



Preface

The CoastalTools™ is a Matlab™ App was developed in response to research being undertaken as part of the [University of Southampton](#) Master's degree courses in Coastal Science and Engineering. This routinely required access and manipulation of the data held by the [Channel Coastal Observatory](#) (CCO) and others as a starting point for further research. Whilst the data held is well documented, with extensive supporting meta-data, accessing the different file formats and making proper use of the quality control information, can be time consuming, thereby limiting the time that can be spent analysing the data. CoastalTools™ aims to address this by providing a framework for time series data analysis, with many of the formats that are in routine use in the UK and elsewhere provided for in the data loading tools. Several models that use the measured data are also provided. These provide the ability to estimate near shore wave parameters and properties that describe how the beaches are changing over time. The models generally use simple algorithms to transform the data and are not intended to be a substitute for more detailed coastal process modelling. The focus is very much on providing a means to rapidly explore the data that is now available from national archives.

Requirements

The model is written in Matlab™ and provided as Open Source code (issued under a GNU General Public License) and runs under v2016b or later. CoastalTools uses the `muitoolbox` and `dstoolbox`.

The following functions developed by others and obtained from the Matlab Exchange Forum, or through personal contact are as follows:

`mat2clip.m` - Copyright © 2009, The MathWorks, Inc.

`wind_rose.m` - Copyright © 2009, M MA

`peakseek.m` - Copyright © 2010, Peter O'Connor

`polarplot3d.m` - Copyright © 2016, Ken Garrard

`mgpdfit.m` (requires statistics toolbox) - Copyright © 2015, Ivan Haigh

`downsample_ts.m` - Copyright © 2014, Chad Greene

`moving.m` - Copyright © 2017, Aslak Grinsted

`inpaint_nans.m` Copyright © 2012, John D'Errico

`utide` (`ut_solv.m` and `ut_reconst.m`) Copyright © 2017, Daniel Codiga

Hurst exponent experimental code derived from several sources in the Matlab™ Exchange Forum, including Conrado Chiarello, 2019; Vilen Abramov, 2018; Athara Aalok, 2022; Ioannis Antoniadis and Guiuseppe Brandi, 2021.

The beach simulation options use the Matlab Global Optimization Toolbox.

Resources

The CoastalTools App and two toolboxes (`muitoolbox` and `dstoolbox`) can be downloaded from www.coastalsea.uk.

Cite as:

Townend, I.H., 2021, CoastalTools manual, CoastalSEA, UK, pp83, www.coastalsea.uk.

Bibliography

Key links and papers setting out the background to the model include:

<http://www.channelcoast.org/introduction/>

http://www.bodc.ac.uk/products/bodc_products/ntslf/

Further sources are detailed in the References towards the end of the Manual.



Revision history

Version	Date	Changes
3.4	Jan 2024	Added function to analyse cycles in tidal range data. Modified handling of obliques waves near shore in the refraction model. Corrected bug in nearshore wave model when using the surf zone depth option (NaN), causing function to misread the offshore depth.
3.3	May 2023	Added option to use wave refraction transfer tables, as produced by the WaveRayModel App, to generate inshore wave climates. Added explanation of the CCO topographic survey feature codes to the manual. Crenulate Bay tool can now use imported image (e.g., aerial) or a profile extracted shoreline as the background for bay plots.
3.2	Nov 2022	Added Hurst statistic and Crenulate Bay to the tool set.
3.1	June 2022	Various bug fixes to plot multiple profiles, order profiles along shore correctly and retain this order when plotting. Check added to Add data to ensure both data sets have same number of variables.
3.0	May 2021	CoastalTools packaged as a Matlab App and migrated to use muitoolbox and dstoolbox.
2.1	Dec 2020	Added tidal analysis and reconstruction options using 'utide' by Daniel Codiga (Matlab™ Forum). Added option to identify shore or profile data on multi-line plots. Fixed issue with ordering of profiles along the coast.
2.0	May 2020	First release in ModelUI framework via www.coastalsea.uk This version supersedes CoastalTools v1.9, which will no longer be supported. With this change it has not been possible to maintain backward compatibility because the class definition used to hold the data has been modified. However, it should be possible to load any data used in the previous version, to create similar projects in this version.



Contents

1	Introduction	1
2	Getting Started.....	1
2.1	Configuration.....	1
2.1.1	Installing the toolboxes	1
2.1.2	Installing the App	1
2.2	CoastalTools™ software	1
2.3	Opening CoastalTools™	2
2.4	Typical workflow for a new project	2
2.5	Sample project.....	4
3	Menus	5
3.1	File.....	5
3.2	Tools.....	5
3.3	Project.....	5
1.1	Setup	6
3.3.1	Import data	6
3.3.2	Site parameters	7
3.3.3	Model parameters.....	9
3.3.4	Data clean-up.....	11
3.4	Run	11
3.4.1	Wave properties.....	11
3.4.2	Beach properties	12
3.4.3	Tidal Analysis.....	15
3.4.4	Derive Output	16
3.4.5	Simulation	17
3.4.6	Vulnerability.....	18
3.4.7	User Model.....	18
3.5	Analysis.....	19
3.5.1	Plotting	19
3.5.2	Statistics.....	22
3.6	Help.....	27
3.7	Tabs	27
3.8	UI Data Selection	28
3.9	Accessing data from the Command Window	29
4	Supporting Information and Model Setup.....	31
4.1	Quality Checks	31
4.1.1	Waves	31



4.1.2	Water levels.....	31
4.2	Nearshore waves.....	32
4.2.1	Refraction, Shoaling and Friction.....	32
4.2.2	Breaking wave height.....	33
4.2.3	Shore profile.....	34
4.3	Deepwater waves.....	34
4.4	Wind-waves.....	34
4.5	Wave power.....	36
4.6	Runup (setup & wash).....	36
4.7	Littoral drift.....	37
4.8	X-shore Transport.....	37
4.9	Overtopping.....	38
4.10	Iribarren number and beach type.....	38
4.11	Tides.....	38
4.12	Beach type.....	39
4.13	Volumes and shoreline position.....	39
4.14	Shoreline Change.....	41
4.15	Crenulate Bay.....	41
4.16	Derive Output.....	42
4.16.1	Calling an external function.....	43
4.16.2	Input and output format for external functions.....	43
4.16.3	Pre-defined functions.....	46
4.16.4	Adding variables to peak and cluster time series.....	50
4.17	Simulation.....	50
4.17.1	YGOR model.....	50
4.17.2	BMV model.....	52
4.18	Beach Vulnerability Index.....	55
4.18.1	Single site BVI (<i>Run> Vulnerability>BVI site:</i>).....	55
4.18.2	Multi-profile BVI (<i>Run> Vulnerability>BVI profile set:</i>).....	56
4.19	Taylor diagram.....	58
4.19.1	Taylor diagram theory.....	58
4.19.2	Skill score.....	59
5	User functions.....	61
5.1	User Data.....	61
5.2	User Models.....	61
5.3	User Statistics.....	61



5.4	User Plots	61
6	Program Structure.....	62
7	Bibliography.....	66
	Appendix A – Input Data File Formats	69
	Appendix B – Data set properties (DSproperties).....	77
	Appendix C - Date and Time Locale.....	78



1 Introduction

CoastalTools was originally designed to enable data held by the [Channel Coastal Observatory](#) (CCO), along with tide data from the [British Oceanographic Data Centre](#) (BODC) to be loaded, analysed, used in a range of simple models and plotted. In principle, the software can handle any timeseries data and has been extended to handle a number of other common file formats. This version has provision for winds, waves, water levels, beach profiles, shorelines and a user defined data set. There are several types of modelling capability relating to:

- wave parameters such as nearshore waves, littoral drift and structure overtopping;
- examining changes in beach volume, or shoreline position;
- simulation of shoreline position and beach profile volume change;
- beach vulnerability index models.

Data and model variables can be used to derive new variables using simple data manipulation expressions. Measured and model data can all be used in the statistical analysis functions to obtain descriptive statistics, peaks and clusters, extreme values, regression and profile interval statistics (the latter provides descriptive statistics of some parameter (eg wave power) for the intervals between beach profile surveys). Finally, the results can be plotted and there is provision for time series plots, profile x-y plots, a range of x-y-z surface plots and wave rose plots.

2 Getting Started

2.1 Configuration

CoastalTools is installed as an App and requires `muitoolbox` and `dstoolbox` to be installed. The download for each of these includes the code, documentation and example files. The files required are:

`dstoolbox`: `dstoolbox.mltbx`

`muitoolbox`: `muitoolbox.mltbx`

The App file: `CoastalTools.mlappinstall`

2.1.1 Installing the toolboxes

The two toolboxes can be installed using the *Add-Ons > Manage Add-Ons* option on the Home tab of Matlab™. Alternatively, right-click the mouse on the ‘`mltbx`’ files and select install. All the folder paths are initialised upon installation and the location of the code is also handled by Matlab™. The location of the code can be accessed using the options in the *Manage Add-Ons* UI.

2.1.2 Installing the App

The App is installed using the Install Apps button on the APPS tab in Matlab™. Alternatively, right-click the mouse on the ‘`mlappinstall`’ file and select install. Again all the folder paths are initialised upon installation and the location of the code is handled by Matlab™.

Once installed, the App can be run from the APPS tab. This sets the App environment paths, after which the App can be run from the Command Window using:

```
>> CoastalTools;
```

The App environment paths can be saved using the Set Path option on the Matlab™ Home tab.

Documentation can be viewed from the Supplementary Software in the Matlab™ documentation. The location of the code can be accessed by hovering over the App icon and then finding the link in the pop-up window.

2.2 CoastalTools™ software

The CoastalTools™ software has been written using the software language Matlab™, version 2016b. The software utilises core Matlab™ functions, except for the extremes analysis function which makes

use of the statistical toolbox and some standalone functions (e.g. dvolstats) which use other toolboxes. Some information on the program structure, that will be of use to those who wish to explore alternative model parameterisations, or develop the software, can be found in Section 6. To support this, several functions have been written as standalone (i.e. not contained within a set of class methods) to allow the user to modify aspects of the model behaviour. These are detailed in Section 5.

2.3 Opening CoastalTools™

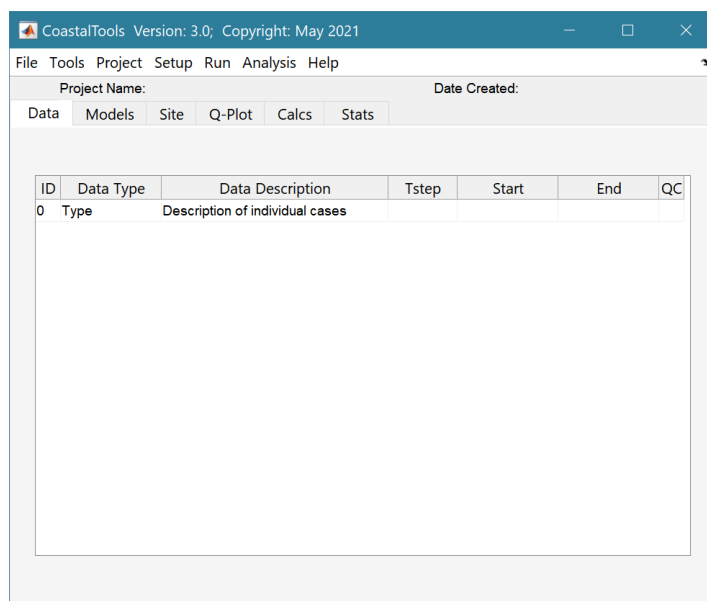
A graphical user interface (GUI) is used to load data, run models, manipulate the data, carry out some simple statistical procedures, plot results and export model output.

If you installed CoastalTools as an App use the icon on the APPS tab, or if the App environment paths have been set, simply type CoastalTools at the command prompt:

>> CoastalTools; OR >>ct = CoastalTools; if you want a handle to the model instance.

A splash screen crediting the developers appears for a few seconds before being replaced by the CoastalTools interface (Figure 1). Documentation can be viewed from App Help menu, or the Supplemental Software in the Matlab™ documentation. The location of the code can be accessed by hovering over the App icon and then finding the link in the pop-up window.

Figure 1 – The CoastalTools GUI



Version number

Drop-down menus

Project title and date

Display Tabs

Description of data sets that have been loaded

2.4 Typical workflow for a new project

The following steps explain how to set-up a model using the various tools provided. Text in *Red italic* refers to drop down menus. Text in *Green italic* refers to Tab titles.

File>New: define project name and date

Setup>Import data>Waves>Load data: load selected CCO wave data from file (when bulk loaded they should be in order and not contain data that overlaps in time)

Setup> Import data> Water levels>Load data: load selected CCO or BODC water level data from file (when multiple files are loaded, they should be in order and not contain data that overlaps in time)

Setup> Import data>Beach profiles>Load data: load selected CCO beach profile data from file (when multiple files are loaded, they should be in order and not contain data that overlaps in time)

Setup>>Add data: load additional files (one at a time)

Setup>>Edit data: open UI to select a time series and edit the values (but not the dates)



Setup>...>Data QC: apply quality control to the data set. This makes use of the quality flags that are included with some data sets and additional measures to remove records that are spurious.

Once loaded, data can be plotted, used to add modelled data, or statistically analysed.

To examine what has been set-up the Tabs provide a summary of what is currently defined. Note: the Tabs update when clicked on, using a mouse, but the values cannot be edited from the Tabs.

Data: lists the data sets that have been loaded.

Models: lists the modelled parameters that have been created.

Site: tabulates the properties required for model calculations, and is split into two tabs:

Site>Waves: tabulates the site properties used in the various wave models and structure properties, as used in overtopping calculations (only one definition at any one time).

Site>Simulation: tabulates the parameter settings required for the YGOR and BMV simulations models (see Sections 3.4.5 and 4.17 for further details).

Q-Plot: provides rapid access to default plotting of any variable in any data set shown on the **Data** or **Models** tabs. These plots cannot be edited and are designed to check data records when loading.

Calcs: tabulates the results from model analysis, and is split into two tabs:

Calcs>Volumes: tabulates the results of the most recent beach volume calculations. Includes a button to copy results to the clipboard.

Calcs>Shoreline: tabulates the results of the most recent shoreline calculations. Includes a button to copy results to the clipboard.

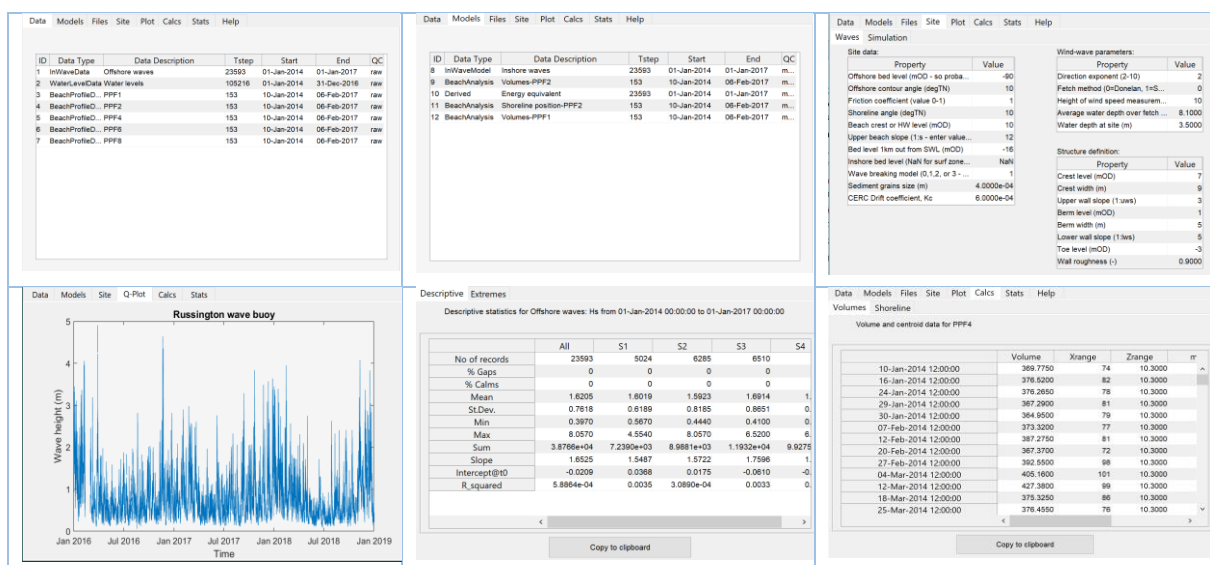
Calcs>Profile: plot an idealised profile and closure depths, or the Dean profile.

Stats: tabulates the results from statistics UI, and is split into two tabs:

Stats>Descriptive: tabulates the results of the most recent descriptive statistics. Includes a button to copy results to the clipboard.

Stats>Extremes: tabulates the results of the most recent extremes analysis. Includes a button to copy results to the clipboard.

Figure 2 – Tabs in Main GUI





NB: *Data* and *Models* tabs provide access to the associated metadata. Click on the first column of a row to generate a table figure with details of the variables for the case and any associated metadata. Buttons on the figure provide access the class definition metadata, source information (files input or models used) and any user data (e.g., tables of derived parameters) that is saved with the data set.

Running a model:

Setup>Site parameters: define the characteristics of the site being studied

Run>Wave properties>: select the type of wave parameters to be calculated

Run>Beach properties>: select the type of beach parameters to be calculated

Statistical analysis:

Analysis>Statistics: launches the statistical analysis UI. Using the *Timeseries* Tab, any data set and variable can be selected and used in a number of different statistical analysis routines, such as general descriptive statistics over the defined time interval, cluster analysis, or extremes analysis.

To plot the results:

Analysis>Plots: launches the plotting UI. Using the *Time* Tab, any data set can be selected and the results plotted for a chosen variable over a user defined time interval.

To write the results to an Excel file:

Project>Cases>Save Data Set: list dialogue allows user to select which data set is to be written to file (only a single data set can be selected at a time). A user input dialogue then prompts for a file name (no extension required).

2.5 Sample project

The *example* folder contains several sample data files for some of the different types of data that can be loaded, and a demonstration project file.

3 Menus

3.1 File

File>New: clears any existing model (prompting to save if not already saved) and a popup dialog box prompts for Project name and Date (default is current date).

File>Open: existing CoastalTools projects are saved as *.mat files. User selects a project file from dialog box.

File>Save: save a file that has already been saved.

File>Save as: save a file with a new or different name.

File>Exit: exit the program. The close window button has the same effect.

3.2 Tools

Tools>Refresh: updates *Data* tab.

Tools>Clear all>Project: deletes the current project (data, models and all project specific variables).

Tools>Clear all>Figures: deletes **all** results plot figures (useful if a large number of plots have been produced).

Tools>Clear all>Data: deletes **all** Data sets listed on the *Data* tab.

Tools>Clear all>Models: deletes **all** models listed on the *Models* tab.

3.3 Project

Project>Project Info: edit the Project name and Date.

Project>Case>Edit Description: select a Case description to edit.

Project>Cases>Edit DS properties: edit the properties that define the meta-data for a dataset.

Project>Case>Edit Data Set: edit a data set. Initialises a data selection UI to select the record to be edited and then lists the variable in a table so that values can be edited. The user can also limit the data set retrieved based on the variable range and the independent variable (X) or time. This can be useful in making specific edits (eg all values over a threshold or values within a date range). This option also works for profiles. Ensure the variable to be edited is selected (Chainage, Elevation, etc). The user is then prompted to select a survey date and the relevant profile data can then be edited. An alternative for cleaning up profile data is *Setup>Data clean-up>Edit or Delete profile in timeseries* (see Section 3.3.4). Using the Copy to Clipboard button also provides a quick way of exporting selected data.

