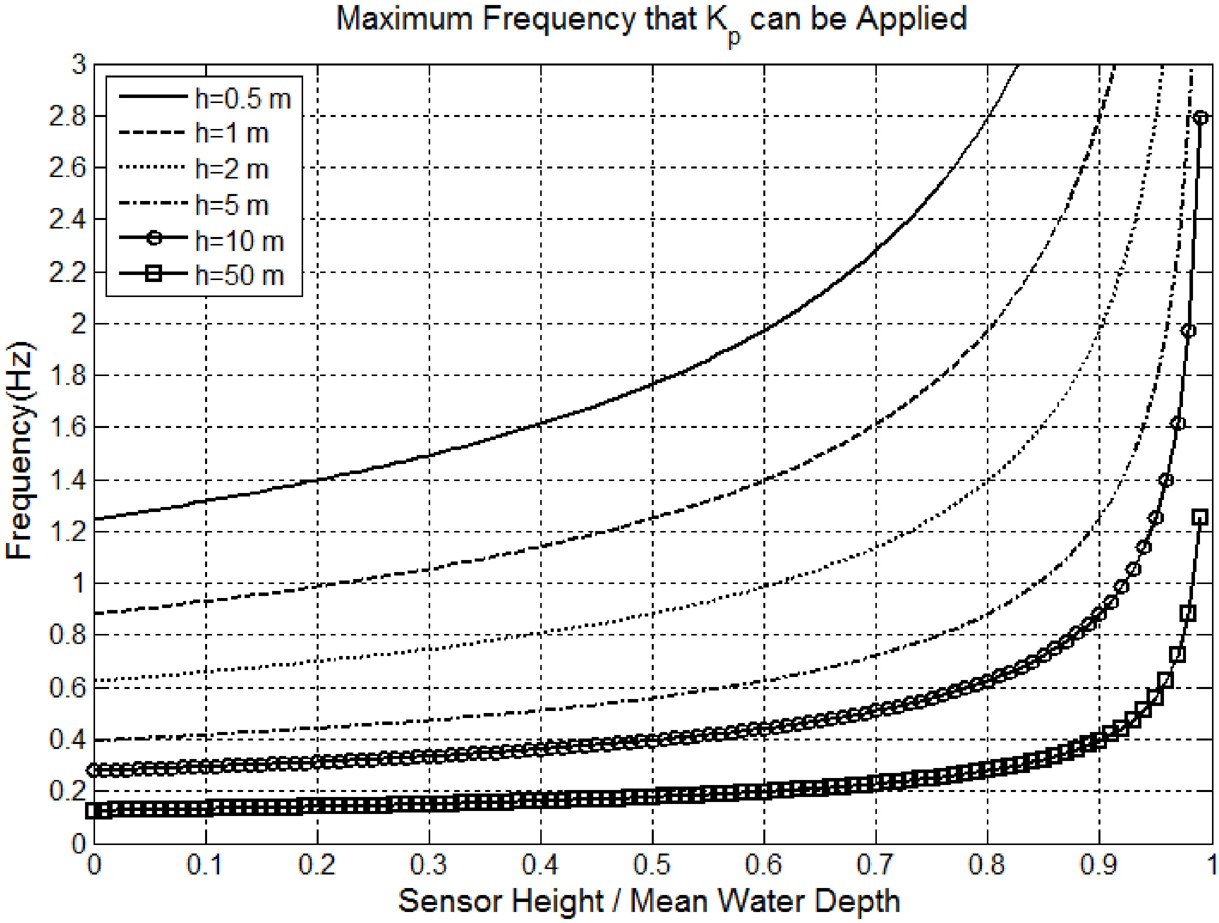

Fig. 3: Figure 3: Maximum frequency that K_P should not applied beyond that frequency.

5.1 References

- Karimpour, A., & Chen, Q. (2017). Wind Wave Analysis in Depth Limited Water Using OCEANLYZ, a MATLAB toolbox. Computers & Geosciences, 106,181-189.
- Karimpour A., (2018), Ocean Wave Data Analysis: Introduction to Time Series Analysis, Signal Processing, and Wave Prediction, KDP.

