coastlib

Release 0.9.1

Georgii Bocharov

Sep 02, 2019

TABLE OF CONTENTS

CHAPTER

ONE

GETTING STARTED WITH COASTLIB

1.1 Installing coastlib

To install coastlib run the following:

```
pip install coastlib
```

To make sure everything works as expected run the tests:

```
py.test -v tests
```

CHAPTER

TWO

DATA EXTRACTION AND PROCESSING TOOLS

Tools related to data extraction and processing are contained within the `data` package available through:

```
from coastlib import data
```

2.1 NOAA CO-OPS Module

The `noaa_coops` module is a part of the coastlib.data package. This module provides interface to the NOAA CO-OPS data portal via the CO-OPS API. It allows retrieval of data collected by CO-OPS sensors such as wind, water levels, currents, salinity, air pressure, etc. in the form of pandas DataFrame. With the help of this tool one can automate extraction of large amounts of data from NOAA stations for further processing and storing.

Core tools from this module are available through either of these commands:

```
>>> from coastlib.data import coops_api, coops_api_batch, coops_datum
>>> from coastlib.data.noaa_coops import coops_api, coops_api_batch, coops_datum
```

An in-depth tutorial for the `noaa_coops` module is available in this Jupyter notebook.

CHAPTER

THREE

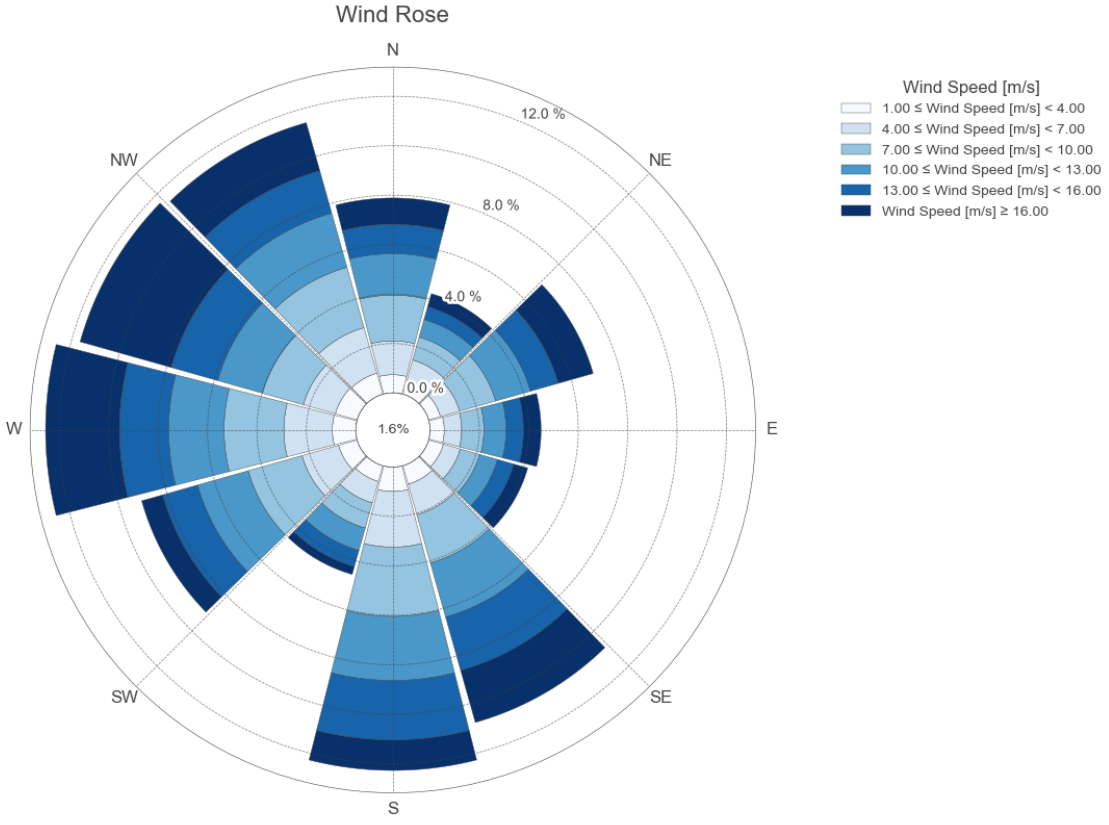
DATA VISUALIZATION TOOLS

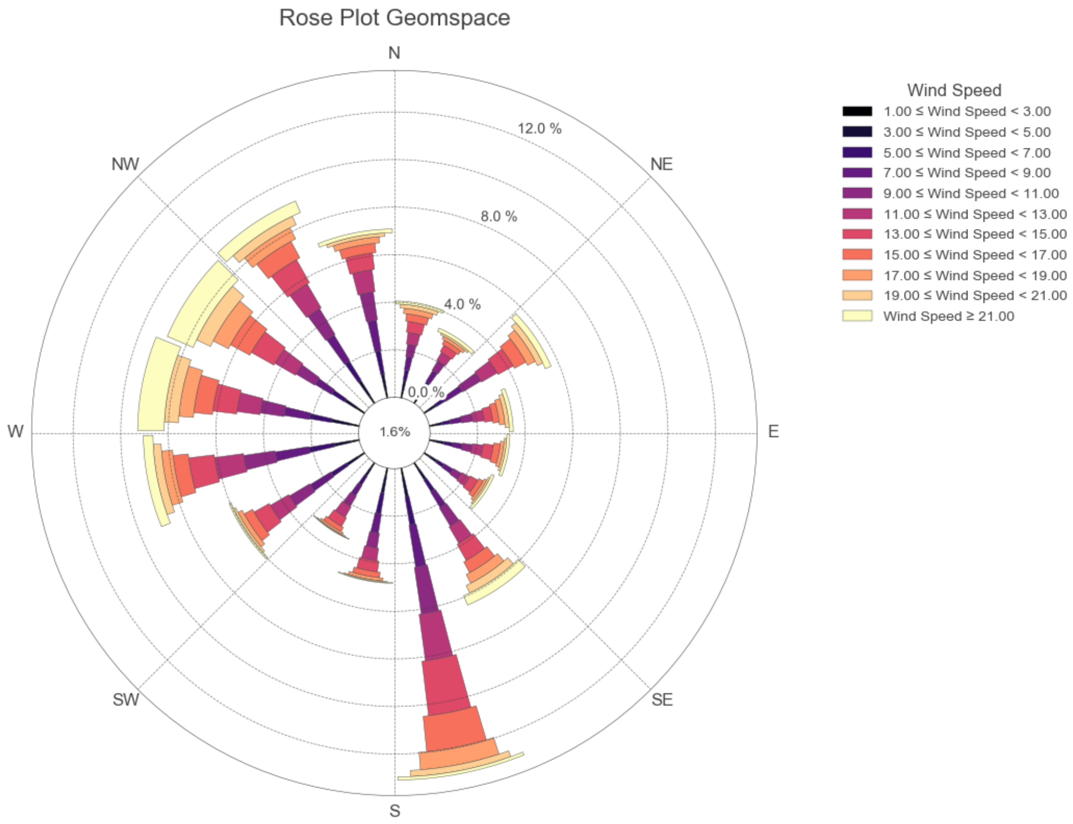
Tools related to data visualization are contained within the `plotting` package available through:

```
from coastlib import plotting
```

3.1 Rose Plot

```
from coastlib.plotting import rose_plot
```




Rose Plot Geomspace
N
NE
E
SE
S
SW
W
NW
12.0 %
8.0 %
4.0 %
0.0 %
1.6%
Wind Speed
1.00 ≤ Wind Speed < 3.00
3.00 ≤ Wind Speed < 5.00
5.00 ≤ Wind Speed < 7.00
7.00 ≤ Wind Speed < 9.00
9.00 ≤ Wind Speed < 11.00
11.00 ≤ Wind Speed < 13.00
13.00 ≤ Wind Speed < 15.00
15.00 ≤ Wind Speed < 17.00
17.00 ≤ Wind Speed < 19.00
19.00 ≤ Wind Speed < 21.00
Wind Speed ≥ 21.00

CHAPTER

FOUR

WATER WAVE ANALYSIS

Tools related to water wave analysis are contained within the `waves` package available through:

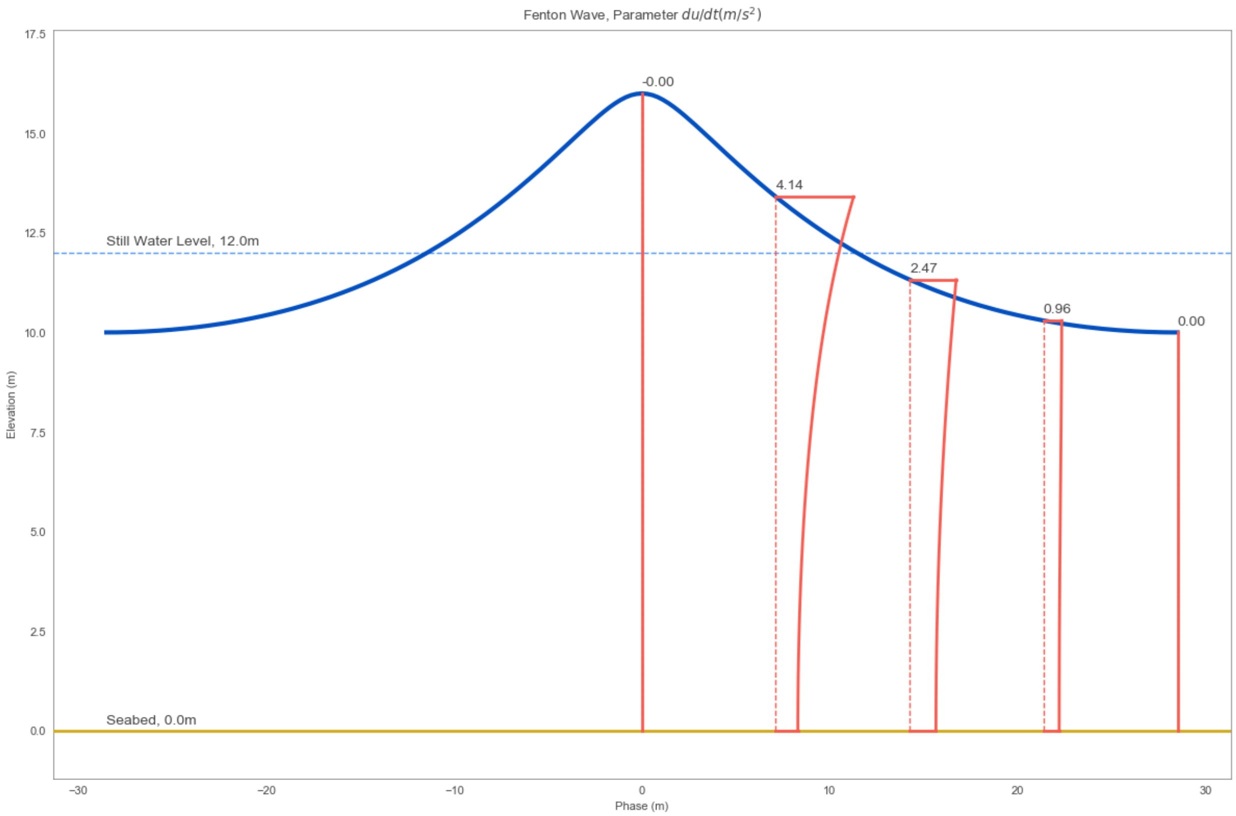
```
from coastlib import waves
```

4.1 FentonWave Module

The FentonWave module is a high level interface to John D Fenton's steady water wave solver Fourier program. The Fourier program solves a steady two-dimensional periodic wave propagating without change of form over a layer of fluid on a horizontal bed. The FentonWave module provides a pythonic way of creating a FentonWave object exposing the steady wave solution results and methods for visualizing wave summary. The FentonWave object stores all data in pandas DataFrame and numpy array objects, which allow for simple integration with other functions and classes.

Shown below is a simple example of using the FentonWave class to calculate a steady wave:

```
>>> from coastlib.waves import FentonWave
>>> wave = FentonWave(wave_height=3, wave_period=6, depth=20)
>>> wave
                            Fenton Wave
    ===========================================================
                                              Unit        Value
    Parameter
    depth                                        m       20.000
    wave length                                  m       56.516
    wave height                                  m        3.000
    wave period                                  s        6.000
    wave speed                                 m/s        9.419
    eulerian current                           m/s        0.000
    stokes current                             m/s        0.058
    mean fluid_speed                           m/s        9.419
    wave volume flux                         m^2/s        1.168
    bernoulli constant r                   (m/s)^2       44.390
    volume flux                              m^2/s      187.220
    bernoulli constant R                   (m/s)^2      240.523
    momentum flux                 kg/s^2 or (N/m)  3813694.427
    impulse                               kg/(m*s)     1197.458
    kinetic energy                kg/s^2 or (N/m)     5639.670
    potential energy              kg/s^2 or (N/m)     5557.035
    mean square of bed velocity            (m/s)^2        0.055
    radiation stress              kg/s^2 or (N/m)     7023.656
    wave_power                   kg*m/s^3 or (W/m)    60062.853
    ===========================================================
>>> wave.plot()
```



An in-depth tutorial for the `waves` module is available in this Jupyter notebook.