Project>Case>Save Data Set: select the Case to be saved from the list of Cases, select whether to save the Case as a *dstable* or a *table* and name the file. The dataset *dstable* or *table* are saved to a mat file.

Project>Case>Delete Case: user selects the Case(s) to be deleted from a list box of Cases and these are then deleted (the model setup is not changed). NB: once deleted they cannot be recovered.

Project>Case>Reload Case: select a previous model run from the current list of Model Cases and reload the input values as the current input settings.

Project>Case>View Case Settings: display a table of the model input parameters used for a selected Case.

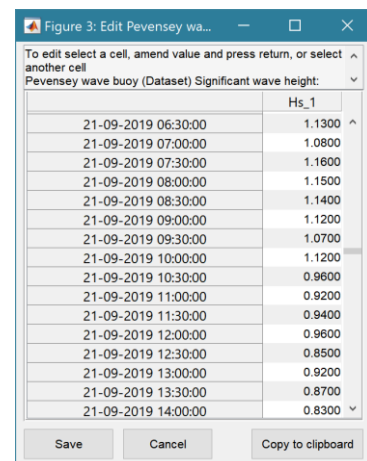


Figure 3: Edit Pevensey wa... — □ ×

To edit select a cell, amend value and press return, or select another cell

Pevensey wave buoy (Dataset) Significant wave height: ▾

	Hs_1
21-09-2019 06:30:00	1.1300
21-09-2019 07:00:00	1.0800
21-09-2019 07:30:00	1.1600
21-09-2019 08:00:00	1.1500
21-09-2019 08:30:00	1.1400
21-09-2019 09:00:00	1.1200
21-09-2019 09:30:00	1.0700
21-09-2019 10:00:00	1.1200
21-09-2019 10:30:00	0.9600
21-09-2019 11:00:00	0.9200
21-09-2019 11:30:00	0.9400
21-09-2019 12:00:00	0.9600
21-09-2019 12:30:00	0.8500
21-09-2019 13:00:00	0.9200
21-09-2019 13:30:00	0.8700
21-09-2019 14:00:00	0.8300

Save Cancel Copy to clipboard

Project>Export/Import>Export Case: saves a Case class instance to a Matlab binary ‘mat’ file.

Project>Export/Import>Import Case: loads data from a Matlab binary ‘mat’ file. Only works for data sets saved using Export. These two functions can be used to move class instances between projects or models.

NB: to export the data from a Case for use in another application (eg text file, Excel, etc), use the *Project>Cases>Edit Data Set* option to make a selection and then use the ‘Copy to Clipboard’ button to paste the selection to the clipboard.

1.1 Setup

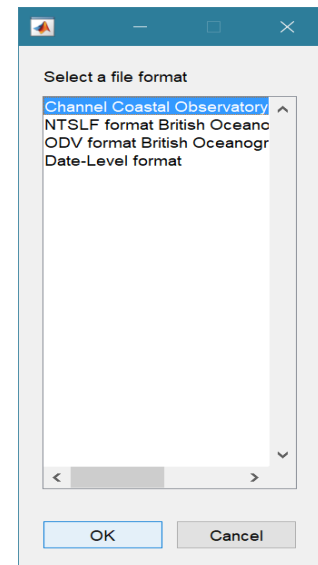
3.3.1 Import data

The setup menu provides a series of menus to enable different types of data to be loaded, edited and quality controlled. The following applies to each type of data (waves, water levels, beach profiles, beach dips, shoreline, BlueKenue file formats and user defined data). For details of User Data see Section 5.1 and for file format information see Appendix A – Input Data File Formats.

Setup>Import data>[Data type]>Load: prompts user for file format to be loaded. The options available vary with Data type. The user then selects one or more files to be loaded. Once files have been read, user is prompted for a description (working title) for the data set.

NB: CCO format Beach profile files use the file name to get the profile id. The file name format for beach profile data must be id_datexxx.txt, eg: 4dMU2_20090824tip.txt.

Files with multiple surveys and multiple profiles can take time to load.



Setup>Import data >[Data type]>Add: prompts user for file to be loaded (only one file at a time can be added). Only files with the format used to create the data set can be used to Add data to a data record (see note below).

Setup>Import data >[Data type]>Quality Control: runs a series of checks on the data. This is only available for waves and water levels in this version. See Section 4.1 for details of the checks applied. Once run, the last column of the **Data** tab is updated to show that qc has been applied.

Files with different formats can be included in a project but they will need to be defined as different data records (i.e., class instances). The format to be used for a given data record is selected when the first file is loaded.

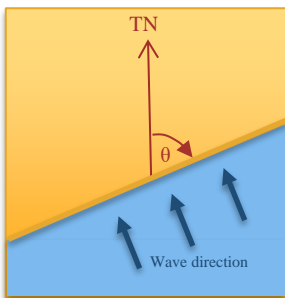
When new file formats are added for a particular Data type, these need to be defined in the class constructor. The additional file required is the same as when adding User Data, as explained in Section 5.1.

3.3.2 Site parameters

Setup > Site parameters > Wave propagation: input dialogue allows user to define characteristic properties of the site. The current values can be viewed on the *Site/Waves* tab.

Bed level offshore, relates to the depth of the offshore wave data.

The angle of the offshore bed contour from True North (degTN) – set this to be the same as Shoreline angle for plane bed refraction. Otherwise an iterative procedure is used between the two angles (see Section 4.2.1).



Friction coefficient can be used to further modify the wave offshore to inshore wave transformation¹.

The angle of the shoreline from True North (degTN) is used to refract the offshore waves to a point near the shoreline.

Beach crest level or high water level (mOD)

Upper beach slope is the beach slope above mean tide level (MTL) used to extend the idealised profile from MTL to high water².

Bed elevation 1km from shore defines an idealised profile to compute beach slope across the surf zone. There is an option to use a co-ordinate definition of an offshore point (y, z) where y is the distance from MTL and z is the bed elevation at that distance offshore (see Section 4.2.3).

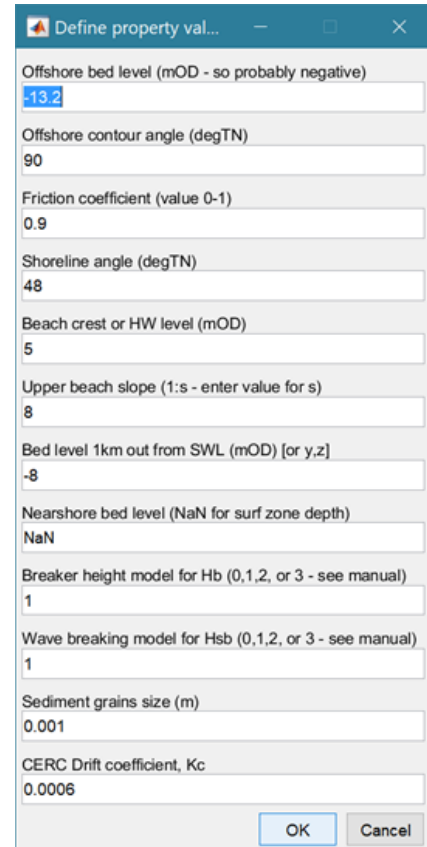
Inshore bed level is the depth used to calculate inshore wave conditions. Set to NaN to use an iterative procedure that finds the depth at the edge of the surf zone.

Breaker height model used to calculate monochromatic breaking wave height (see Section 4.2.2).

Wave breaking model takes a value of 0, 1, 2 or 3 and defines the model to be used (see Section 4.2.2).

Grain size is a representative d50 diameter for the sediments that make up the beach.

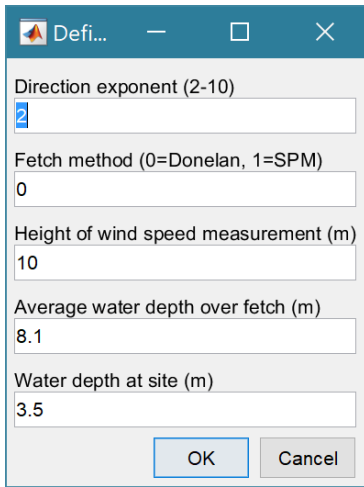
The Drift coefficient is as used in the CERC longshore drift formula.



¹ A value of 1 results in no friction loss and a value 0 results in total wave damping – see Section 4.2.1.

² Note that in CoastalTools slopes are all defined as the horizontal distance, bs, for a unit change in the vertical distance. Hence the slope angle, $\tan(\theta) = 1/bs$ and the slope is often expressed as 1:bs.

Setup > Site parameters > Wind-wave hindcast: input dialogue to define model parameters for the wind-wave hindcast model.



This is the exponent used to define the amount of directional wave spreading. A local wind wave has a small value (large spread) typically around 2, a swell wave has a larger value (narrow spread) with a value of about 10.

The 0=Donelan method, uses a $1/\cosh$ directional spreading function, whereas 1=SPM method uses a \cos function. A value of -1 uses the fetch along the wind direction.

Height at which the input wind data was measured (m).

Average water depth over the model domain. As this will vary with direction an average value for the offshore area can be used (or one based on the dominant fetches).

Water depth at the site (m) – used to apply spectral saturation for limiting water depths.

See the Shore Protection Manual (SPM) and other references given below for details.

Setup > Site parameters > Structure parameters: input dialogue to define the parameters used when calculating wave overtopping. The current values can be viewed on the *Site/Waves* tab.

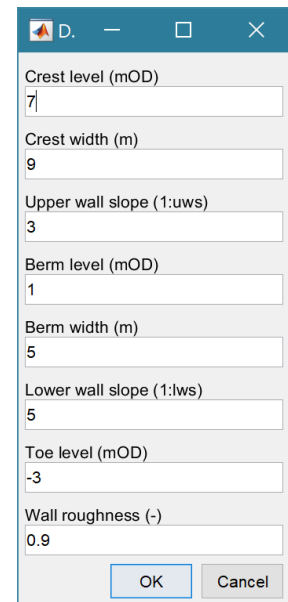
Crest level: top level of beach or structure.

Crest width: horizontal extent of crest.

The model allows for an upper and lower beach or wall slope that may or may not be separated by a horizontal berm.

Toe level: the elevation of the beach or structure toe.

Wall roughness: a coefficient to account for frictional effects on the beach or wall.





3.3.3 Model parameters

Setup>Model parameters>YGOR simulation parameters: used to define the fit coefficients to be used in YGOR simulations (see Sections 3.4.5 and 4.17.1). The current values can be viewed on the *Site/Simulation* tab.

Forecast origin position (m):	66
Forecast origin date (dd-MM-yyyy):	01-06-2014
Equilibrium slope, a:	-0.0075481
Equilibrium intercept, b:	0.090721
Accretion rate coefficient, C+:	-2.0688
Erosion rate coefficient, C-:	-1.099
Offset to initial shore position:	-0.0005179
Fitted coefficients RMSE (read only)	3.7886

The Origin position and date allow the forecast to be adjusted so that the forecast position matches the origin on the specified date.

The values for equilibrium slope, a, equilibrium intercept, b, accretion rate, C+, erosion rate, C- and the initial offset are the fit parameters for the YGOR model obtained by fitting a data set (see Section 4.17.1).

This field is used to report the Root Means Square Error on the *Site/Simulation* tab when the YGOR fitting function has been used to derive the fit parameters a, b, C+, C- and offset.

Setup>Model parameters> BMV simulation parameters: used to define the parameters needed to run the BMV simulation (see Section 3.4.5 and 4.17.2). The current values can be viewed on the *Site/Simulation* tab.

Sediment sphericity, S , from literature or data

Beach porosity, ϕ , from literature or data

Typical sediment grain size of the beach sediment (in metres).

Indicative upper dune slope, for the profile data set

Indicative upper beach slope, of the profile data set (used to initialise estimate of runup slope).

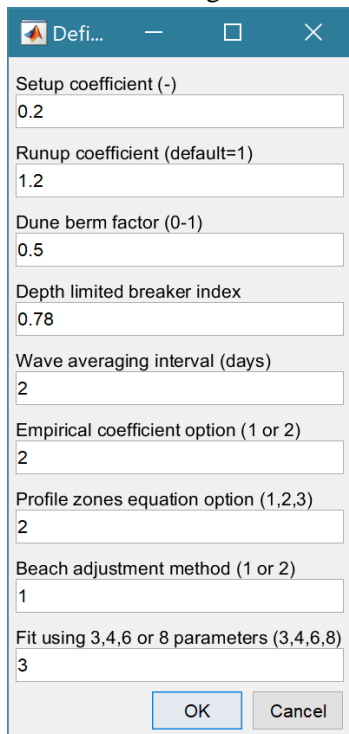
Indicative beach crest level, Z_c for the profile data set, or co-ordinate pair, X_c, Z_c .

High water level used to define upper limit of the equilibrium profile

Low water level, used to define lower limit of the equilibrium profile

The minimum distance over the study period from the origin to high water

The UI on the right is followed by the following UI:



Defi... [OK] [Cancel]

Setup coefficient (-)
0.2

Runup coefficient (default=1)
1.2

Dune berm factor (0-1)
0.5

Depth limited breaker index
0.78

Wave averaging interval (days)
2

Empirical coefficient option (1 or 2)
2

Profile zones equation option (1,2,3)
2

Beach adjustment method (1 or 2)
1

Fit using 3,4,6 or 8 parameters (3,4,6,8)
3

Setup coefficient, scaling of wave setup at the shoreline (used in runup slope formula)

Runup coefficient, scales runup based on Iribarren number

Dune berm factor, proportion of berm and slope between toe of dune and the limit of wave runup.

The breaker index to use (defines the limit of the surf zone closure options 1 and 2.

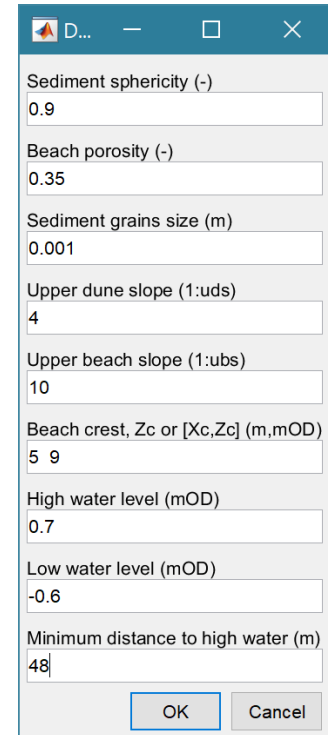
The period (in days) prior to the date of a measured profile over which wave data are averaged.

Empirical coefficient option for the equilibrium profile:

1. Using coefficients defined in Bernabeu *et al* (2003b)
2. Using coefficients defined in Bernabeu *et al* (2003a)

Define the limits of the surf and shoal zones using one of 3 options

1. $h_r = \text{breaker index} \times H_s$, $h_a = 0.008gT^2$
2. $h_r = \text{breaker index} \times H_s$, $h_a = 3H_s$
3. h_r and h_a defined using Hallermeier equations (Hallermeier, 1981).



D... [OK] [Cancel]

Sediment sphericity (-)
0.9

Beach porosity (-)
0.35

Sediment grains size (m)
0.001

Upper dune slope (1:uds)
4

Upper beach slope (1:ubs)
10

Beach crest, Z_c or $[X_c, Z_c]$ (m,mOD)
5 9

High water level (mOD)
0.7

Low water level (mOD)
-0.6

Minimum distance to high water (m)
48

Beach adjustment method:

1. Using the “minimum profile” for all adjustments
2. Using the profile from the previous time step

Fit using 3 or 7 parameters:

1. Using the parameters for the “dune” profile: i.e. setup coefficient, runup coefficient, dune berm factor defined above.

2. Using the upper “dune” profile parameters, as in option 1, and the equilibrium profile coefficients.

The initial estimates defined by the option selected for the Empirical coefficients are relaxed to fit the entire profile.

Setup>Model parameters>Model constants: Various constants are defined for use in models, such as the acceleration due to gravity, viscosity and density of sea water, and density of sediment. Generally, the default values are appropriate (9.81, 1.36e-6, 1025 , 2650 respectively) but these can be adjusted and saved with the project if required.

3.3.4 Data clean-up

Setup>Data clean-up>Concatenate two timeseries: allows two timeseries data sets to be joined. Two records, or two variables, of a similar type can be joined to form a single timeseries or timeseries collection. The user is prompted to select the timeseries records to be used (only records of the same type are shown for the second selection). If there is an overlap in time, the user is prompted to select which timeseries to use for the overlap. The option is then provided to create a new timeseries based on selecting a Single variable or a timeseries collection that concatenates All variables. For single variables they can have different variable names. For All variables, the selected records must have the same set of variables with the same names. This is useful to merge data sets from different sites to extend the record.

Setup>Data clean-up>Resample timeseries: allows a selected timeseries to be resampled at user specified interval, using a user specified method (e.g. mean/max/min over the interval). This can be useful to reduce size of a data set (e.g. long tidal records before using the data in models).

Setup>Data clean-up>Patch timeseries: allows gaps in a selected timeseries to be patched using the data from another timeseries that overlaps the primary timeseries (at least for some or all of the gaps). The function assumes that the primary data set has time intervals defined with no data (NaNs). These times are matched with coincident times in the timeseries used to make the patch (there is currently no interpolation in time) and the data values of the patch timeseries are added to the primary timeseries.

Setup>Data clean-up>Trim timeseries: allows the start and end dates of a timeseries to be modified.

Setup>Data clean-up>Delete multiple profiles: allows all profile data sets with less than ‘N’ surveys available to be deleted. The user selects a threshold for the minimum number of time steps required to retain a profile. Useful for reducing the number of profiles when analysing long-term change.

Setup>Data clean-up>Edit or Delete profile in timeseries: options to scroll through all profiles in the data set of select individual profiles and then to either edit or delete individual profiles. For editing profiles this is easier to use than the *Project>Cases>Edit Data Set* option.

3.4 Run

3.4.1 Wave properties

Run> Wave properties>Deepwater Waves:

Run> Wave properties>Nearshore Waves:

Run> Wave properties>Wind-waves:

Run> Wave properties>Spectral Transfer:

Run> Wave properties>Wave Power:

Run> Wave properties>Runup:

Run> Wave properties>Littoral Drift:

Run> Wave properties>X-shore Transport:

Run> Wave properties>Overtopping:

Run> Wave properties>Iribarren Number:

All of these models require the Site parameters to have been defined (except Spectral Transfer)³. Overtopping also requires the Structure parameters to be defined (even if this is being used to calculate beach overtopping). If there is only one wave data source of offshore data, the models run without additional input. If there is more than one possible source, the user is prompted to select a wave data source. The same is true for water levels, although the model will also run if there are no water levels available, and uses a still water level of 0mOD. The first three models listed above use imported data as input, the subsequent models all use a nearshore wave data set as input. Hence the Nearshore waves model needs to be run before running these other models.

The results are listed on the *Models* tab and can be checked visually on the *Q-Plot* tab or selected for use in one of the analysis or plotting UIs.