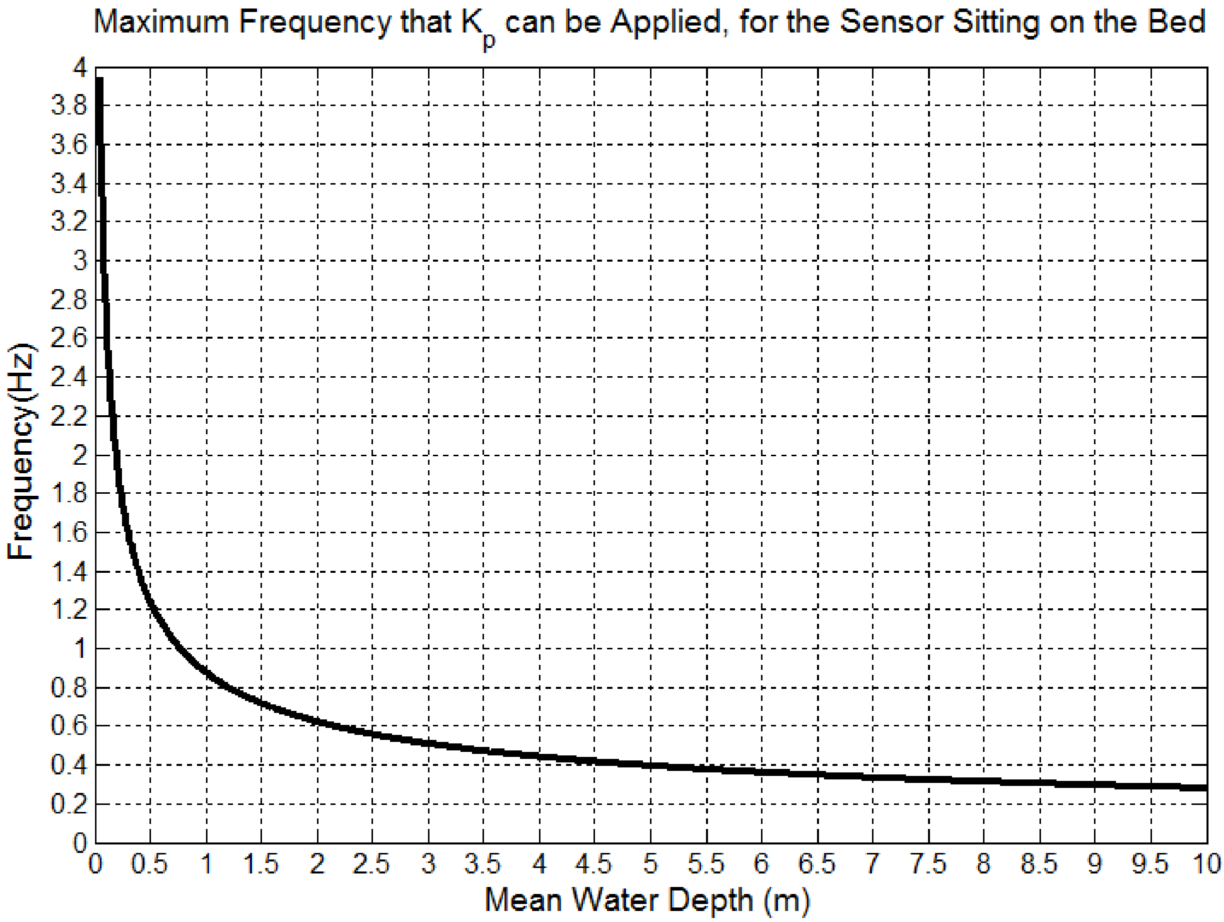

Fig. 4: Figure 4: Maximum frequency that K_P should not apply beyond that frequency, for a sensor sitting on a seabed.

CHAPTER 6

Apply Diagnostic Tail

A diagnostic tail is used to correct a high frequency section of the wave spectrum. It is not recommended to use a diagnostic tail for measured data, unless data are recorded with a low sampling frequency which leads to no data in higher frequency, or in case that the noise in higher frequency contaminated the data. In these cases, a higher section of the spectrum can be replaced by a diagnostic tail as (Siadatmousavi et al. 2012):

$S_{yy}(f) = S_{yy}(f_{tail}) \times (\frac{f}{f_{tail}})^{(-n)}$ for $f > f_{tail}$

where, $S_{yy}(f)$ is a water surface elevation power spectral density, f is a frequency, f_{tail} is a frequency that a tail applied after that, and n is a power coefficient. The f_tail typically set at 2.5f_m, where $f_m = 1/T_{m01}$ is a mean frequency (Ardhuin et al. 2010). A value of n depends on deployment conditions, however, typically it is -5 for deep and -3 for shallow water (e.g. Kaihatu et al. 2007, Siadatmousavi et al. 2012).

For more details on this topic refer to Karimpour and Chen (2017) and Karimpour (2018).

6.1 References

- Ardhuin, F., Rogers, E., Babanin, A. V., Filipot, J. F., Magne, R., Roland, A., … & Collard, F. (2010). Semiempirical dissipation source functions for ocean waves. Part I: Definition, calibration, and validation. Journal of Physical Oceanography, 40(9), 1917-1941.
- Kaihatu, J. M., Veeramony, J., Edwards, K. L. & Kirby, J. T. 2007 Asymptotic behaviour of frequency and wave number spectra of nearshore shoaling and breaking waves. J. Geophys. Res. 112, C06016

- Karimpour, A., & Chen, Q. (2017). Wind Wave Analysis in Depth Limited Water Using OCEANLYZ, a MATLAB toolbox. Computers & Geosciences, 106,181-189.
- Karimpour A., (2018), Ocean Wave Data Analysis: Introduction to Time Series Analysis, Signal Processing, and Wave Prediction, KDP.
- Siadatmousavi, S. M., Jose, F., & Stone, G. W. (2011). On the importance of high frequency tail in third generation wave models. Coastal Engineering.

OCEANLYZ functions can be used as a standalone command in Matlab/GNU Octave command line or it can be embedded in Matalb/GNU Octave script file (.m file). Here are main OCEANLYZ functions:

CHAPTER 7

Functions

This code contains 5 main functions to analyze wave data (water level data) time series. Water level could be in a form of the total depth or it can be in the form of surface elevation. If a pressure sensor is used for data collection, then it is required to correct the data for pressure attenuation in the water column. In such cases, only water depth data should be used as an input. In any other cases, either of water surface elevation or water depth time series can be used for data analysis.

The provided "RunOceanlyz.m" file analyzes the the entire time series by calling an appropriate function(s). Note that any of these functions may be called by a user in his/her own code as well.

The main functions that are used in this toolbox are:

Function	Description
WaveSpectraFun	Calculates wave properties from spectral analysis
WaveZerocrossingFun	Calculates wave properties from zero-crossing
PcorFFTFun	Corrects pressure attenuation effect using Fast Fourier transform (FFT)
PcorZerocrossingFun	Corrects pressure attenuation effect using zero-crossing
SeaSwellFun	Separates wind sea and swell

CHAPTER 8

WaveSpectraFun

```
[Hm0,Tm01,Tm02,Tp,fp,f,Syy]=WaveSpectraFun(input,fs,duration,nfft,h,heightfrombed,
↪fmin,fmax,ftailcorrection,tailpower,mincutoff,maxcutoff,tailcorrection,dispout)
```