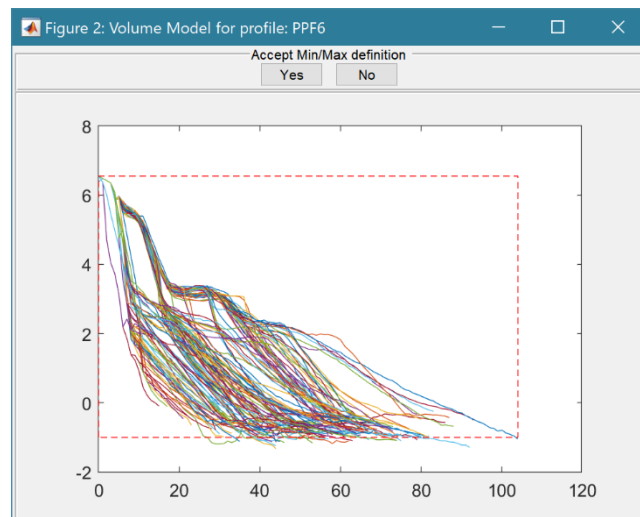
The Wind-Wave model requires a fetch file to be loaded (see Appendix A for format). For each location to be modelled, the fetch file defines the fetch exposure as a function of direction. The file Test_Fetch_Lengths.txt is an example file. This defines the fetch lengths at the site in 10-degree intervals, measured from true north (defined at intervals from 0 to 350 degrees, even if the fetch is zero for some directions).

The Spectral Transfer option uses a Spectral Transfer table from the WaveRayModel to transfer an offshore timeseries to an inshore point. This requires a SpectralTransfer table to have been imported from the WaveRayModel App using *Project> Export/Import>Import Case*. For further details of the how the model works and input requirements see the WaveRayModel App manual.

The Littoral Drift model prompts the user to select which formulation to use. See Section 4.7 for further details on the options. Note: this model uses the shoreline angle defined in Site settings (which can be different to the value used to compute the nearshore wave climate).

3.4.2 Beach properties

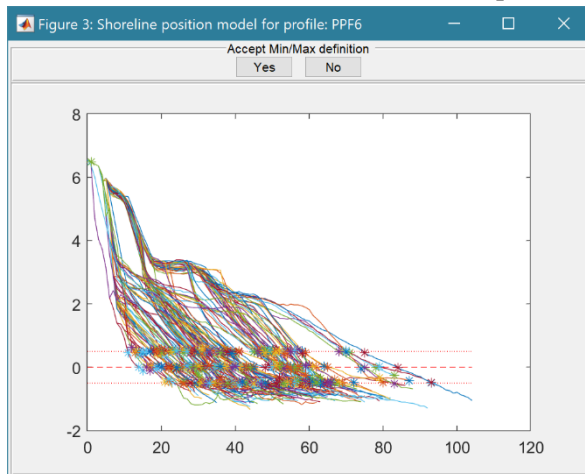
Run> Beach properties>Profiles>Volumes: a list box prompts the user to select a beach profile from those currently loaded. Once the data is retrieved a plot is shown with all of the profiles for the selected location and a separate dialogue box prompting the Minimum X distance and Minimum Z elevation. Values are suggested based on the data available but these can be adjusted to suit the analysis. ON clicking OK, a defining rectangle is added to the plot (dashed red line) and there are also buttons to accept or reject the definition. If the definition is rejected, the user is prompted to amend the Minimum values. Once accepted the analysis proceeds and the resulting volume parameters (strictly area per metre run of beach) are listed on the *Models* tab.



³ Models use the current settings of the *Setup> Site parameters>Wave propagation* when the model is run. The Deepwater and Nearshore wave models use the following site parameters: offshore bed level, inshore bed level, angle of contours at offshore wave data location, angle of contours at shoreline, friction coefficient, beach crest level, upper beach slope, bed level 1km from shore, flags for wave breaking model; The other models use the following parameters where required: upper beach slope, bed level 1km from shore, angle of shoreline from north, grain size d50, drift coefficient.

The results are also tabulated on the *Calcs/Volumes* tab and can be copied to the clipboard for use in other applications.

Run > Beach properties > Profiles > Shoreline position: the analysis of shoreline position is very similar to beach volumes. Initially the user selects a profile from the list provided. A figure then shows the profiles and initial level at which shoreline position (x distance) will be evaluated, together with the



upper and lower bounds for estimating the average slope at the selected elevation. Once completed the plot is redrawn showing the mid-point and upper and lower bounds on each profile that will be used in the analysis of position and slope. The buttons on the figure allow the user to reject the solution and define a new elevation and limits or proceed to the analysis. The start distance is for profiles that have a crest and backslope. The start distance should be set beyond the distance of the crest to avoid two sets of distances being found (i.e. on the front and back sides of the beach).

The results are also tabulated on the *Calcs/Shoreline* tab and can be copied to the clipboard for use in other applications.

Beach properties are saved with a label generated from the source data set. If more than one analysis is performed with the same source data set the output has the same label with the addition of a sequential number. To edit the default label use *Project > Cases > Edit Description*.

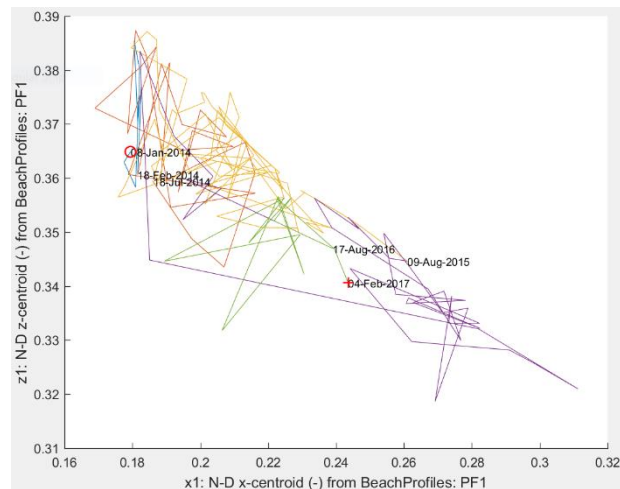
Run > Beach properties > Profiles > Location plot: if some beach profiles have been loaded the first profile at each location can be plotted to produce a map of profile locations.

Run > Beach properties > Profiles > Centroid plot: if volumes for a set of profiles have been calculated (use Profiles > Volumes) a plot of the movement of the profile centroid can be plotted. To produce a plot of the beach centroid over time:

- (i) run the Volumes model for a profile;
- (ii) select *Run > Beach properties > Profiles > Centroid plot*;
- (iii) select the profile to use;
- (iv) select whether to number or assign a time stamp to the output;
- (v) select the dates to subdivide the record into intervals (hold down 'Ctrl' button to make multiple selections, or just accept without making any selections to plot the full trace as one line);
- (vi) select how frequently the output should be numbered or time-stamped. Default value of zero only labels the dates defined in step (v). If a marker interval is defined (>0) then time labels are added at this interval but the line work is coloured according to the sub-periods defined in step (v);

- (vii) data input ui shows the minimum x and z values of the current axes. Accept or modify the extent to plot the same range on both axes press OK, or press Cancel to leave the plot unchanged. This can be useful to avoid visual distortion of the x and z dimensions.

This produces the style of plot shown here.



Run> Beach properties>Profiles>Space-time plot: if volumes or shoreline position have been calculated, a surface plot of the change in a volume property (vol, m0, x1, z1, bs) or a position property (chainage, slope) can be generated for a selected set of profiles and plotted as a function of time (x) and alongshore position (y).

Run> Beach properties> Shore change>Shoreline: a list dialogue allows the user to select several profiles (or use Select All button) to define a beach “set”. The elevation that is to be used to define the “shoreline” is then requested. This routine makes use of some of the properties set in *Setup> Site parameters>Wave propagation*, including the angle of the shoreline from True North (degTN). beach crest level, or high-water level (mOD), the upper beach slope above mean tide level (MTL) and the bed elevation 1km from shore (see Sections 3.3.2 and 4.2.3). The output is saved as both:

- Chainage: the movement of the “shoreline” as a distance from the baseline as a new variable in the dataset for each profile.
- Eastings & Northings: the Easting and Northing location of the “shoreline” on each profile.
- Beach slope (as bs in 1:bs) of the profile over the range +/-0.5m of the shoreline elevation.
- Orientation of the extracted shoreline at the defined elevation.
- Regression parameters for Chainage, Slope, and Orientation (shoreline angle degTN).

Run> Beach properties> Shore change>Change plot: generates a plot of the E,N shoreline positions for the selected beach “set” as a series of shorelines for each time sample. Output can also be plotted as a function of profile location along the shore rather than E,N points. This allows the changes in chainage (distance from profile baseline), beach slope and beach orientation to be plotted.

Run> Beach properties> Shore change>Rates plot: generates a plot of the rate of change of shoreline position and beach slope for the selected beach “set”. The plots show the slope of the regression line for each variable and the associated standard error. The rate of shoreline change is in m/yr. The beach slope is defined in terms of the horizontal distance, bs, for a unit change in vertical elevation and the rate of change of bs is therefore also m/yr.

Run> Beach properties>Beach Type: uses inshore wave data to estimate the dimensionless fall velocity (which can be used to characterise beach type – see Section 4.10). If there is more than one wave data set available, the user is prompted to select one. Results can be used to identify beach type: as: $dfv \geq 6$: dissipative, $dfv < 6$ & $dfv > 1$: intermediate, $dfv \leq 1$: reflective.

Run> Beach properties>Crenulate Bay: plots crenulate (also known as zeta) bays on an image or shoreline. Select image or shoreline and either import image (e.g., an aerial jpg, png, etc), or use a set of beach profiles to generate a shoreline over the user selected sub-set at a user specified profile elevation. The user is then prompted to interactively select a spiral control point and the end point of

the bay on the plot. A further UI prompts for the wave crest angle, the angle the wave crest makes with the control line, and the direction of spiral rotation. The resulting crenulate bay is added to the plot and the UI reappears, allowing input values to be modified. Press Cancel to exit the updating loop. Note there is an option to retain or delete crenulate bay linework with each iteration. However, delete removes all previous linework.

For further details of the method used see Section 4.15.

Run> Beach properties>Shore profile: plots an idealised profile on the *Calcs>Profile* tab. The user is prompted to select an inshore wave data set and then either define a wave condition (H_s and T_p) or specify a percentile value to use. The profile is based on the parameters defined using *Setup>Model parameters> BMV simulation parameters* (see Section 3.3.2). Closure Option 3 gives Z_a and Z_r based on Hallermeier's algorithm (Hallermeier, 1981). Other options are based on the set-up for the BMV model (see Section 4.17.2).

Run> Beach properties>Dean profile: plots an idealised Dean profile on the *Calcs>Profile* tab. The profile is based on the parameters defined using *Setup> Site parameters>Wave propagation*: (see Section 3.3.2). This profile model is used to compute shoreface slopes in the wave propagation model that iterates to find wave heights on the edge of the surf zone (see Section 4.2.3).

3.4.3 Tidal Analysis

Water level data can be analysed to obtain the tidal constituents, or the constituents can be used to reconstruct a tide only water level time series (see Section 4.11 for further details).

Run>Tidal analysis>Analysis: a list box prompts the user to select a water level data set and this is passed to `ut_solv` with the 'lat' property left blank, 'const' set to 'auto' and the following options included: 'NodsatNone','White','NoDiagn'. For details of the methods used and alternative analysis options see the `ut_tide` manual. The estimated constituents are output to the Command Window and the source data, predicted record and the differences are plotted. The predicted record and constituents can be saved.

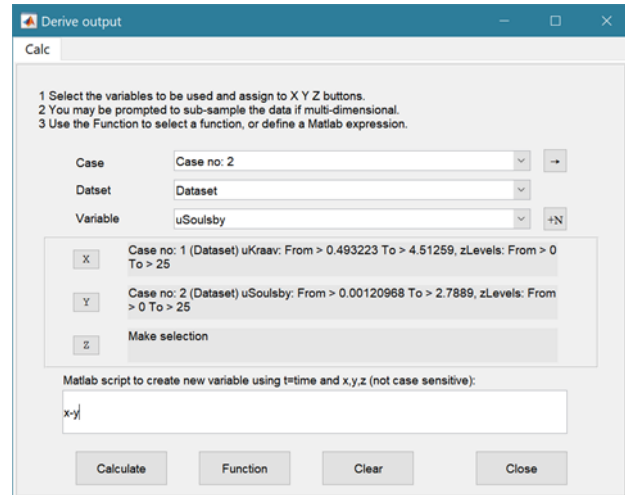
Run>Tidal analysis>Reconstruction: a list box prompts the user to select an existing tidal analysis data set and then to define the data range and time interval to be used for the predictions. The saved constituents are used to predict the tides for the defined date range and saved as a dataset using `ut_reconstr`.

Water level data can be used to generate tidal constituents and predict tidal data for periods of time not covered by the water level data. The source water level data can then be merged with the tidal data to give a more complete record using *Setup>Data clean-up>Patch timeseries*, selecting the water levels as the primary dataset and tidal dataset for the patches.

3.4.4 Derive Output

Time series data that have been added (either as data or modelled values) can be used to derive new variables using the *Run>Derive Output* menu option. The UI allows the user to select data and use a chosen selection of data/variable/range to define either a Variable, or Time. Each data set is sampled for the defined data range. If the data set being sampled includes NaNs the default is for these to be included (button to right of Var-limits is set to '+N'). To exclude NaNs press the button so that it displays '-N'.

The selection is assigned by clicking one of the X, Y or Z buttons. The user is prompted to assign a Variable, XYZ dimension, or Time (the options available varies with the type of variable selected) – see Section 3.8 for details of how this works.



An equation is then defined in the text box below using the x, y, z or t variables⁴. Based on the user selection the routine applies the defined variable ranges to derive a new variable. In addition text inputs required by the call and the model object (mobj) can also be passed.

Adding the comment %time or %rows, allows the the row dimension to be added to the new dataset. For example if x and y data sets are timeseries, then a Matlab™ expresion, or function call, call can be used to create a new time series as follows:

```
x^2+y %time
```

The output from function calls can be figures or tables, a single numeric value, or a dataset to be saved (character vectors, arrays or dstables). External functions should return the table RowNames (e.g., time or location) as the first variable (or an empty first variable), followed by the other variables to be saved.

If there is no output to be passed back the function should return a string variable.

If `varout = 'no output'`; this suppresses the message box, which is used for single value outputs. For expressions that return a result that is the same length as one of the variables used in the call, there is the option to add the variable to the input dataset as a new variable. In all there are three ways in which results can be saved:

1. As a new dataset;
2. As an additional variable(s) to one of the input datasets;
3. As an additional variable(s) to some other existing dataset.

For options 2 and 3, the length of the new variables must be the same length (numeroe of rows) as the existing dataset.

An alternative when calling external functions is to pass the selected variables as dstables, thereby also passing all the associated metadata and RowNames for each dataset selected. For this option up to 3 variables (plus time if defined for a selected variable) can be selected but they are defined in the call using dst, for example:

```
[time,varout] = myfunction(dst, 'usertext', mobj);
```

⁴ Various pre-defined function templates can be accessed using the 'Function' button. Alternatively, text can be pasted into the equation box from the clipboard by right clicking in the text box with the mouse.

```
dst = myfunction(dst, 'usertext', mobj);
```

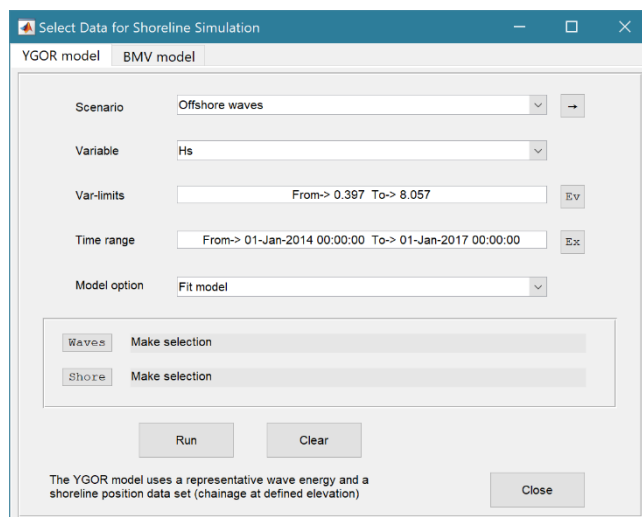
This passes the selected variables as a struct array of dstables to the function. Using this syntax the function can return a dstable, or struct of dstables, or a number of variables. When a dstable, or struct of dstables is returned, it is assumed that the dsproperties have been defined in the function called and dstables are saved without the need to define the meta-data manually.

Some further details on using this option and the ‘**Function**’ library available are provided in Section 4.16

3.4.5 Simulation

The simulation menu option provides access to two simulation models: the so called YGOR and BMV models. Whilst both models simulate change for a single cross-shore transect (profile), the YGOR model simulates the change in shoreline position at some chosen elevation over time. In contrast, the BMV model considers an idealised profile and examines how the section volume and centroid change over time. Further background to the models is provided in Section 4.17.

The two models are accessed via the respective tabs on the Simulation UI. The UI’s for the two models are similar but the selections required for each model are different.



For the **YGOR model** the inputs required are:

1. a timeseries of the forcing variable, such as wave energy. Yates et al (2009) a surrogate for wave energy (e.g. $H_{s1}^2/16$) which must be setup as new data set derived from the Nearshore wave timeseries using the Derive Output UI (*Run>Derive Output*).
2. a timeseries of shoreline positions. These can be loaded as a data set, or generated from a profile data set using the *Run>Beach properties>Profiles>Shoreline position: model*.

The two data sets are assigned to the Waves and Shore buttons, respectively. Any bounds on the data selection, for magnitude or time, should be defined using the Var-limits and Time range boxes before assigning the variable to the Wave or Shore buttons. A description of the selection made is given in the text box alongside the button.

To derive a set of fit coefficients from an observed data set use the ‘Fit model’ option. The computes a set of fit coefficients which can be saved, viewed on the *Site/Simulation* tab and edited using the *Setup>Run parameters>YGOR simulation parameters* menu option (see Section 3.3.3).

To use the model to make a projection/forecast use the ‘Simulation’ option. For this option, the fit coefficients to be used need to have been defined using the *Setup>Run parameters>YGOR simulation parameters* menu option (see Section 3.3.3). When running a simulation, as well as selecting a wave variable for the desired period, it is necessary to select the shoreline data set that was used to “fit” the model. Further explanation and details of the workflow are given in Section 4.17.1.

On the ‘Model option’ drop-down list there is also an option to ‘Preload mean forcing’. For very large data sets loading the data can take some time. When using the ‘Fit model’ option to try multiple fits this can be time consuming. This option allows the mean values of the forcing between profile surveys

to be added to the selected shoreline position data set. Subsequent runs using the ‘Fit model’ option, use these saved mean values rather than re-calculating them. If different wave climates are used, or different variables are used, in Preload mean forcing, the mean values are saved as a new variable. When running Fit Model, the user is prompted to select a mean data set of there is more than one mean wave data set available.

NB. The mean forcing values between profile surveys are used to make an initial estimate of the fit parameters ‘a’ and ‘b’ but are not used for the subsequent optimisation which uses the full wave timeseries.

The model is initialised using the Run button and the selection can be cleared using the Clear button.

The BMVmodel is under development

3.4.6 Vulnerability

There are two options accessible from the Run menu to examine beach vulnerability:

Run > Vulnerability > BVI site: runs a form of Beach Vulnerability Index for a single location. A prerequisite is that there must be at least one case available for each of the following models:

- Nearshore waves: *Run > Wave properties > Nearshore Waves*
- Runup: *Run > Wave properties > Runup*
- Littoral drift: *Run > Wave properties > Littoral Drift*
- Cross-shore transport: *Run > Wave properties > X-shore Transport*
- Shoreline: *Run > Beach properties > Shore change > Shoreline*

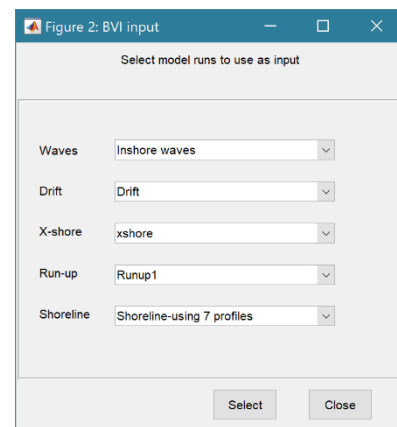
Different scenarios can be considered by generating multiple cases of the above models and then selecting different combinations in the Vulnerability UI. Results are averaged over the profiles used in the selected Shoreline model case. The tabulated results are presented in a stand-alone figure. For details of how the BVI is calculated see Section 4.18.1.