8.1 DESCRIPTION

Calculate wave properties from power spectral density

8.2 INPUT

input=importdata(‘h.mat’); Load water depth (h)/surface elevation (Eta) data and rename it “input” in (m)

fs=10; Sampling frequency that data collected at in (Hz)

duration=1024; Duration time that data collected in input in each burst in second

nfft=2^10; NFFT for Fast Fourier Transform

h=1; Mean water depth in (m)

heightfrombed=0.0; Sensor height from bed

fmin=0.04; Minimum frequency for cut off the lower part of spectra

fmax=1; Maximum frequency for cut off the upper part of spectra

ftailcorrection=1; Frequency that diagnostic tail apply after that (typically set at 2.5fm, fm=1/Tm01)

tailpower=-4; Power that diagnostic tail apply based on that (-3 for shallow water to -5 for deep water)

mincutoff=’off’;

> **Define if to cut off the spectra below fmin** mincutoff=’off’: Cutoff off
>
> > mincutoff=’on’: Cutoff on

maxcutoff='off';

Define if to cut off the spectra beyond fmax maxcutoff='off': Cutoff off

maxcutoff='on': Cutoff on

tailcorrection='off';

Define if to apply diagnostic tail correction or not tailcorrection='off': Not apply

tailcorrection='jonswap': JONSWAP Spectrum tail

tailcorrection='tma': TMA Spectrum tail

dispout='on'; Define to display outputs or not ('off': not display, 'on': display)

8.3 OUTPUT

Hm0 Zero-Moment Wave Height (m)

Tm01 Wave Period from m01 (second), Mean Wave Period

Tm02 Wave Period from m02 (second), Mean Zero Crossing Period

Tp Peak Wave Period (second)

fp Peak Wave Frequency (Hz)

f Frequency (Hz)

Syy Wave Surface Elevation Power Spectrum (m^2s)

8.4 EXAMPLE

```
[Hm0,Tm01,Tm02,Tp,fp,f,Syy]= WaveSpectraFun(input,10,1024,2^10, 0.05,0.04,2,0.9,-4,'on
↪','on','off','on');
```

8.5 NOTES

In case that data are measured by a pressure sensor, data should be corrected for pressure attenuation. In that case, function "PcorFFTFun.m" or "PcorZerocrossingFun.m" should be called first to correct pressure (water depth) data before calculating wave properties. Results from either of those functions can be imported to this function for spectral analysis of wave parameters. If provided "RunOceanlyz.m" file is used, it will do this procedure if proper input parameters are selected in "oceanlyzinput.m" file. Please read the note in "PcorFFTFun.m".

This function can be used as a standalone command in Matlab/GNU Octave command line or it can be embedded in Matalb/GNU Octave script file (.m file) as:

```
[Hm0,Tm01,Tm02,Tp,fp,f,Syy]=WaveSpectraFun(input,fs,duration,nfft,h,heightfrombed,
↪fmin,fmax,ftailcorrection,tailpower,mincutoff,maxcutoff,tailcorrection,dispout);
```

Example for using a provided sample input file:

```
[Hm0,Tm01,Tm02,Tp,fp,f,Syy]= WaveSpectraFun(input,10,1024,2^10, 0.05,0.04,2,0.9,-4,'on
↪','on','off','on');
```

CHAPTER 9

WaveZerocrossingFun

```
[Hs,Hz,Tz,Ts,H,T]=WaveZerocrossingFun(input,fs,duration,dispout)
```

9.1 DESCRIPTION

Calculate wave properties using Up-going zero crossing method

9.2 INPUT

input=importdata('h.mat'); Load water depth (h)/surface elevation (Eta) data and rename it "input" in (m)

fs=10; Sampling frequency that data collected at in (Hz)

duration=1024; Duration time that data collected in input in each burst in second

dispout='on'; Define to display outputs or not ('off': not display, 'on': display)

9.3 OUTPUT

Hs Significant Wave Height (m)

Hz Zero Crossing Mean Wave Height (m)

Tz Zero Crossing Mean Wave Period (second)

Ts Significant Wave Period (second)

H Wave Height Data Series (m)

T Wave Period Data Series (second)

9.4 EXAMPLE

```
[Hs,Hz,Tz,Ts,H,T]=WaveZerocrossingFun(input,10,1024,'on');
```

9.5 NOTES

In case that data are measured by a pressure sensor, data should be corrected for pressure attenuation. In that case, function “PcorFFTFun.m” or “PcorZerocrossingFun.m” should be called first to correct pressure (water depth) data before calculating wave properties. Results from either of those functions can be imported to this function for zero-crossing calculation of wave parameters. If provided “RunOceanlyz.m” file is used, it will do this procedure if proper input parameters are selected in “oceanlyzinput.m” file. Please read the note in “PcorFFTFun.m”.