Run > Vulnerability > BVI profile set: runs a more complex version of the Beach Vulnerability Index for a beach defined by a set of beach profiles. The models used are similar to those for the *Single site* option but are run for all profiles as part of the simulation rather than being run beforehand. The prerequisites are that the Site Run parameters for the wave model have been defined and the Nearshore wave model has been run (see Sections 3.3.2 and 3.4.1). The user selects which profiles to include to define a beach “set” of profiles, defines the number of years to be considered as the exposure period (default is 1 year) and selects which nearshore wave data set to use (if there are more than one). The tabulated results are presented on the *Calcs > Shoreline* tab. The results for the Mean Tide Level (0mOD) shoreline change used in the analysis can be viewed in the same way as other Shore change results using the *Run > Beach properties > Shore change > Change plot* and *Rates plot* menu options. For details of how the BVI is calculated see Section 4.18.2.

Run > Vulnerability > BVI set plot generates a plot of the indices as a function of beach profiles along the shore.

3.4.7 User Model

This class is provided for the user to add their own models by defining a new model class (see Section 5.2 for further details on how to add a model).



3.5 Analysis

Analysis allows data and model variables to be plotted or used in various forms of statistical analysis.

3.5.1 Plotting

Analysis>Plots: initialises the Plot UI to select variables and produce various types of plot. The user selects the Case, Dataset, and Variable to be used and the plot Type from a series of drop-down lists, Figure 3. There are then buttons to create a New figure or Add or Delete variables from an existing figure.

Alongside the Data set drop down menus there are various button, which alter properties, or selection:

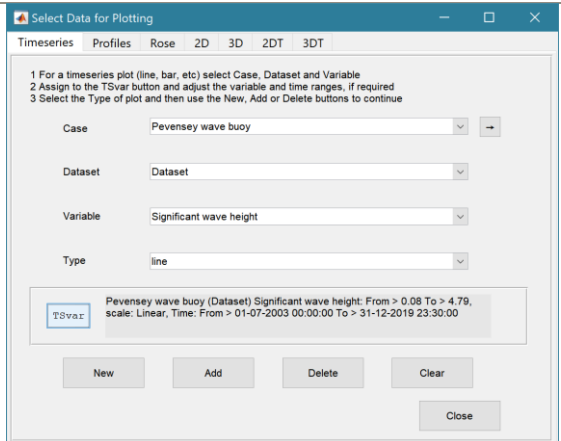
- : updates the list of Cases
- XY : swaps the X and Y axes
- + : switches between cartesian and polar plot type

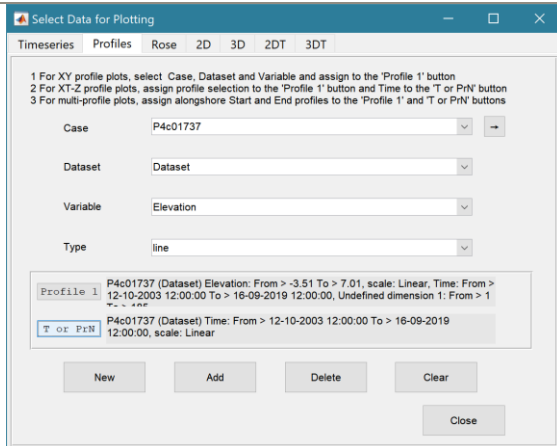
[NB: not all buttons are included on all plotting UIs].

The plotting UI has seven tabs:

- Timeseries - to handle time series plotting;
- Profiles - to handle plotting of beach profiles;
- Rose - for wind and wave rose plots, or polar plots;
- 2D - for a range of types of plot with two variables (line, scatter, bar etc);
- 3D- for a range of surface plots
- 2DT-animates a 2D plot
- 3DT- animates a 3D plot

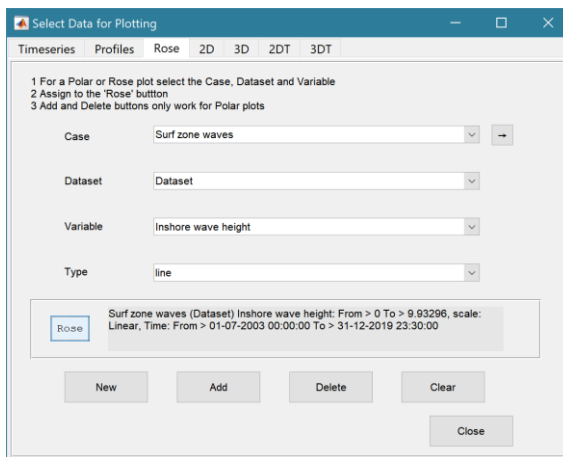
Figure 3 - Plot menus and notes on their use

	<p>Time series plot</p> <p>The X variable is time and does not need to be set. > Select a Case, Dataset and Variable to plot.</p> <p>The Y variable can be scaled (log, normalised, etc) and the range to be plotted can be adjusted from the range values when assigning a variable to the TSvar button. > Select plot type (line, bar, scatter, stem, etc)</p> <p>Buttons:</p> <p>New – generate a new figure Add – add data to an existing figure Delete – remove selected data from an existing figure</p>
---	--



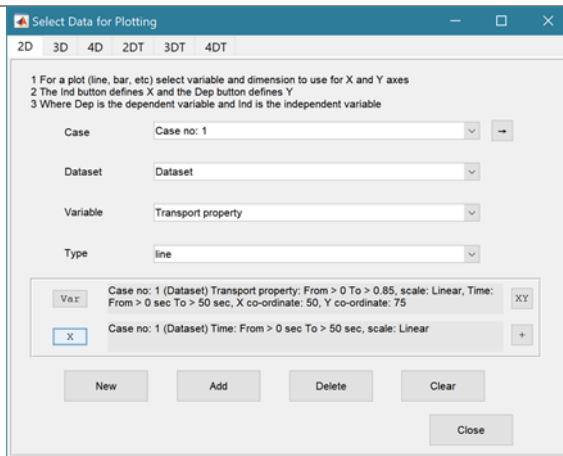
Profiles plot

- > Select Case, Dataset and Variable (only cases that are beach profile data are shown).
- > For X-Y chainage-elevation plots of one or more profiles use the New/Add/Delete buttons. The popup UI lists the surveys available. This UI returns after each plot until Cancel is used to exit.
- > By assigning a selection to the 'Profile 1' button and time to the 'T or PrN' button, a surface plot of the variable as a function of chainage and time is produced.
- > By assigning a start profile to the 'Profile 1' button and an end profile to the 'T or PrN' button, plots of the variable as a function chainage and alongshore position are produced. The option is provided to plot individual survey dates, or an animation. Animation figures include buttons to replay and/or save the output. Alternatively, use the Save button on the Plot Selection UI to save the most recently created animation as an mp4 or avi file. The Clear button to resets the UI to an empty selection



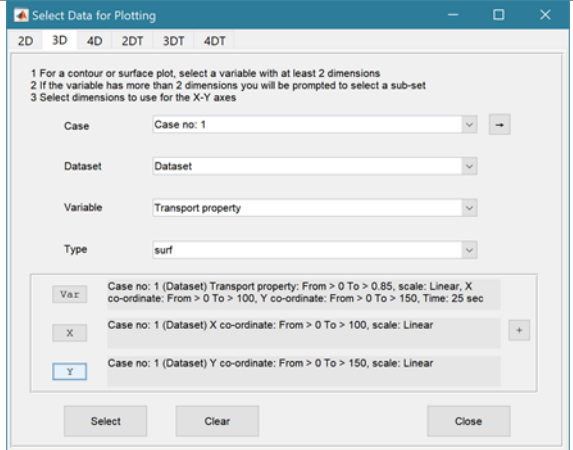
Rose plot

- The Direction variable is selected by default.
- > Select Scenario, Variable, Scaling, Limits, date range and Plot type.
- Prompted to use a polar or a rose plot.
- Additional variables can be added and deleted for polar plots. A second rose plot can be added to a rose plot but there is no delete option for this plot type.



2D plot

- For each selection choose the Case, Dataset and Variable to be used.
- > Assign a variable, or a dimension, to the Var and X buttons to set the Y and X axes, respectively
- Each selection can be scaled (log, normalised, etc) and the range to be plotted can be adjusted when assigning the selection to a button.
- > Select plot type (line, bar, scatter, stem, etc)
- Control Buttons:**
- : updates the list of Cases
 - XY : swaps the X and Y axes
 - + : switches between cartesian and polar plot type
- If polar selected then Ind assumed to be in degrees.*



3D plot

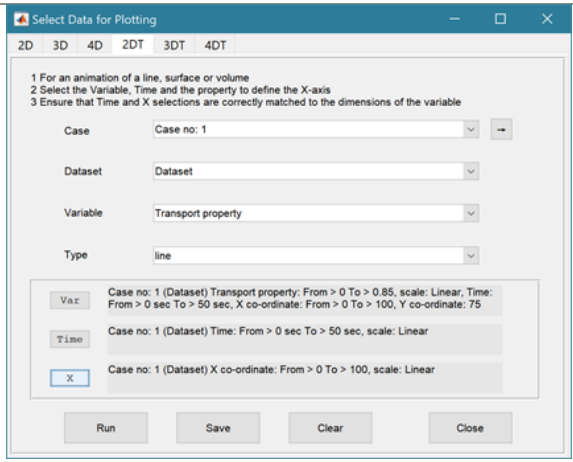
For each selection choose the Case, Dataset and Variable to be used.

> Assign selections to the Var, X and Y buttons

Take care to ensure that the assignments to X and Y correctly match the dimensions selected for the variable (including any adjustment of the dimension ranges to be used).

> Select plot type.

Control Buttons: see 2D plot above.



2DT animation

For each selection choose the Case, Dataset and Variable to be used.

> Assign a variable, or a dimension, to the Var, Time and X buttons.

Each selection can be scaled (log, normalised, etc) and the range to be plotted can be adjusted when assigning the selection to a button.

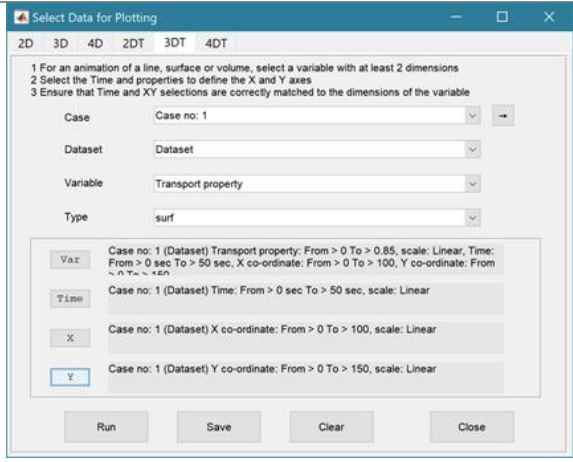
> Select plot type (line, bar, scatter, stem, etc)

Control Buttons:

→ : updates the list of Cases

+ : switches between cartesian and polar plot type

If polar selected, then X assumed to be in degrees and when prompted select Polar and NOT Rose.



3DT animation

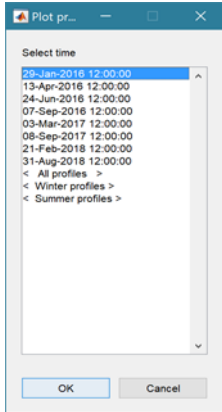
For each selection choose the Case, Dataset and Variable to be used.

> Assign selections to the Var, Time, X and Y buttons

Take care to ensure that the assignments to Time, X and Y correctly match the dimensions selected for the variable (including any adjustment of the dimension ranges to be used).

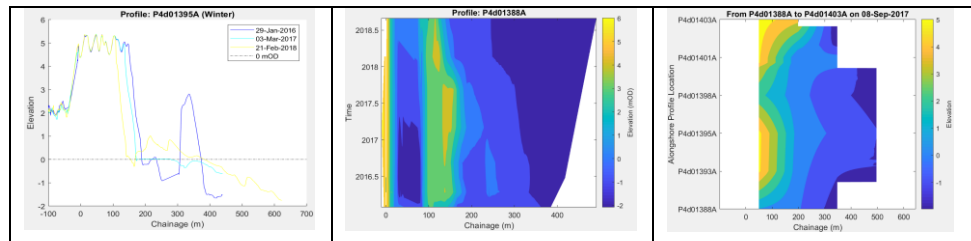
> Select plot type.

Control Buttons: see 2DT plot above.



The selection UI for plotting profiles on the profiles tab includes the options to plot ‘All profiles’, ‘Winter profiles’, or ‘Summer profiles’. These options are appended to the end of the list of survey dates. Winter is defined as October-March and Summer is defined as April-September. Multiple profile plots have a colour order that proceeds from dark blue for the earliest profile to dark red for the most recent. When there are too many profiles to include a legend a colour bar is included instead. When a profile is clicked on with the mouse the date is displayed in the Matlab™ Command Window. This applies to all multi-line plots that have a date stamp.

Example plots:



It is easy to rapidly generate a lot of plots. When the Plotting UI is closed, the user is given the option to delete all the plots generated by the Plotting UI. Alternatively, all the current plots can be deleted using *Tools>Clear all>Figures*.

Setting the plot Type to User calls a user defined function as explained in Section 5.4.

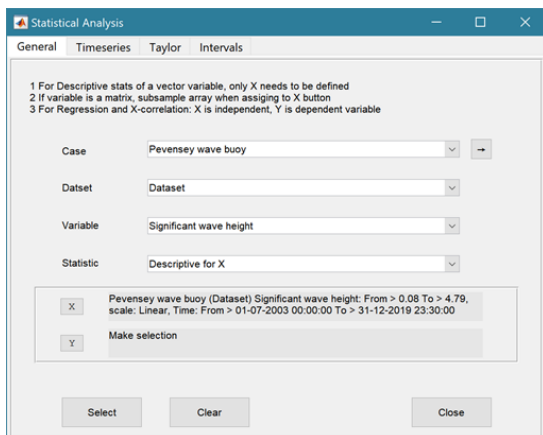
3.5.2 Statistics

Analysis> Statistics: several statistical analysis options have been included within the Statistical Analysis GUI. The tabs are for *General* statistics, *Timeseries* statistics, model comparisons using a *Taylor* Plot, and the generation of a new record based on the statistics over the *Intervals* defined by another timeseries.

General tab

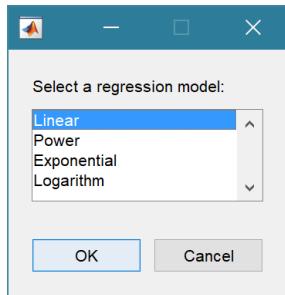
The General tab allows the user to apply the following statistics to data loaded in ModelUI:

- 1) **Descriptive for X**: general statistics of a variable (mean, standard deviation, minimum, maximum, sum and linear regression fit parameters). Only X needs to be defined. The range of the variable can be adjusted when it is assigned to the X button (see Section 3.8). If the variable being used is a multi-dimensional matrix (>2D), the user is prompted to define the range or each additional dimension, or select a value at which to sample. The function can return statistics for a vector or a 2D array.



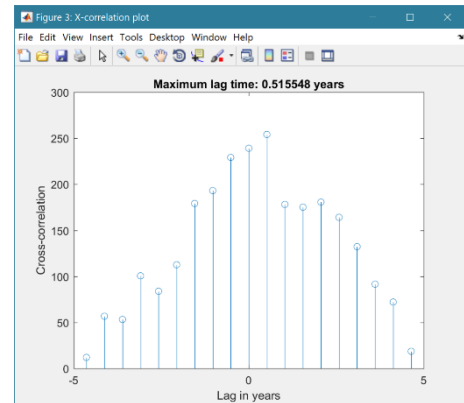
Only X needs to be defined. The range of the variable can be adjusted when it is assigned to the X button (see Section 3.8). If the variable being used is a multi-dimensional matrix (>2D), the user is prompted to define the range or each additional dimension, or select a value at which to sample. The function can return statistics for a vector or a 2D array.

The results are tabulated on the *Stats>General* tab and can be copied to the clipboard for use in other applications.



2) **Regression:** generates a regression plot of the dependent variable, Y, against the independent variable, X. For time series data, the default data range is the maximum period of overlap of the two records. For other data types the two variables must have the same number of data points. After pressing the Select button, the user is prompted to select the type of model to be used for the regression. The results are output as a plot with details of the regression fit in the plot title.

3) **Cross-correlation:** generates a cross-correlation plot of the reference variable, X, and the lagged variable, X (uses the Matlab 'xcorr' function). For time series data, the default data range is the maximum period of overlap of the two records. For other data types the two variables must have the same number of data points. This produces a plot of the cross-correlation as a function of the lag in units selected by the user.



4) **User:** calls the function user_stats.m, in which the user can implement their own analysis methods and display results in the UI or add output to the project Catalogue (see Section 5.3 for further details). Currently implements an analysis of clusters as detailed for Timeseries data below.

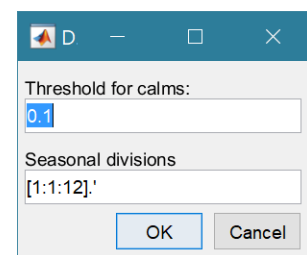
Timeseries tab

The Timeseries tab allows the user to select a single Timeseries variable and apply any of the following statistics:

1) **Descriptive:** general statistics of a variable (mean, standard deviation, minimum, maximum, sum and linear regression fit parameters). The results are tabulated in a new window and can be copied to the clipboard for use in other applications.

Various 'seasonal' sub-divisions can be defined. The required option is selected from the table in the UI, by selecting a Syntax cell and then closing the UI.

The next UI prompts for a threshold for calms (values below threshold are deemed to be "calm" conditions) and allows the selected 'seasonal' divisions to be changed (if the desired option is not in the default list), or edited. The divisions can be expressed in several ways, as detailed below:



Script	Result
1	Descriptive statistics for the full-time series
[1:1:12].'	Descriptive statistics for the full-time series and monthly values (the .' creates a column vector).
[12,1,2; 3,4,5; 6,7,8; 9,10,11]	Descriptive statistics for the full-time series and seasons based on groupings – Dec-Feb, Mar-May, Jun-Aug, Sep-Nov shown.

When seasonal statistics are produced with more than 2 seasons a plot is generated. This can be a cartesian or polar plot of the mean values with error bars used to depict +/- one standard deviation. The polar plot maps the year as one revolution.

- 2) **Peaks:** generates a new timeseries of peaks over a defined threshold. There are three methods that can be selected:
 - 1 = all peaks above the threshold;
 - 2 = the peak value within each up-down crossing of threshold; and
 - 3 = peaks that have a separation of at least '*tint*' hours.
 - 4 = maximum values that are separated by at least '*tint*' hours.For options 3 and 4⁵, the separation between peaks ('*tint*') is also defined in the pop-up UI. This can be used to try and ensure that peaks are independent. The peaks are marked on a plot with the defined threshold. If rejected, new values can be defined. If accepted a new timeseries is added. This has the class of the Data Type that was used as the source timeseries but is not appended to that timeseries because the date/times are a subset of the source.
- 3) **Clusters:** The selection process is similar to peaks, where the user defines a threshold, selection method and time between peaks (for method 3). In addition, the cluster interval is defined in days. This is the period of time separating two peaks for them to be no longer considered part of a cluster (e.g. if a sequence of storms occurs every few days they will form a cluster. If there is then a gap of, say, 31 days to the next storm, with a cluster time interval of 30 days this would be considered as part of the next cluster). Once a selection has been made, a plot is generated that shows the peaks for each cluster with a different symbol. The user can either choose a different definition, or accept the definition. Once accepted, the results are added as a new timeseries, with the class of the Data Type that was used as the source timeseries. Two values are stored at the time of each peak in the clusters: the magnitude of the peak; and the number of the cluster to which it belongs (numbered sequentially from the start). This allows the data for individual clusters to be retrieved, if required.
- 4) **Extremes:** The selection process is similar to peaks, where the user defines a threshold, selection method and time between peaks (for method 3). A figure is generated with two plots. The left-hand plot shows the peaks for the defined threshold and the right-hand plots shows the mean excess above the threshold (circles), the 95% confidence interval (dotted red lines) and the number of peaks (vertical bars + right hand axis) as a function of threshold. This plot can be used to help identify a suitable threshold for the peak-over-threshold extremes analysis method. The user can either choose a different definition, or accept the definition. Once accepted, the user is prompted to select a plot type. Options are: None; Type 1 – a single return period plot; Type 2 – a composite plot showing the probability, quantile, return period and density plots. See Coles (2001) for further details of the method used and the background to these plots. The results are tabulated on the *Stats/Extremes* tab and can be copied to the clipboard for use in other applications.
- 5) **Poisson Stats:** user is prompted to select a threshold, method and peak separation (see Peaks above) and the function generates a plot of the peak magnitude, time between peaks (interarrival time) and the duration above the threshold for each peak. The plot shows a histogram of each variable and the exponential pdf derived from the data, along with the μ value for the fit.
- 6) **Hurst Exponent:** user is prompted to select from one of 3 methods, which are based on different computation routines taken from the Matlab Forum, as follows:
 - 1= Chiarello matrix method
 - 2=Abramov loop code

⁵ Option 3 finds the peaks (values that are greater than the values before and after) and checks for a separation between the peaks. Option 4 sorts the date in descending order and finds the maximum values in the record that are separated by the required interval.

3=Aalok-Ihlen code

4=Aste using unweighted option.

Methods 1 and 2 are similar, whereas method 3 explores the effect of scale and method 4 derives the unweighted generalized Hurst exponent. The main difference between the first two methods is that Abramov uses a different form for S, rather than the Matlab standard deviation function (std).

The Hurst parameter H is a measure of the extent of long-range dependence in a time series (while it has another meaning in the context of self-similar processes). H takes on values from 0 to 1. A value of 0.5 indicates the absence of long-range dependence. The closer H is to 1, the greater the degree of persistence or long-range dependence. H less than 0.5 corresponds to a lack of persistence, which as the opposite of LRD indicates strong negative correlation so that the process fluctuates violently. H is also directly related to fractal dimension, D, where $1 < D < 2$, such that $D = 2 - H$.

This is experimental code (for code see `.../muitoolbox/psfunctions/hurst_exponent.m`, `hurst_aalok_ihlen.m` and `genhurstw.m`) and the user should refer to the background literature for further details. (Di Matteo *et al.*, 2003; Pacheco *et al.*, 2008; Ihlen, 2012; Morales *et al.*, 2012; Sutcliffe *et al.*, 2016; Abramov and Khan, 2017; Antoniadis *et al.*, 2021; Brandi and Di Matteo, 2021).

- 7) **User:** calls `user_stats.m` function, where the user can define a workflow, accessing data and functions already provided by the particular App, or the `muitoolbox` (see Section 5.3 for further details). The sample code can be found in the `psfunctions` folder and illustrates the workflow to produce a clusters plot. Some code in the header (commented out) shows how to get a time series using the handles passed to the function (`obj` and `mobj`). This code would get the same timeseries as the one passed to the function. However, by modifying the 'options' variable it is possible to access other timeseries variables.

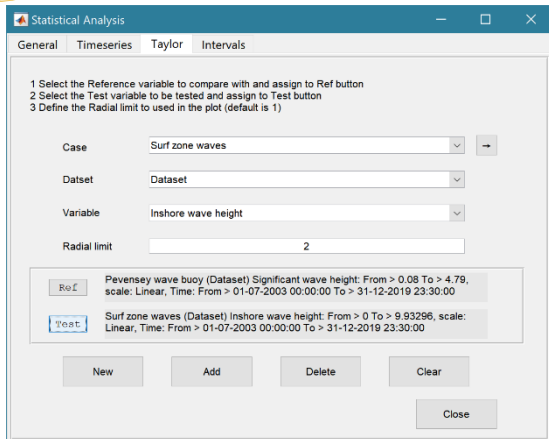
Taylor tab

The Taylor tab allows the user to create a Taylor Plot using 1D or 2D data (e.g timeseries or grids):

A Reference dataset and a Test dataset are selected. Datasets need to be the same length if 1D, or same size if 2D. If the data are timeseries they are clipped to a time-period that is common to both, or any user defined interval that lies within this clipped period. The statistics (mean, standard deviation, correlation coefficient and centred root mean square error) are computed, normalized using the reference standard deviation and plotted on a polar Taylor diagram (Taylor, 2001).

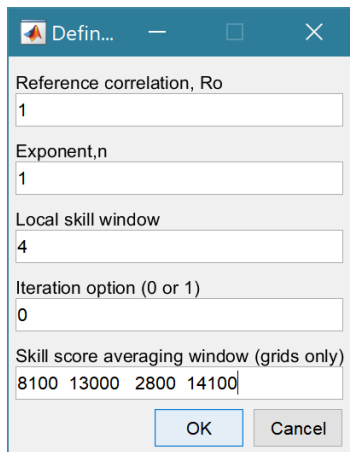
[The ModelSkill App provides additional tools to test data and the ModelSkill App manual provides further details of the methods used.]

Selecting New generates a new Taylor Plot. Selecting the Add button adds the current selection to an existing plot and the Delete button deletes the current selection. The Clear button resets the UI to a blank selection.



Once New or Add are selected, the user is asked whether they want to plot the skill score (Yes/No). If Yes, then the user is prompted to set the skill score parameters. As further points are added to the plot, this selection remains unchanged (i.e. the skill score is or is not included). To reset the option it is necessary to close and reopen the Statistics UI.

If the number of points in the Reference and Test datasets are not the same the user is prompted to select which of the two to use for interpolation.



This is the maximum achievable correlation (see Taylor (2001) for discussion of how this is used).

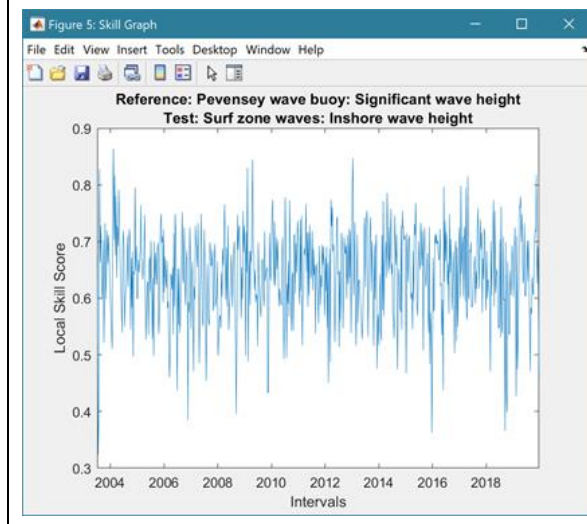
Exponent used in computing the skill score (see Section 4.19.2).

Number of points (+/-W) used to define a local window around the ith point. If W=0 (default) the local skill score is not computed.

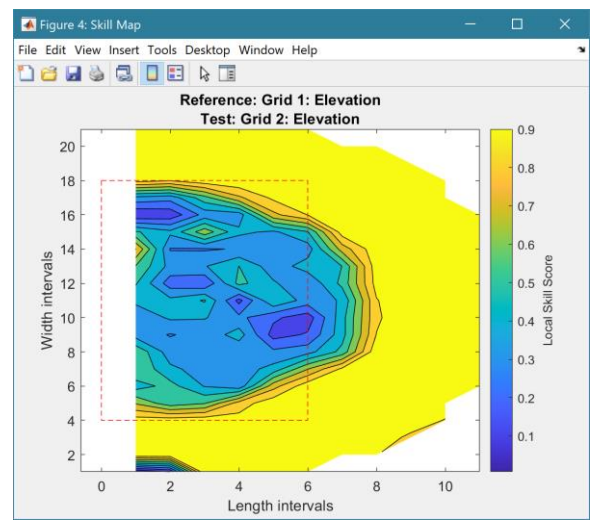
Local skill score is computed for window around every grid cell (=1), or computes score for all non-overlapping windows (=0)

Window definition to sub-sample grid for the computation of the average **local** skill score. Format is [xMin, xMax, yMin, yMax].

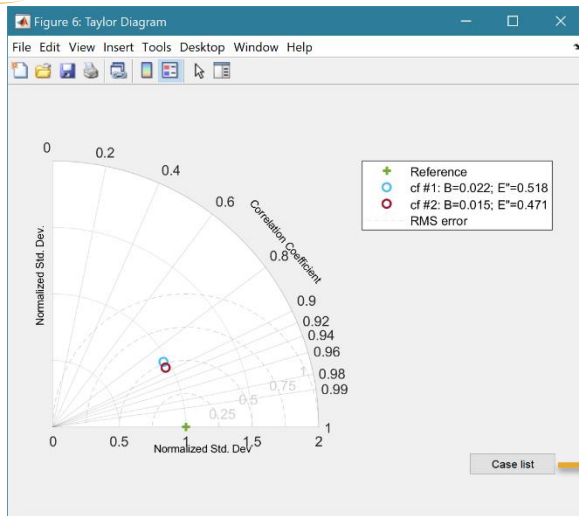
(a) time series skill score plot



(b) grid skill score plot

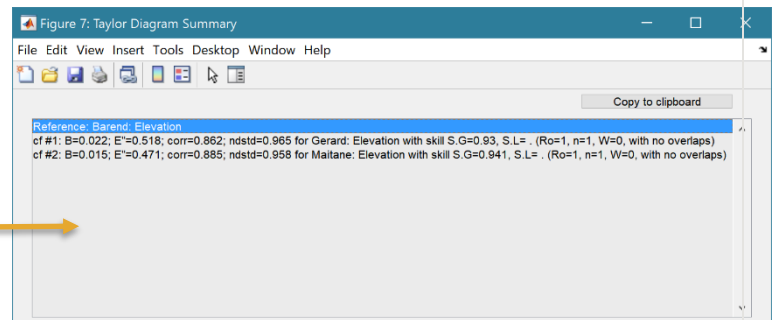


The Taylor Plot shows the Reference point as a green cross and the Test points as coloured circles. The legend details the summary statistics and the Case List button generate a table figure listing all the results. These can be copied to the clipboard.



Taylor diagram legend includes: B – bias; E' – normalised RMS difference

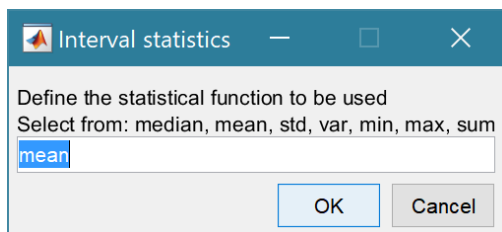
The normalised standard deviation and correlation coefficient are also given in the Case List table, along with the global skill score, S_g, and the average local skill score, S_l.



Intervals tab

The Intervals tab allows the user to compute selected simple statistics of a timeseries variable for the intervals between the times recorded in another timeseries. For example, the mean wave power between beach profile surveys.

The Reference dataset defines the time intervals to be used. As only the times are used it does not matter which variable is selected. The Sample dataset is the timeseries variable to be used. The analysis is run using the Select button and the user is prompted to define the statistical function.



The UI provides a list of the functions that can be used. [Any function that simply requires a single variable as an input could be used. To modify the list, edit the variable 'statoptions' in `muiStats.getIntervalStats.`]

Default properties are derived from the source variable and the selected statistic. The user is prompted to confirm, or edit these default properties. The results are saved as a new timeseries dataset in which the variable describes the statistical value for the interval preceding each time.

3.6 Help

The help menu provides options to access the App documentation in the Matlab™ Supplemental Software documentation, or the App manual.

3.7 Tabs

To examine what has been set-up the Tabs provide a summary of what is currently defined. Note: the tabs update when clicked on using a mouse but values displayed cannot be edited from the Tabs.

Data and **Models**: lists the data loaded and the cases that have been run with a case id and description. Clicking on the first column of a row generates a table figure with details of the variables for the case and any associated metadata. Buttons on the figure provide access the class definition metadata, source information (files input or models used) and any user data (e.g., tables of derived parameters) that is saved with the data set.

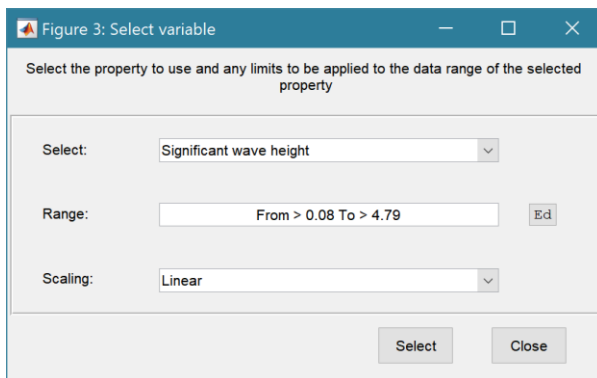
Inputs: tabulates the system properties that have been set (display only).

Q-Plot: displays a quick-plot defined for the class of the selected case (display only).

Stats: displays a table of results for any analyses that have been run (can be copied to clip board).

3.8 UI Data Selection

Functions such as Derive Output (3.4.4 and 4.16), Plotting (3.5.1) and Statistics (3.5.2) use a standardised UI for data selection. The Case, Dataset and Variable inputs allow a specific dataset to be selected from drop down lists. One each of these has been set to the desired selection the choice is assigned to a button. The button varies with application and may be X, Y, Z, or Dependent and Independent, or Reference and Sample, etc. Assigning to the button enables further sub-sampling to be defined if required. Where an application requires a specific number of dimensions (e.g., a 2D plot), then selections that are not already vectors will need to be subsampled. At the same time, the range of a selected variable can be adjusted so that a contiguous window within the full record can be extracted. In most applications, any scaling that can be applied to the variable (e.g., linear, log, relative, scaled, normalised, differences) is also selected on this UI. The selection is defined in two steps:



Step 1.

Select the attribute to use. This can be the variable or any of its associated dimensions, which are listed in the drop-down list.

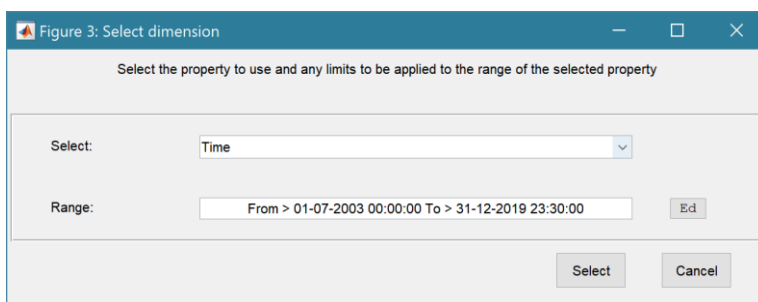
The range for the selection can be adjusted by editing the text box or using the Edit (Ed) button.

Any scaling to be applied is selected from the drop-down list.⁺

Press Select to go to the next step or Close to quit.

⁺ scaling options include Linear; Log; Relative (V-V(x=0)); Scaled (V/V(x=0)); Normalised; Normalised (=ve); Differences; Rolling mean.

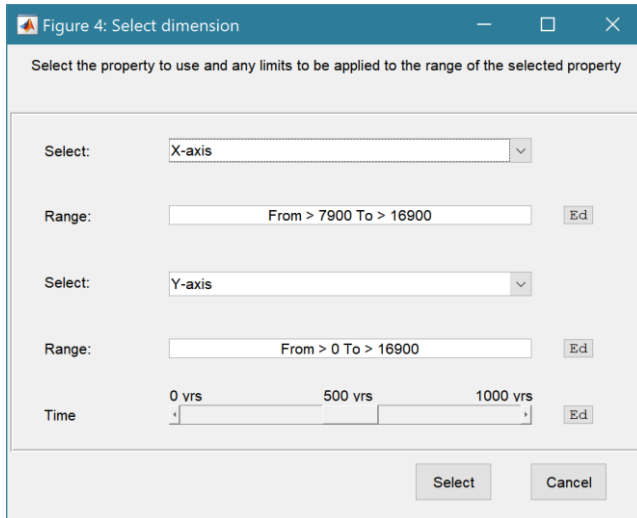
The number of variables listed on the UI depends on the dimensions of the selected variable. For each one Select the attribute to use and the range to be applied.



Step 2 - Variable only has dimension of time.

No selection to be made.

Edit range if required.



Step 2 - Variable has 3 dimensions but only 2 are needed for the intended use.

Select the 1st variable to use as a dimension.

Edit range if required.

Select the 2nd variable to use as a dimension.

Edit range if required.

Use the slider or the Edit (Ed) button to set the value of the dimension to use. (A value of t=500 is selected in the example shown).

Press Select to accept the selection made.

[NB: Only unused dimensions can be selected

from the Select drop-down lists. To adjust from the default list this can sometimes require that the second Select list-box is set first to allow the first Select list-box to be set to the desired value.]

The resulting selection is then detailed in full (including the ranges or values to be applied to all dimensions) in the text box alongside the button being defined.

Note where a variable is being selected as one property and a dimension as a second property, any sub-selection of range must be consistent in the two selections. This is done to allow variables and dimensions to be used as flexibly as possible.

3.9 Accessing data from the Command Window

In addition to the options to save or export data provided by the *Project>Cases>Save* and *Project>Import/Export* options, data can also be accessed directly for use in Matlab™, or to copy to other software packages. This requires use of the Command Window in Matlab™, and a handle to the App being used. To get a handle, open the App from the Command Window as follows:

```
>> myapp = <AppName>;           e.g., >> as = Asmita;
```

Simply typing:

```
>> myapp
```

Which displays the results shown in the left column below with an explanation of each data type in the right hand column.

myapp = <AppName> with properties:	Purpose
Inputs: [1×1 struct]	A struct with field names that match all the model parameter input fields currently
Cases: [1×1 muiCatalogue]	muiCatalogue class with properties DataSets and Catalogue. The former holds the data the latter the details of the currently held records.
Info: [1×1 muiProject]	muiProject class with current project information such as file and path name.
Constants: [1×1 muiConstants]	muiConstants class with generic model properties (e.g. gravity, etc).



To access current model settings, use the following:

```
>> myapp.Inputs.<InputClassName>
```

To access the listing of current data sets, use:

```
>> myapp.Cases.Catalogue
```

To access imported or model data sets, use:

```
>> myapp.Cases.DataSets.<DataClassName>
```

If there are more than one instance of the model output, it is necessary to specify an index. This then provides access to all the properties held by that data set. Two of these may be of particular interest, `RunParam` and `Data`. The former holds the input parameters used for that specific model run.