This function can be used as a standalone command in Matlab/GNU Octave command line or it can be embedded in Matalb/GNU Octave script file (.m file) as:

```
[Hs,Hz,Tz,Ts,H,T]=WaveZerocrossingFun(input,fs,duration,dispout);
```

Example for using a provided sample input file:

```
[Hs,Hz,Tz,Ts,H,T]=WaveZerocrossingFun(input,10,1024,'on');
```

CHAPTER 10

PcorFFTFun

```
[Eta,ftailcorrection]=PcorFFTFun(input,fs,duration,nfft,h,heightfrombed,fminpcorr,
↪fmaxpcorr,ftailcorrection,pressureattenuation,autofmaxpcorr,dispout)
```

10.1 DESCRIPTION

Apply pressure correction factor to water depth data from pressure gauge reading using FFT

10.2 INPUT

input=importdata('h.mat'); Load water depth (h)/surface elevation (Eta) data and rename it "input" in (m)

fs=10; Sampling frequency that data collected at in (Hz)

duration=1024; Duration time that data collected in input in each burst in second

nfft=2^10; NFFT for Fast Fourier Transform

h=1; Mean water depth in (m)

heightfrombed=0.0; Sensor height from bed

fminpcorr=0.15; Minimum frequency that automated calculated fmaxpcorr can have if autofmaxpcorr='on' in (Hz)

fmaxpcorr=0.8; Maximum frequency for applying pressure attenuation factor

ftailcorrection=1; Frequency that diagnostic tail apply after that (typically set at 2.5fm, fm=1/Tm01)

pressureattenuation='all';

> **Define if to apply pressure attenuation factor or not** pressureattenuation='off': No pressure attenuation applied
>
> pressureattenuation='on': Pressure attenuation applied without correction after fmaxpcorr

pressureattenuation='all': Pressure attenuation applied with constant correction after fmaxpcorr

autofmaxpcorr='on';

Define if to calculate fmaxpcorr and ftailcorrection based on water depth or not autofmaxpcorr='off': Off

autofmaxpcorr='on': On

dispout='on'; Define to display outputs or not ('off': not display, 'on': display)

10.3 OUTPUT

Eta Corrected Water Surface Level Time Series (m)

10.4 EXAMPLE

```
[Eta]=PcorFFTFun(input,10,1024,2^10,1.07,0.05,0.04,0.8,'all','off','on');
```

10.5 NOTES

This function can be used as a standalone command in Matlab/GNU Octave command line or it can be embedded in Matalb/GNU Octave script file (.m file) as:

```
[Eta,ftailcorrection]=PcorFFTFun(input,fs,duration,nfft,h,heightfrombed,fminpcorr,
↪fmaxpcorr,ftailcorrection,pressureattenuation,autofmaxpcorr,dispout);
```

Example for using a provided sample input file:

```
[Eta]=PcorFFTFun(input,10,1024,2^10,1.07,0.05,0.04,0.8,'all','off','on');
```

In case of using a pressure attenuation correction function, choosing a proper upper limit for applying a correction is essential. If the upper limit is chosen unreasonably high, it will lead to a wrong over-estimation of the results. The under-estimation also can happen, if the upper limit is chosen unreasonably low. The waves that are large enough that can be sensed by a sensor should be included in attenuation correction. Note that pressure from small waves with high frequency can be damped in the water column and therefore may not reach a sensor depth. In that case, these small waves should not be included in attenuation correction. Wave height, wave frequency, water depth, height of a sensor above a seabed all play roles on if a wave effect reaches down to a sensor depth or not. Because of that, a deployment situation and wave properties should be used to define the highest frequency that a sensor senses its effect. Pressure correction should not be applied beyond that point. (refer to Applying Pressure Response Factor section)