`RunParam` is a struct with fields that are the class names required to run the model (similar to `Inputs` above). The `Data` property is a model specific struct with field names defined in the code for the model class. If there is only a single table assigned this will be given the field name of 'Dataset'. To access the *dstable* created by the model, use:

```
>> myapp.Cases.DataSets.<DataClassName>(idx).Data.Dataset
```

```
>> myapp.Cases.DataSets.<DataClassName>(idx).Data.<ModelSpecificName>
```

To access the underlying *table*, use:

```
>> myapp.Cases.DataSets.<DataClassName>(idx).Data.Dataset.DataTable
```

The result can be assigned to new variables as required. Note that when assigning *dtables* it may be necessary to explicitly use the copy command to avoid creating a handle to the existing instance and potentially corrupting the existing data.

4 Supporting Information and Model Setup

This section provides some background on the methods used in the quality control of data and the models included within CoastalTools.

4.1 Quality Checks

Once the Data QC option has been run for a data set, 'qc' is displayed in the QC column on the right-hand side of the Data tab. If the Data QC is re-run this applies the quality checks to the currently saved timeseries and overwrites the values. The following checks are currently run when Data QC is selected.

4.1.1 Waves

The CCO checks for Hs, Tz and Tp are as follows:

- Flag = 0 - all data pass
- Flag = 1 - either Hs or Tz fail, so all data fail
- Flag = 2 - Tp fail + derivatives
- Flag = 3 - Direction fail + derivatives
- Flag = 4 - Spread fail + derivatives
- Flag = 5 - Tp fail Jump test only (exclude on advice from Travis Morgan at CCO)
- Flag = 7 - Buoy adrift
- Flag = 8 - Sea Temperature fail
- Flag = 9 - Missing data (already assigned as NaN)

If any of flags 1-4 are set, the data are changed to NaN. In addition, the following checks are performed and again data that fail are set to NaN:

- ~ $0 > \text{Direction} < 360$
- ~ $\text{Hs} < \text{Hmax}$
- ~ $\text{Hs-Tz steepness} < 1/7$ (using deep water steepness)
- ~ $\text{Hmax-Tp steepness} < 1/7$ (ditto)
- ~ $\text{Tp} < \text{MaxTp}$ (defined in *Setup>Run parameters>Model constants* but the user is also prompted to confirm or adjust this value).

4.1.2 Water levels

Some BODC file formats have an error flag for bad data. This is written as a 9 when the timeseries is loaded. Running the Data QC for water levels sets data values with a 9 flag to NaNs. In addition, a time series plot is generated and the user is prompted to define minimum and maximum values. Values outside this range are assigned NaN values.

For CCO tidal data the flags used are loaded with the data. The flags are used in the Data QC tool to remove suspect data. Flags 3, 4, 8 and 9 are set to NaN when Data QC is applied.

- Flag = 1 correct value
- Flag = 2 interpolated
- Flag = 3 doubtful
- Flag = 4 isolated spike or wrong value
- Flag = 5 correct but extreme
- Flag = 6 reference change detected
- Flag = 7 constant value
- Flag = 8 out of range
- Flag = 9 missing

4.2 Nearshore waves

For each time interval the model uses the offshore values of wave height, period and direction. If water levels are included, these are interpolated to the time intervals of the wave data. Wave refraction and shoaling is computed to either the specified inshore depth, or to the edge of the surf zone. For the latter, an iterative procedure is used to calculate the depth at which the inshore wave height based on refraction, shoaling and bed friction is equal to the depth limited breaking wave height. The model outputs a time series of inshore wave heights and the inshore depth used (which is either the depth at the specified bed level, or the edge of the surf zone). Wave period is assumed to remain constant.

Method	Specified inshore point Inshore bed level = value	Edge of surf zone Inshore bed level = NaN
0 – no breaking	Refraction to inshore point	Not applicable
1 – SPM breaking on a slope	Refraction and breaking at inshore point	Refraction to breaker depth taken to be the edge of the surf zone
2 – La Roux method		
3 – Holmes/TCP method		

4.2.1 Refraction, Shoaling and Friction

Nearshore waves are calculated using linear wave theory and plane bed refraction and shoaling, as follows:

$$H_{si} = k_r k_s k_f H_{so}$$

where H_s is the significant wave height (inshore and offshore), k_r is the refraction coefficient, k_s is the shoaling coefficient, k_f is the friction coefficient. The coefficients are defined as follows:

$$k_r = \sqrt{\frac{\cos(\alpha_o)}{\cos(\alpha_i)}} \quad \text{and} \quad \alpha_i = \sin^{-1}\left(\frac{c_i}{c_o} \sin \alpha_o\right)$$

where α is the angle between wave crest and bed contour, and c is the wave celerity (both onshore (i) and offshore (o) respectively). The default method of computing refraction is to assume a plane bed and constant contour angles. This requires the depths at the offshore and inshore points and a shoreline angle. For this option set the Offshore and Shoreline contour angles to be the same, or leave the Offshore contour angle blank.

Two other options are possible.

- (i) An offshore contour angle is specified (see *Setup>Run parameters>Site parameters*) that is different to the shoreline angle. The inshore wave angle and refraction coefficient are calculated in increments from the source point to the target point. At each increment the depth and angle of the bed contour are adjusted as follows:

$$\delta\theta = \frac{|\theta_1 - \theta_0|}{nint} \quad \text{and} \quad \delta d = \frac{|d_1 - d_0|}{nint}$$

where θ is the contour angle (degTN), d is the depth of water and $nint$ is the number of intervals (=10). The number of intervals does influence the result (mainly for oblique wave angles). The value of $nint$ is hard coded at the head of the refraction.m function.

- (ii) A series of Offshore contour angles and Offshore bed levels are defined in *Setup>Run parameters>Site parameters* (e.g., Offshore bed levels of -13 -9 -6 and Offshore angles of 75 63 54). The number of bed levels and angles must be the same. The Shoreline angle and

Inshore bed level define the last (inshore) point. The inshore angle and refraction coefficient are computed by iterating through the specified values.

The shoaling coefficient is given by:

$$k_s = \sqrt{\frac{c_{go}}{c_{gi}}} \quad \text{where} \quad c_g = \frac{c}{2} \left(\frac{1 + \frac{4\pi d}{c \cdot T_p}}{\sinh\left(\frac{4\pi d}{c \cdot T_p}\right)} \right)$$

where d is the water depth, c is the wave celerity, c_g is the group celerity, T_p is the peak period.

The friction coefficient, k_f , is defined in *Setup>Run parameters>Site parameters*.

The resultant nearshore wave height, H_{si} , is then checked for wave breaking.

4.2.2 Breaking wave height

The monochromatic wave breaking height, H_b , is calculated using one of the following methods, depending on the value set in *Setup>Run parameters>Site parameters* for the Breaker Height model:

- 0: simple ratio of 0.78 (maximum for a solitary wave)
- 1: breaking on a slope using equations (2-92 to 2-94) in the Shore Protection Manual (1984)
- 2: breaking on a slope in a water depth one plunge length in front of a structure toe, using equations (2-92 to 2-94) in the Shore Protection Manual (1992)
- 3: as option 2 but modified for overtopping as detailed on Owen (1980). This is set automatically when the overtopping model is called.

The Wave Breaking Model to be used is defined in *Setup>Run parameters>Site parameters*. Option '0' does not consider wave breaking. Options 1-3 use different methods to include wave breaking height, based on one of three methods:

0. No breaking included in the model.
1. The depth limiting wave height, H_b , is estimated using equations (2-92 to 2-94) in the Shore Protection Manual (1984) based on the nearshore bed slope (see Section 4.2.3), water depth and wave period.

$$\frac{d_b}{H_b} = \frac{1}{b - \left(\frac{aH_b}{gT^2}\right)}, \quad \text{where } a = 43.75(1 - e^{19m}), \quad \text{and } b = \frac{1.56}{1 + e^{19.5m}}$$

2. The peak period is used in accordance with the suggestion of Soulsby (1997, p69).
3. The formulation of Le Roux (2007) as given in equation (25) of the paper.
4. Formulations for the wave breaking of the significant wave height as presented by Tucker et al (1983) and Holmes (pers. comm.). These provide upper and lower bounds to the change in significant wave height which are calculated as follows:

$$H_{sb} = \left(\frac{\beta^2 H_s^2 \left(1 - \exp\left(-2\left(\frac{H_b}{H_s}\right)^2\right) \left(1 + 2\left(\frac{H_b}{H_s}\right)^2 \right) \right)}{1 - \exp\left(-2\left(\frac{H_b}{H_s}\right)^2\right)} \right)^{1/2} \quad \text{a lower bound (Holmes, pers.comm.)}$$

$$H_{sb} = \beta^2 H_s^2 \left(1 - \exp\left(-2\left(\frac{H_b}{H_s}\right)^2\right) \right)^{1/2} \quad \text{an upper bound (Tucker et al., 1983)}$$

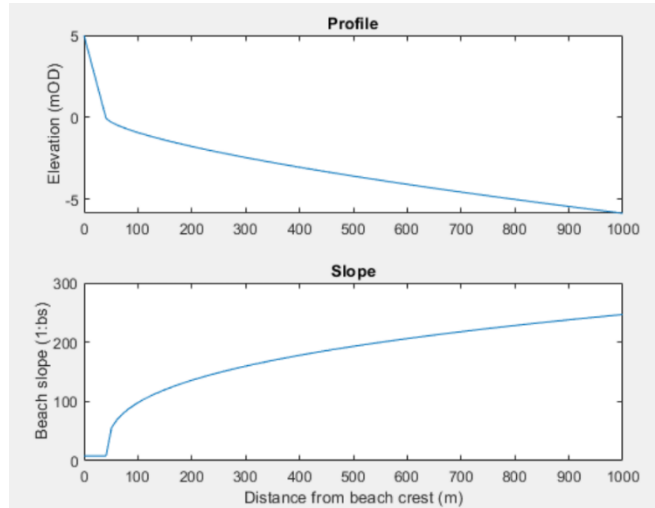
It is assumed that the lower bound applies for very flat slopes (1:100 and greater) and the upper bound for very steep shores (less than 1:20). Between these two a linear transition is applied. *It should be stressed that the basis for the limits and the transition have not been tested against real data.*

4.2.3 Shore profile

An idealised beach profile is used to define the slope as a function of depth. The profile is based on a linear slope (upper beach slope) between the beach crest level and mean tide level (SWL=0mOD) and an equilibrium profile out to deep water. This uses the Dean equation (Dean, 1977):

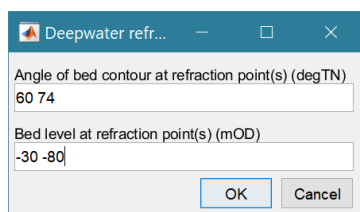
$$h = Ay^{2/3}$$

In the Dean formulation, A is a function of sediment fall velocity. Here it is derived from the depth 1km out from the mean tide level, which is input as a bed level (mOD and so a negative value). Alternatively, if a (y, z) coordinate is used to define an offshore point, this is used to determine A. The figure shows the case the profile and slope variation for a crest level of +5mOD, upper beach slope of 1:8 and a bed elevation at 1km of -6mOD.



4.3 Deepwater waves

This option treats the selected wave data record as a “nearshore” data set and applies refraction and shoaling to a deepwater depth and a user specified offshore seabed angle (specified using the same convention as the Shoreline Angle above). The three methods of calculating refraction are the same as for Inshore waves (Section 4.2.1). The input dialogue defaults to a contour angle that is the same as the Offshore contour angle defined in the main UI (*Setup>Run parameters>Site parameters*) and a bed level of -100m.



Changing the angle to a different value invokes the incremental method; option (i) in Section 4.2.1. Entering multiple angles and bed levels invokes option (ii) in Section 4.2.1. The deepwater wave angle and refraction coefficient are computed by iterating through the specified values.

4.4 Wind-waves

The Wind-wave hindcast model uses measured wind data and a set of defined fetch lengths for a given (usually nearshore) location. The latter are used to calculate the effective fetch, based on the method of Donelan (1985), or the method given in the Shore Protection Manual (USACE, 1984). Wave height and period are computed using the effective fetch lengths and the TMA spectrum (Hughes, 1984; Bouws *et al.*, 1985; 1987).

The TMA spectrum modifies the JONSWAP spectrum to take account of shallow water effects (depth saturation of the energy spectra). The JONSWAP spectrum is defined as:

$$S(\omega) = E(f) \exp \left[-1.25 \left(\frac{f_p}{f} \right)^4 \right] \gamma^\Gamma$$

where

$$\Gamma = \exp\left[-\frac{(f - f_p)^2}{2\beta^2 f_p^2}\right]; \quad E(f) = \frac{\alpha g^2}{(2\pi)^4 f^5}$$

f is the frequency, f_p is the peak frequency (frequency at which $S(f)$ is a maximum), α is the Phillips constant (sometimes called the equilibrium-range parameter), γ is the peak-enhancement factor (usually taken to be 3.3), and $\beta = 0.07$ for $f < f_p$ and $\beta = 0.09$ for $f > f_p$.

To correct for depth-dependent effects, Bouws et al. (1985) also manipulated the linear term $E(f)$ to reflect the loss of energy due to enhanced dissipation of shallow water. They replace $E(f)$ with $E_k(f, H)$, where:

$$E_k(f, H) = \frac{\alpha g^2 \varphi_k(\omega_H)}{(2\pi)^4 f^5}, \quad \varphi_k(\omega_H) = \frac{\left[(k(\omega, h))^{-3} \frac{\partial k(\omega, H)}{\partial f} \right]}{\left[(k(\omega, \infty))^{-3} \frac{\partial k(\omega, \infty)}{\partial f} \right]} \quad \text{and} \quad \omega_H = 2\pi f \sqrt{h/g}$$

The solution of the partial derivatives is detailed in Kitaigorodskii et al. (1975) and Bouws et al. give the following approximate solution:

$$\begin{aligned} \varphi_k &\approx 0.5 \cdot \omega_H^2 && \text{for } \omega_H \leq 1 \\ \varphi_k &\approx 1 - 0.5 \cdot (2 - \omega_H)^2 && \text{for } 1 < \omega_H \leq 2 \\ \varphi_k &\approx 1 && \text{for } \omega_H > 2 \end{aligned}$$

The peak frequency is estimated from the peak period using the equation proposed by Donelan (1985):

$$T_p = 0.54 \cdot \frac{U^{0.54} F^{0.23}}{g^{0.77}}$$

Where U is the wind speed at 10m (m/s), F is the fetch length (m) and g is the acceleration due to gravity (m/s^2)

Integrating the energy spectrum and the second moment of the spectrum gives the significant wave height and mean zero crossing period.

i.e
$$m_n = \int_0^{\infty} \omega^n S(\omega) d\omega \quad \text{gives} \quad H_{m_0} = 4\sqrt{m_0} \quad \text{and} \quad T_z = \sqrt{m_0/m_2}$$

See Carter (1982) for further details.

Two methods are available for calculating the effective fetch, as proposed in the Shore Protection Manual and by Donelan (1985). They both modify the fetch along the mean wave direction to account for directional spreading.

$$effF = \frac{\sum F_i < trigfn > (\beta(\varphi_i - \theta))^n}{\sum < trigfn > (\beta(\varphi_i - \theta))^n}$$

Where F_i is the fetch length in the direction of φ_i , $< trigfn >$ is a trigonometric function, θ is the wind direction, β is a constant and n is the spreading coefficient.

For the Donelan formulation, developed for wind-wave generation in lakes where the fetches can be long and narrow, the trigonometric function is hyperbolic secant $\text{sech}(z) = 1/(\cosh(z))$ and the constant is 2.28.

For the SPM method, the trigonometric function is a cosine and the constant is 1.

Typically, a spreading coefficient of $n=2$ is appropriate for locally generated wind-waves whereas an exponent of $n=10$ is more representative of developed sea states.

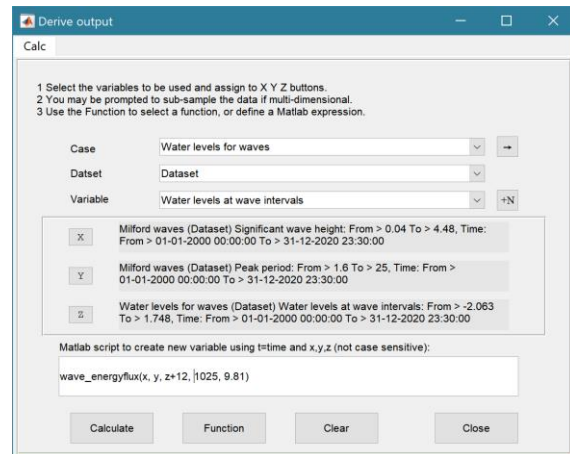
4.5 Wave power

The wave power or wave energy flux (J/ms) is calculated using linear wave theory as:

$$F = g \cdot \rho_w \cdot \frac{H_s^2}{8} \cdot c_g$$

Where ρ_w is the density of water (kg/m^3) and g is the acceleration due to gravity (m/s^2).

The option from the *Run > Wave properties > Wave Power* menu uses nearshore waves and requires the nearshore wave model to have been run (Sections 3.4.1 and 4.2). The wave power for the input wave data (e.g. buoy location) or offshore point can be calculated using the *Run > Derive Output* menu option (Section 3.4.4). The inputs required are wave height, H_s , wave period, T_p , water depth, d , density of water, and acceleration due to gravity. The water depth record needs to be at the same sampling intervals as the waves. This can be created by first subsampling the record at time intervals defined by another record (use Function button on Derive Output UI to access this option). Assign the water level time series to be used the X button and set the ‘method’ to ‘linear’. The user is prompted to select the time series to define the new time intervals and the wave data to be used should be selected. This establishes and new water level timeseries with the same number of records as the wave timeseries. Using the Derive Output UI, assign wave heights to the X button, wave periods to the Y button, and the newly created water levels to the Z button. Then use the following function call:



`wave_energyflux(x, y, z+depth at the site, water density, g)`

Note that the water levels, z , are combined with the *depth at the site* (e.g., buoy location) which should be measured relative to the same datum that is being used for the water levels. This may be relative to National/Ordnance Datum, or Chart Datum, depending on the source data.

4.6 Runup (setup & wash)

The runup is based on the formulation of Stockdon et al (2006) Their equation combines setup ($\langle \eta \rangle$), and swash made up of incident, S_{inc} and infragravity, S_{ig} components to give the 2% runup level as:

$$R_2 = 1.1 \left\{ 0.35 \beta_f (H_o L_o)^{1/2} + \frac{[H_o L_o (0.56 \beta_f^2 + 0.004)]^{1/2}}{2} \right\}$$

Where β_f is the foreshore slope, H_o is the deep water wave height and T_o is the deep water wave period. The runup elevation is the runup added to the water level time series. The algorithm takes the measured wave height and water depth at the buoy and applies the shoaling coefficient to get the deep water wave height (depth of 50m used). The resultant deep water wave height and wave length (obtained from the peak period and wave celerity based on linear wave theory) are then used in the above equation.

4.7 Littoral drift

Four formulations for calculating littoral drift rates are available. The inputs used are the inshore wave parameters and the site parameters defined in *Setup>Run parameters>Site parameters*. When the inshore depth is set to NaN this function uses an iterative procedure to find the depth limited wave conditions at the edge of surf zone. Alternatively, the inshore depth can be specified, see Section 4.2.

The available options for the littoral drift formula are:

- 1) the US Army Corps CERC formula ((1984). The drift coefficient can be defined in *Setup>Run parameters>Site parameters* (and is only applied to this formulation).
- 2) the SANDS formula, this is essentially the CERC with adjustments for wave breaking and the group celerity (Soulsby, 1997, p194).
- 3) the Kamphuis formula, to take account of grain size and beach slope (Kamphuis, 1991)
- 4) the Damgaard and Soulsby formula for shingle (Soulsby, 1997, p195).