CHAPTER 11

PcorZerocrossingFun

```
[Eta]=PcorZerocrossingFun(input,fs,duration,h,heightfrombed,dispout)
```

11.1 DESCRIPTION

Apply pressure correction factor to water depth data from pressure gauge reading using up-going zerocrossing method

11.2 INPUT

input=importdata('h.mat'); Load water depth (h)/surface elevation (Eta) data and rename it "input" in (m)

fs=10; Sampling frequency that data collected at in (Hz)

duration=1024; Duration time that data collected in input in each burst in second

h=1; Mean water level (m)

heightfrombed=0.0; Sensor height from bed

dispout='on'; Define to display outputs or not ('off': not display, 'on': display)

11.3 OUTPUT

Eta Corrected Water Surface Level Time Series in (m)

11.4 EXAMPLE

```
[Eta]=PcorZerocrossingFun(input,10,1024,1.07,0.05,'on');
```

11.5 NOTES

This function can be used as a standalone command in Matlab/GNU Octave command line or it can be embedded in Matalb/GNU Octave script file (.m file) as:

```
[Eta]=PcorZerocrossingFun(input,fs,duration,h,heightfrombed,dispout);
```

Example for using a provided sample input file:

```
[Eta]=PcorZerocrossingFun(input,10,1024,1.07,0.05,'on');
```

CHAPTER 12

SeaSwellFun

```
[Hm0,Hm0sea,Hm0swell,Tp,Tpsea,Tpswell,fp,fseparation,f,Syy]=SeaSwellFun(input,fs,
↪duration,nfft,h,fmin,fmax,ftailcorrection,tailpower,fminswell,fmaxswell,mincutoff,
↪maxcutoff,tailcorrection,dispout)
```

12.1 DESCRIPTION

Separate sea wave from swell wave

12.2 INPUT

input=importdata('h.mat'); Load water depth (h)/surface elevation (Eta) data and rename it "input" in (m)

fs=10; Sampling frequency that data collected at in (Hz)

duration=1024; Duration time that data collected in input in each burst in second

nfft=2^10; NFFT for Fast Fourier Transform

h=1; Mean water depth in (m)

fmin=0.04; Minimum frequency for cut off the lower part of spectra

fmax=1; Maximum frequency for cut off the upper part of spectra

ftailcorrection=1; Frequency that diagnostic tail apply after that (typically set at 2.5fm, fm=1/Tm01)

tailpower=-4; Power that diagnostic tail apply based on that (-3 for shallow water to -5 for deep water)

fminswell=0.1; Minimum frequency that is used for Tpswell calculation

fmaxswell=0.25; Maximum frequency that swell can have, It is about 0.2 in Gulf of Mexico

mincutoff='off';

Define if to cut off the spectra below fmin mincutoff='off': Cutoff off

mincutoff='on': Cutoff on

maxcutoff='off';

Define if to cut off the spectra beyond fmax maxcutoff='off': Cutoff off

maxcutoff='on': Cutoff on

tailcorrection='off';

Define if to apply diagnostic tail correction or not tailcorrection='off': Not apply

tailcorrection='jonswap': JONSWAP Spectrum tail

tailcorrection='tma': TMA Spectrum tail

dispout='on'; Define to display outputs or not ('off': not display, 'on': display)

12.3 OUTPUT

Hm0 Zero-Moment Wave Height (m)

Hm0sea Sea Zero-Moment Wave Height (m)

Hm0swell Swell Zero-Moment Wave Height (m)

Tp Peak wave period (second)

Tpsea Peak Sea period (second)

Tpswell Peak Swell Period (second)

fp Peak Wave Frequency (Hz)

f Frequency (Hz)

fseparation Sea and Swell Separation Frequency (Hz)