The convention used for drift volumes is that positive values are from left to right when looking at the shore from the sea.

See also the function for the calculation of littoral drift volumes using littoral drift stats in *Run>Derive Output* (Section 4.16).

4.8 X-shore Transport

Cross-shore transport is estimated using the method proposed by Bailard and Inman (Bailard and Inman, 1981). They derive an equation for the cross-shore transport rate, Q_c , comprising bedload and suspended load components, of the form:

$$Q_c = Q_b + Q_s = \rho C_d u_m^3 \left\{ \frac{\varepsilon_b}{\tan \phi} \left(\psi_1 + \frac{2}{3} \delta_u - \frac{\tan \beta}{\tan \phi} u_3^* \right) + \frac{u_m}{w_s} \varepsilon_s \left(\psi_2 + \delta_u u_3^* - \frac{u_m}{w_s} \varepsilon_s u_5^* \tan \beta \right) \right\}$$

where $\varepsilon_b=0.21$, $\varepsilon_s=0.025$, C_d is the drag coefficient (which Bailard takes as ~ 0.005), w_s is the sediment fall velocity (m/s), ϕ is the angle of repose, β is the beach slope, u_m is the near bed velocity (m/s), ρ is the water density (kg/m^3), ρ_s is the sediment density (kg/m^3) and δ_u , ψ_1 , ψ_2 , u_m , and u_3^* , u_5^* are cross-shore velocities (positive onshore). These cross-shore velocities are presented in (Alexandrakis and Poulos, 2014), based on (Bailard, 1981; 1982):

$$\psi_1 = 0.00303 - 0.00144 H_s$$

$$\psi_2 = 0.00603 - 0.00510 H_s$$

$$\delta_u = 0.0 - 0.00157 H_s$$

$$u_m = 0.319 + 0.403 H_s$$

$$u_3^* = 0.00548 + 0.000733 H_s$$

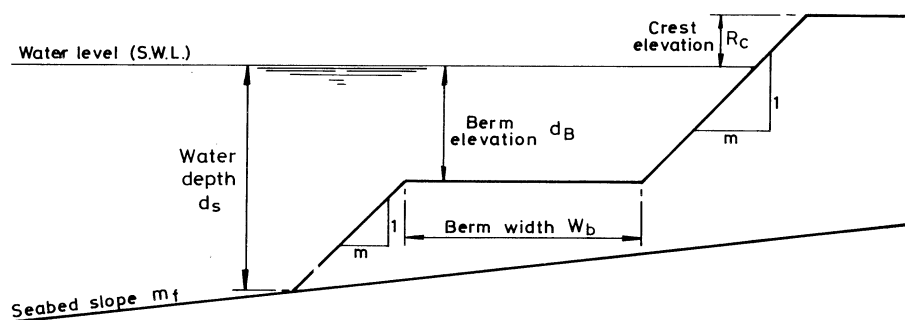
$$u_5^* = 0.015 + 0.00346 H_s$$

in which H_s is the significant wave height. Note the original paper presented the equation in units of cm/s and the above equation are in units of m/s. The equation for δ_u has also been modified to correctly represent the data presented in the source paper. The cross-shore transport rate has units of m^3/s and is positive in the onshore direction. For an example of application see Masselink *et al* (2008).

4.9 Overtopping

The volume of water overtopping a sea wall, or beach crest can be estimated using the overtopping formula proposed by Owen (1980). This can take account of upper and lower slopes, a berm and the wall or beach ‘roughness’ (essentially a friction or dissipation coefficient). The definition sketch, for the variables used, is shown below. Minimum definition requires a crest and toe level, relative to the datum used in the water level data set, and the lower wall slope (input as m for a slope of 1:m). Specifying a berm level with zero berm width allows a compound slope with different upper and lower wall slopes. Strictly this model is limited to slopes in the range 1:1 to 1:5 and berm widths from 0-80m at depths of 0 to 4m below swl (see Owen, 1980 for details). When the berm is above the swl a compound slope is used.

Figure 4 – Seawall definition sketch (Owen, 1980)



4.10 Iribarren number and beach type

The Iribarren wave number is calculated as follows:

$$Ir = \frac{\tan(\alpha)}{\sqrt{H/L_o}}$$

Where α is the beach slope, H is the wave height and L_o is the deep-water wave length. The function is passed the nearshore wave conditions (see Section 4.2) and calculates the wave length from $gT_p^2/2\pi$, where T_p is the peak wave period. The function also returns Breaker Type based on Ir where (SPM, USACE, 1984):

- $Ir > 2$: Breaker Type = 3; surging
- $Ir \leq 2$ & $Ir > 0.4$: Breaker Type = 2; plunging
- $Ir \leq 0.4$: Breaker Type = 1; spilling

4.11 Tides

Measured wave data can be used to derive the tidal constituents, or to create a tide only water level record for a chosen period of time using *Run>Tidal analysis>Analysis* and *Run>Tidal analysis>Reconstruction* (see Section 3.4.3). These make use of the functions from U-Tide, developed by Daniel Codiga (2011). In this implementation the function `ut_solv.m` is used to derive constituents from water level time series and the function `ut_reconst.m` is used to generate tide level time series from a set of constituents. When analysing a water level time series the predicted time series can be saved along with the constituents (saved to the class property `Constituents`). To recreate records the user can select from the analysed constituents available. [Note: there is currently no provision to input constituents manually]. The paper by Codiga (2011) provides a detailed explanation of the methods being used and a range of options that have not been implemented within

CoastalTools. The implementation provided is sufficient for extracting surges from water level data and to patch gaps in water level records, with the tidal record, for use in models such as the Nearshore Wave model. The calls used are as follows:

Input variables: `lat = []`; `cnsit = 'auto'`; `options = {'NodsatNone','White','NoDiagn'}`;

Function call: `coef = ut_solv(time,data,[],lat,cnsit,options{:})`;

Input variables: `options = {}`; Function call: `predicted = ut_reconstr(time,coef,options{:})`;

4.12 Beach type

The Beach Type is determined from dimensionless fall velocity (Ω) for the grain size defined in Site Parameters and the selected wave time series as:

$$\Omega = \frac{H_s}{w_s T_p}$$

Where H_s is the significant wave height, T_p the peak period and w_s is the settling velocity obtained using eqn (102) in ‘Dynamics of marine sands’, (Soulsby, 1997). The dimensionless fall velocity can be used to identify beach type based on the following categories (Wright and Short, 1984):

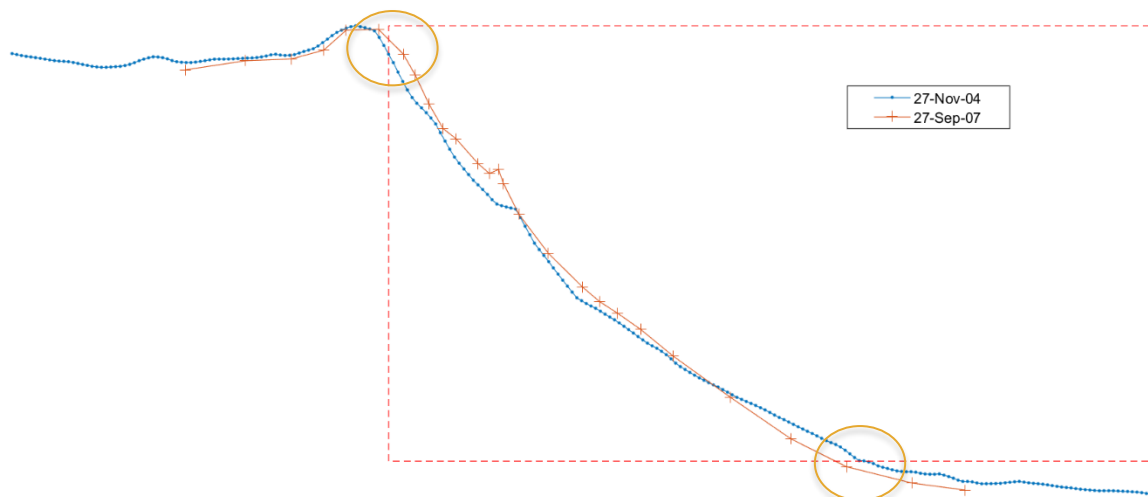
- $\Omega \geq 6$: dissipative
- $\Omega < 6$ & $\Omega > 1$: intermediate
- $\Omega \leq 1$: reflective

4.13 Volumes and shoreline position

When calculating volumes (cross-sectional area per metre run of each), a defining rectangle is used as the control area (see Section 3.4.2). Within this control area the following parameters are calculated:

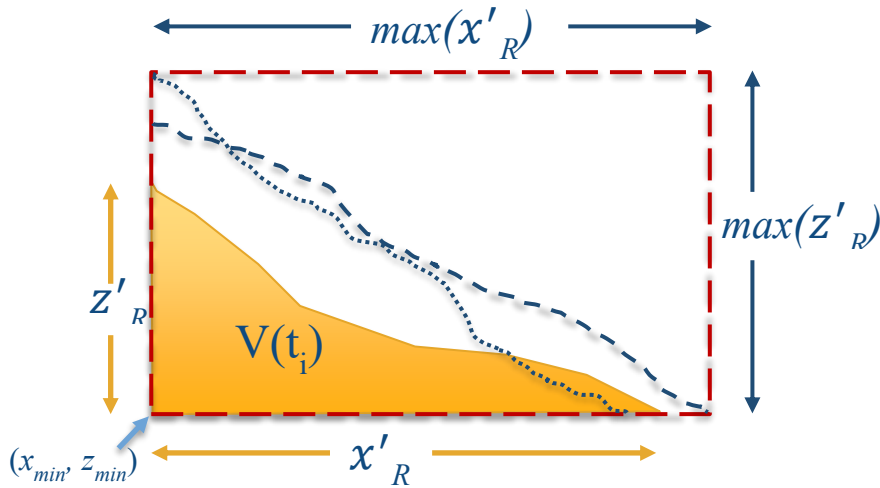
- Volume: $V = \int z' dx'$
- Horizontal centroid: $X_1 = \int x' z' dx' / V$ (first moment in x)
- Vertical centroid: $Z_1 = \int x' z' dz' / V$ (first moment in z)
- Second moment distance: $X_2 = (\int x'^2 z' dx' / V)^{1/2}$ (legacy – no longer output)

Where x and z are the raw profile data. x' is then the horizontal distance from the origin of the control area and z' is the elevation above the same origin (as defined by x_{min} and z_{min}). For each profile, the points on the profile nearest the defined minimum x or minimum z are found and used to define the two end points for that profile.



These values are then subtracted from the profile x-z co-ordinates, such that

$$x' = x - x_{min}; \quad x' \geq 0 \quad \text{and} \quad z' = z - z_{min}; \quad z' \geq 0$$



The range of x and z is then defined as: $L_x = \max(x')$ and $L_z = \max(z')$

These are used to define non-dimensional variables $x'' = x'/L_x$ and $z'' = z'/L_z$. The non-dimensional moments are then:

Non-dimensional volume: $m_0 = \int z'' dx''$

Non-dimensional x centroid: $x_1 = \int x'' z'' dx'' / m_0$

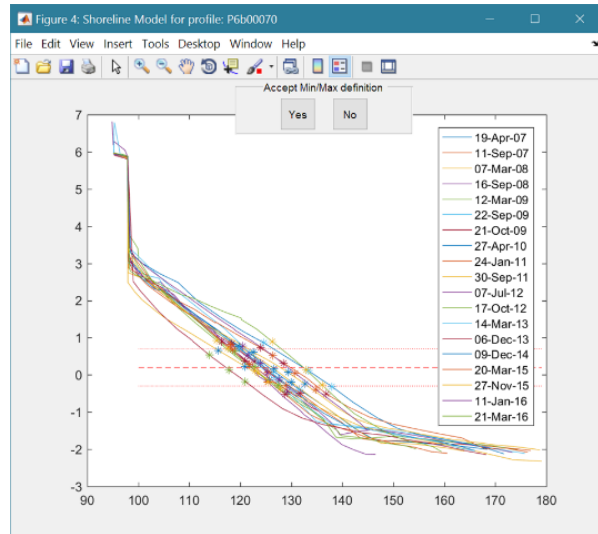
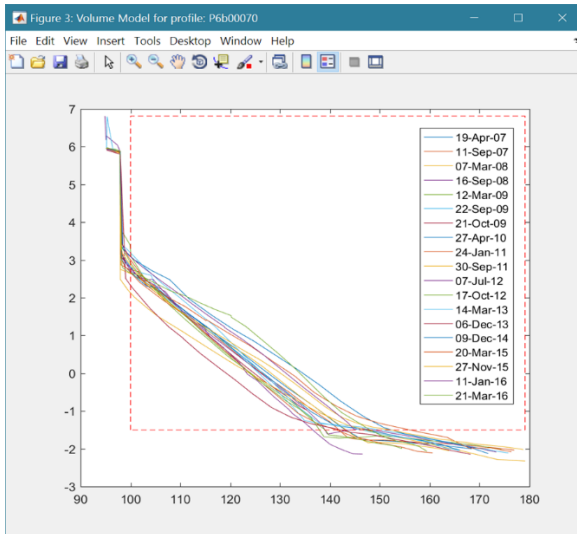
Non-dimensional z centroid: $z_1 = \int x'' z'' dz'' / m_0$

Non-dimensional second moment arm: $x_2 = (\int x''^2 z'' dx'' / m_0)^{1/2}$ (legacy – no longer output)

The beach slope (1:m) for each profile is defined as L_x/L_z .

The values saved as time series are V, m_0, x_1, z_1 and beach slope. The values reported to the *Calcs/Volumes* tab are V, m_0, L_x, L_z, X_1 and Z_1 .

The Shoreline position follows a similar procedure and finds the points on each profile nearest the defined elevation and the upper and lower bounds. The shoreline distance is the interpolated x-distance to the defined elevation and the slope is the mean gradient of the profile between the upper and lower bounds. The plots show volume (left) shoreline (right) selection plots:



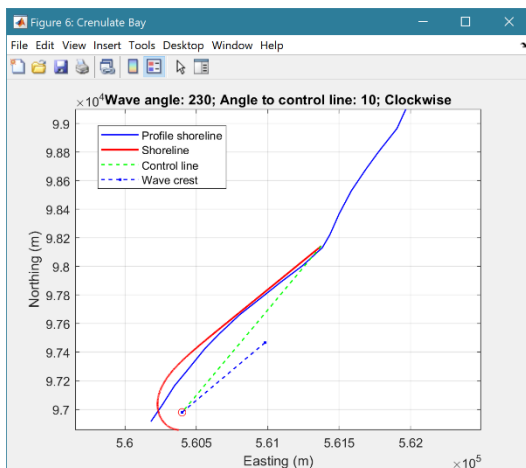
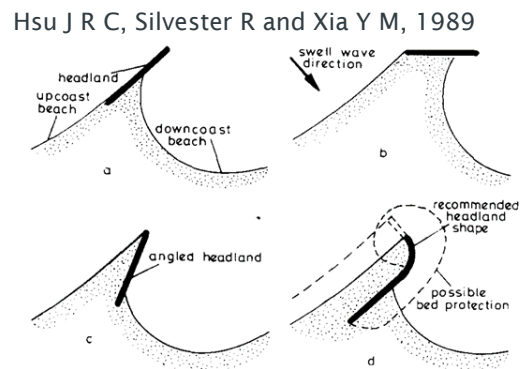
The *Setup>Shoreline>Load data*: provides the option to load a time series of positions (distance values from a back-beach datum measured at some elevation on the beach). This provides an equivalent data set to the shoreline position calculated from the beach profiles as outlined above.

4.14 Shoreline Change

The option to compute the change in a shoreline uses the method for finding the shoreline position that is used in the profile analysis tools, see Section 4.13, based on the user defined shoreline elevation.

4.15 Crenulate Bay

Crenulate bays (also known as logspiral or zeta) were first reported by (Yasso, 1965). This idea was developed into a simple model by Silvester who wrote numerous papers and eventually a book (Silvester, 1985; Hsu *et al.*, 1989; Silvester and Hsu, 1997; Hsu *et al.*, 2010). The option in CoastalTools implements the method proposed in (Hsu and Evans, 1989) and is run as one of the Beach analysis tools (see Section 3.4.2).



By defining a control point, and end point for the control line, along with the wave crest angle and the angle the wave crest makes with the control line, the bay formation is plotted against the selected shoreline. Where the profiles provide for multiple time steps from which to construct the shoreline at a user defined elevation, the mean shoreline position is used (solid blue line in the plot). The Bay can expand from the control point in a clockwise or anticlockwise manner. The combination of control line definition, wave angle and direction of rotation should be sufficient to obtain a fit to most shorelines where the crenulate bay concept is applicable.

4.16 Derive Output

The *Run > Derive Output* option allows the user to make use of the data held within App to derive other outputs or, pass selected data to an external function (see Section 3.4.4). The equation box can accept t , x , y , z in upper or lower case. Time can be assigned to X, Y, or Z buttons, or simply included in the equation as t (as long as the data being used in one of the variables includes a time dimension). Each data set is sampled for the defined data range. If the data set being sampled includes NaNs, the default is for these to be included (button to right of Variable is set to '+N'). To exclude NaNs press the button so that it displays '-N'. The selection is based on the variable limits defined whenever a variable is assigned to X, Y or Z using the X, Y, Z buttons.

The equation string entered in the UI is used to construct an anonymous function as follows:

```
heq = str2func(['@(t,x,y,z,mobj) ',inp.eqn]); %handle to anonymous function
[varout{:}] = heq(t,x,y,z,mobj);
```

or when using dstables:

```
heq = str2func(['@(dst,mobj) ',inp.eqn]); %handle to anonymous function
[varout{:}] = heq(dst,mobj);
```

This function is then evaluated with the defined variables for t , x , y , and z and optionally $mobj$, where $mobj$ passes the handle for the main UI to the function. Some functions may alter the length of the input variables (x , y , z , t), or return more than one variable. In addition, the variables selected can be sub-sampled when each variable is assigned to the X, Y, or Z buttons. The dimensions of the vector or array with these adjustments applied need to be dimensionally correct for the function being called. This may influence how the output can be saved (see Section 4.16.2).

If the function returns a single valued answer, this is displayed in a message box, otherwise it is saved, either by adding to an existing dataset, or creating a new one (see Section 4.16.2 and 3.4.4).

NB1: functions are forced to lower case (to be consistent with all Matlab functions), so any external user defined function call must be named in lower case.

Equations can use functions such as $\text{diff}(x)$ - difference between adjacent values - but the result is $n-1$ in length and may need to be padded, if it is to be added to an existing data set. This can be done by adding a NaN at the beginning or the end:

e.g.: `[NaN;diff(x)]`

NB: the separator needs to be a semi-colon to ensure the correct vector concatenation. Putting the NaN before the equation means that the difference over the first interval is assigned to a record at the end of the interval. If the NaN is put after the function, then the assignment would be to the records at the start of each interval.

Another useful built-in function allows arrays to be sub-sampled. This requires the array, z , to be multiplied by an array of the same size. By including the dimensions in a unitary matrix, the range of each variable can be defined. For a 2D array that varies in time one way of doing this is:

```
>> [z.*repmat(1, length(t), length(x), length(y))]
```

NB2: the order of the dimensions t , x , y must match the dimensions of the array, z .

NB3: When using Matlab compound expressions, such as the above sub-sampling expression, the expression must be enclosed in square brackets to distinguish it from a function call.