Syy Wave Surface Elevation Power Spectrum (m^2s)

12.4 EXAMPLE

```
[Hm0,Hm0sea,Hm0swell,Tp,Tpswell,fp,fseparation,f,Syy]= SeaSwellFun(input,10,1024,2^10,
↪1.07,0.05,2,0.9,-4,0.1,0.25,'on','on','off','on');
```

12.5 NOTES

In case that data are measured by a pressure sensor, data should be corrected for pressure attenuation. In that case, function "PcorFFTFun.m" or "PcorZerocrossingFun.m" should be called first to correct pressure (water depth) data before calculating wave properties. Results from either of those functions can be imported to this function for spectral analysis of wave parameters. If provided "RunOceanlyz.m" file is used, it will do this procedure if proper input parameters are selected in "oceanlyzinput.m" file. Please read the note in "PcorFFTFun.m".

This function can be used as a standalone command in Matlab/GNU Octave command line or it can be embedded in Matalb/GNU Octave script file (.m file) as:

```
[Hm0,Hm0sea,Hm0swell,Tp,Tpsea,Tpswell,fp,fseparation,f,Syy]=SeaSwellFun(input,fs,
↪duration,nfft,h,fmin,fmax,ftailcorrection,tailpower,fminswell,fmaxswell,mincutoff,
↪maxcutoff,tailcorrection,dispout);
```

Example for using a provided sample input file:

```
[Hm0,Hm0sea,Hm0swell,Tp,Tpswell,fp,fseparation,f,Syy]= SeaSwellFun(input,10,1024,2^10,
↪1.07,0.05,2,0.9,-4,0.1,0.25,'on','on','off','on');
```

OCEANLYZ changelog

CHAPTER 13

Changelog

13.1 Version 1.5

What is new in ver 1.5:

- Now, a wave spectrum in PcorFFTFun function is obtained from pwelch function instead of fft function
- Parameter ‘burst’ changed to ‘n_burst’
- Parameter ‘duration’ changed to ‘burst_duration’

13.2 Version 1.4

What is new in ver 1.4:

- Now, OCEANLYZ can be run on both Matlab and GNU Octave
- Now, a separate input file is used to define calculation parameters
- Wavenumber calculation performance is improved
- TMA diagnostic tail is improved
- User manual is re-written

13.3 Version 1.3

What is new in ver 1.3:

- Tail correction for spectral analysis is improved
- Correcting pressure data for pressure attenuation is improved
- Sea/Swell partitioning is improved

13.4 Version 1.2

What is new in ver 1.2:

- The NFFT now can be assigned as an input.
- Automatic calculation of the upper limit frequency for the dynamic pressure corrections added.
- Zero-Crossing method modified.
- Mean water level calculation modified.
- Significant wave period added.
- Calculation of peak wave frequency based on the weighted integral added.
- Parameter notations are updated.

CHAPTER 14

Recommended Book

For more details on coastal and ocean wave data analysis refer to:

Karimpour A., (2018), Ocean Wave Data Analysis: Introduction to Time Series Analysis, Signal Processing, and Wave Prediction, KDP.

Book link: https://www.amazon.com/dp/0692109978

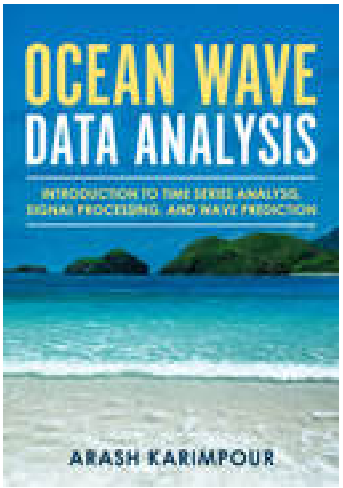
OCEAN WAVE
DATA ANALYSIS
INTRODUCTION TO TIME SERIES ANALYSIS, SIGNAL PROCESSING, AND WAVE PREDICTION
ARASH KARIMPOUR

Fig. 1: Figure: Ocean Wave Data Analysis