Adding the comment `%time` or `%rows`, allows the the row dimension to be added to the new dataset. For example if `x` and `y` data sets are timeseries, then a Matlab™ expresion, or function call, can be used to create a new time series as follows:

```
x^2+y %time
```

4.16.1 Calling an external function

The Derive Output UI can also be used as an interface to user functions that are available on the Matlab search path. Simply type the function call with the appropriate variable assignment and the new variable is created. (NB: the UI adopts the Matlab convention that all functions are lower case). Some examples of functions provided in CoastalTools are detailed in Section 4.16.3 **Error! Reference source not found.**

The input variables for the function must match the syntax used for the call from the Derive Output UI, as explained above. In addition, functions can return a single value, one or more vectors or arrays, or a dstable (see Section 4.16.2). If the variables have a dimension (e.g., *time*) then this should be the first variable, with other variables following. If there is a need to handle additional dimensions then use the option to return a dstable.

If there is no output to be passed back, the function should return a variable containing the string 'no output' to suppress the message box, which is used for single value outputs (numerical or text).

An alternative when calling external functions is to pass the selected variables as dstables, thereby also passing all the associated metadata and RowNames for each dataset selected. For this option up to 3 variables can be selected and assigned to the X, Y, Z buttons but they are defined in the call using *dst*, for example:

```
[time,varout] = myfunction(dst, 'usertext', mobj);
```

```
dst = myfunction(dst, 'usertext', mobj);
```

where *'usertext'* and *mobj* are call strings and a handle to the model, respectively.

This passes the selected variables as a struct array of dstables to the function. Using this syntax, the function can return a dstable or struct of dstables, or as variables, containing one or more data sets.

4.16.2 Input and output format for external functions

There are several possible use cases:

4.16.2.1 Null return

When using a function that generates a table, plots a figure, or some other stand alone operation, where the function does not return data to the main UI, the function should have a single output variable. The output variable can be assigned a text string, or 'no output', if no user message is required, e.g.:

```
function res = phaseplot(x,y,t,labels)
...
res = {'Plot completed'}; %or res = {'no output'}; for silent mode
...
end
```


4.16.2.2 Single value output

For a function that may in some instances return a single value this should be the first variable being returned and can be numeric or text, e.g.:

```
function [qtime,qdrift] = littoraldriftstats(qs,tdt,varargin)
...
%Case 1 - return time and drift
qdtype = array1;
qdrift = array2;
%Case 2 - return summary value
qtime = mean(array2); %return single value
%Case 3 - return summary text
qtime = sprintf('Mean drift = %.1f',mean(array2)); %return test string
...
end
```

4.16.2.3 Using variables

If only one variable is returned (length>1), or the first variable is empty and is followed by one or more variables, the user is prompted add the variables to:

- i) Input Cases – one of the datasets used in the function call;
- ii) New Case – use output to define a new dataset;
- iii) Existing Case – add the output to an existing dataset (data sets for the selected existing case and the data being added must have the same number of rows.

In each case the user is prompted to define the properties for each of the variables.

Note that variable names and descriptions must be unique within any one dataset.

```
function y = moving(x,m,fun)
%a single variable is returned with no rows
y is a vector or array
...
end
```

or

```
function [x,y,z] = afunction(x,m,fun)
%multiple variables returned but the first variable is empty
x = [];
y and z are a vectors or arrays
...
end
```

When the first variable defines the rows of a table and subsequent variables the table entries, all variables must be the same length for the first dimension. This is treated as a new Case and the user is prompted to define the properties for each of the variables.

```
function [trange,range,hwl,lwl] = tidalrange(wl,t,issave,isplot)
%first variable is row dimension followed by additional variables
trange,range,hwl,lwl are vectors or arrays
...
end
```

4.16.2.4 Using dstables

When the output has multiple variables of a defined type it can be more convenient to define the dsproperties within the function and return the data in a dstable. This avoids the need for the user to manually input the meta-data properties. In addition, if the function generates multiple dstables, these can be returned as a struct, where the struct fieldnames define the Dataset name.

```
function dst = tidalrange(wl,t,issave,isplot)
    %dst is a dstable with variables, dimensions and dsproprties assigned
    %as required, or a struct of dstables with the struct fieldnames defining
    %each Dataset.
    dst = ...
    ...
end
```

Similarly, if the input is also using dstables, the syntax is as follows:

```
function dst_out = myfunction3(dst_in,'usertext',mobj)
    %dst_in is one or more input dstables, 'usertext' is some additional
    %instruction to the function and mobj is a handle to the model
    %allowing access to other datasets. dst_out is either a dstable, or a
    %struct of dstables with the struct fieldnames defining each Dataset.
    dst = ...
    ...
end
```

4.16.2.5 Saving additional model parameters

When saving function results as dstable, it is also possible to save additional parameters as part of the table. The following example puts a table of summary statistics in the dstable UserData property.

```
function dst = tidalrange(wl,t,issave,isplot)
    ....
    dsp = setDSproperties();
    results = {R,hwl,lwl};
    dst = dstable(results{:}, 'RowNames',rt, 'DSproperties',dsp);
    %Put fit parameters in UserData
    dst.UserData = summary_stats_table;
    ...
end
```

These results are then accessible using the user data button, when accessing the data set properties from the Data or Model tabs (see Section 3.7).

Adding functions to the Function library

To simplify accessing and using a range of functions that are commonly used in an application, the function syntax can be predefined in the file functionlibrarylist.m which can be found in the utils folder of the muitoolbox. This defines a struct for library entries that contain:

- fname - cell array of function call syntax
- fvars - cell array describing the input variables for each function
- fdesc - cell array with a short description of each function

New functions can be added by simply editing the struct in `functionlibrarylist.m`, noting that the cell array of each field in the struct must contain an entry for the function being added. In addition a sub-selection of the list can be associated with a given App based on the class name of the main UI. To amend the selection included with an App or to add a selection for a new App edit the `'switch classname'` statement towards the end of the function.

The Function button on the Derive Output UI is used to access the list, select a function and add the syntax to the function input box, where it can be edited to suit the variable assignment to the XYZ buttons.

4.16.3 Pre-defined functions

The following examples are provided within CoastalTools, where the entry in the UI text box is given in Courier font and X, Y, Z, refer to the button assignments. Some useful examples primarily for timeseries data include:

1. **Moving Average.** There are several moving average functions available from the Matlab Exchange Forum, such as `moving.m`. The call to this function is:

`moving(X, n, 'func')`, where `x` is the variable to be used, `n` specifies the number of points to average over and `'func'` is the statistical function to use (e.g. mean, std, etc). If omitted the *mean* is used. Add `%time` to the call, to include time in the output dataset.

2. **Moving average (or similar) of timeseries**, over defined duration, advancing at defined interval
`movingtime(x, t, tdur, tstep, 'func')`, where `x` is the variable to be used and `t` the associated datetimes (defined by variable selection), `tdur` is the duration over which to apply the statistic, `tstep` is the interval to advance the start time for the averaging period and `'func'` is the statistical function to use (e.g. mean, std, etc). If omitted the *mean* is used. `tdur` and `tstep` are both duration character strings of form `'2.5 d'`. Any of the following duration intervals can be used: `y`, `d`, `h`, `m`, or `s`. Returns a time series based on the defined `tstep`, where the time used is for the beginning of each stepping interval, i.e. every `tstep` from the start of the record to the nearest interval that is less than `tdur` from the end of the record.
3. **Down-sampling a time series.** This allows a timeseries to be resampled at a different interval (that must be less than the source timeseries). The call to this function is:

`downsample(x, t, 'period', 'method')`, where `x` is the variable to be resampled, `time` is the associated time for that variable, `period` can be `'year'`, `'month'`, `'day'`, `'hour'`, `'minute'`, `'second'`, and `method` can be any valid function call such as `'mean'`, `'std'`, etc. The `'period'` is required but the `'method'` is optional and if omitted the mean is used.

For timeseries with gaps the `'nanmean'` function is particularly useful but requires the Statistics toolbox.

4. **Interpolate and add noise.** To infill a record with additional points and, if required, add some random noise to the interpolated values. This is called using:
`interpwithnoise(x, t, npad, scale, method, ispos)`, where `X` is the variable, `t` is time, `npad` is the number of points to add between the existing data points, `scale` determines the magnitude of the random noise (a value of 0 results in an interpolated record with no noise), `method` is the Matlab algorithm used for the interpolation (the default is linear) and `ispos` is a true/false flag which sets negative values to zero if true.
5. **Subsample one record at the time intervals of another record** (e.g. subsample water levels to be at the same intervals as the wave data). Function is:

`subsample_ts(X, t, mobj)`, where X and t are the variable to be subsampled and $mobj$ is the UI handle (must be $mobj$). The user is prompted to select the dataset to be used to define the time intervals. A time series is returned and added as a Derived data set. The user is prompted to define the metadata for the new data set.

6. **Subsample one record based on a threshold defined for another record** (e.g. subsample waves based on a threshold water level). Function is:

`subsample(X, t, thr, mobj)`, where X and t are the variable to be subsampled, thr is the threshold value and $mobj$ is the UI handle (must be $mobj$). The user is prompted to select the dataset and variable to be used to define the condition and a condition operator (\leq , $==$, etc). A time series is returned and added as a Derived data set. The user is prompted to define the metadata for the new data set.

7. **Phase plot**. This function is similar to the recursive plot function but generates a plot based on two variables that can, optionally, be functions of time. The call to this function is:

`phaseplot(X, Y, t)`, where X and Y are the variables assigned to the respective buttons and t is time (this does not need to be assigned to a button and t can be omitted if a time stamp for the datapoints is not required).

8. **Recursive plot**. Generates a plot of a variable plotted against itself with an offset (e.g. $x(i)$ versus $x(i+1)$). This is called from the Derive Output GUI using:

`recursive_plot(x, 'varname', nint)`, where x is the variable, 'varname' is a text string in single quotes and $nint$ is an integer value that defines the size of the offset.

9. **Add sea level rise to tidal water levels** (ie typically predictions rather than measured water levels). Based on exponential growth from 1900 and zeroed to a defined year using:

`addslrtotides(X, t, delta, exprate, pivotyear)`, where X and t relate to the water level variable to be adjusted, $delta$ is a rate for the year 1900 (e.g. 0.001 m/yr), $exprate$ is the rate of exponential growth (e.g. 0.011 for a fit to observations to-date) $pivotyear$ is the year to use for zero sea level rise (e.g. 1900 adds slr based on change since 1900, whereas 2000 assumes that the tidal predictions are correct to the datum for the year 2000 and adjusts the record based on the slr function relative to that year).

10. **Tidal range time series from a water level series**. The call to the function is:

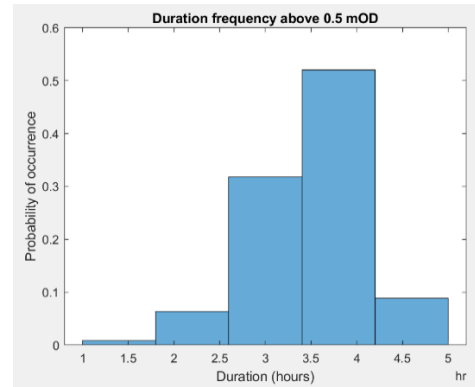
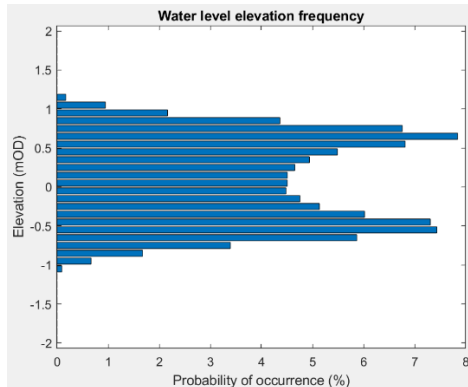
`tidalrange(X, t, issave, isplot)` where X is the water level and t is the times of the water level values. Assumes that there are multiple water level values per tide. Also, outputs mean water level and tidal range values as a table. $issave = true$ to save the results as a dataset and $isplot = true$ to generate a plot of the results.

11. **Fit trends and cycles to tidal range or high/low water time series**: The call to the function is:

`tidalrange_nltc(X, t, issave, isplot, 'titletxt')` where X is the water level and t is the times of the water level values. Assumes that there are multiple water level values per tide. When $issave = true$ the fit parameters of the selected are also saved. If $isplot = true$ the fit plots are generated by the function in addition to the summary table of fit parameters. The 'titletxt' variable is optional and is used to define the output label used for the dataset.

12. **Selection of plots for water level frequency and duration** using the function:

`waterlevelfreqplots (X, t)` where X is the variable and t is time. Plot options include Water level elevation frequency, Water level spectrum, Elevations above a threshold, Duration of threshold exceedance, Elevation frequency above threshold. Designed to analyse water levels but could easily be adapted for other variables.



13. Selection of frequency analysis plots of timeseries data using the function:

`frequencyanalysis(X, t, 'vardesc')` where *X* is the variable, *t* is time and *vardesc* is the description of the variable to be used in the plots (optional – defaults to ‘Variable’). Plot options include Time series plot of variable, Time series plot of variable above threshold, 'Plot variable frequency, Plot variable frequency above threshold, Spectral analysis plot, Duration of threshold exceedance, Rolling mean duration above a threshold.

14. Depth dependent wave steepness using the function:

`wave_steepness(X, Y, Z, t)` where *X* is the wave height, *Y* the wave period and *Z* the water depth, with *t* passing the time associated with the timeseries variables. Note that the water depth data should be a times series of the same length and at the same time intervals as the wave data, or specified as a single constant value (i.e.: `wavesteepness(X, Y, 3.5, t)`).

15. Wave height-period scatter plots using the function:

`wave_scatter(dst)` where *dst* invokes the option to pass the data as dstable. This requires that the wave height, wave period and water depth time series to be used are assigned to the *x*, *y* and *z* buttons respectively.

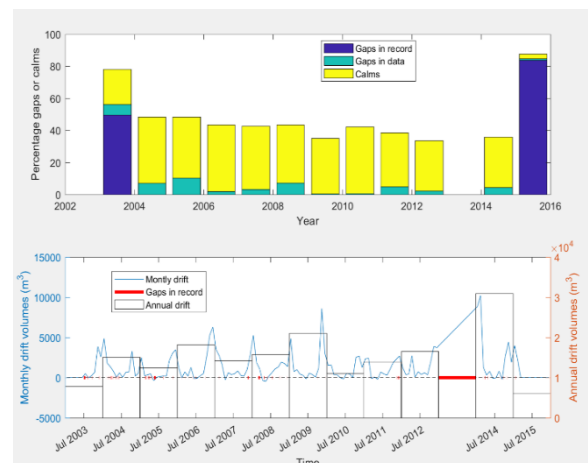
16. Ratio of alongshore to cross-shore transport using the function:

`beachtransportratio(x, theta)` where *X* is the wave direction and *theta* is the beach or shoreline angle (both in degTN).

17. Littoral drift statistics. Plots the annual and monthly volumes of drift along with details of gaps and calms. The call to this function is:

`littoraldriftstats(X, t, 'period')`, where *X* is the rate of drift, time is the associated time for that variable and *period* can be ‘year’, or ‘month’.

If no *period* is specified, the default is month. The *period* selection does not alter the plot (which shows both) but if the results are saved as a timeseries, *period* determines the timeseries interval. In the lower plot, the diamonds denote the start and end of the timeseries.



18. **Ratio of alongshore to cross-shore transport.** The CERC formula for littoral transport is based on the energy flux (P) in the direction of wave advance per unit length of beach. ie: $F = P \cdot \cos(\alpha)$, where α is the angle between wave crest and bed contour. The longshore component of energy flux is $P \cdot \cos(\alpha) \cdot \sin(\alpha)$, which leads to the main terms in the CERC formula. It follows that the cross-shore component is $P \cdot \cos^2(\alpha)$. The ratio of longshore/cross-shore energy flux (or transport potential) = $\tan(\alpha)$. The call to the function is:

`beachtransportratio(X, theta)` where X is a timeseries of inshore wave directions and 'theta' is the angle of the shoreline to True North.

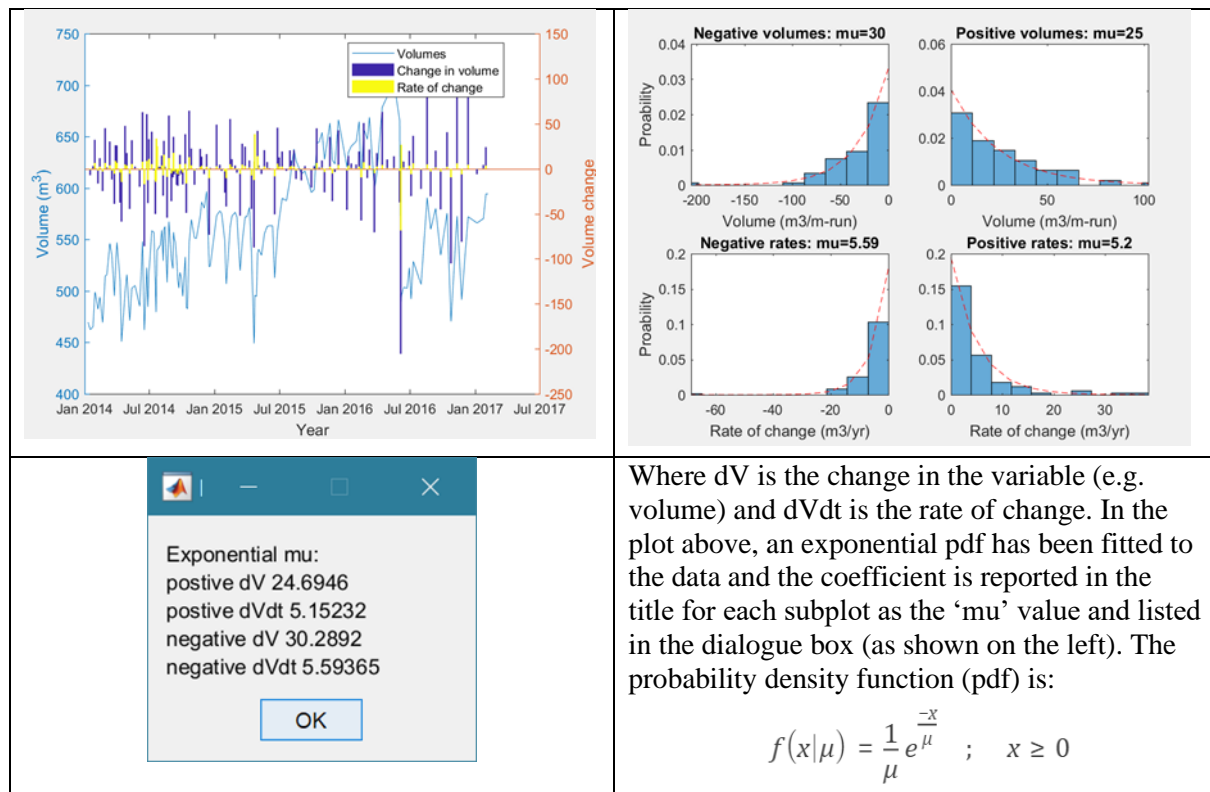
19. **Examining the rate of beach profile change (accretion and erosion).** The function computes the change in the variable over each time interval and the rate of change (assuming a linear rate of change between surveys), subdivides the population into positive and negative change values (typically this represents accretion and erosion when using volumes, or shoreline position, data) and presents the histogram and exponential fit for each data set.

Warning: the results are entirely dependent on the adequacy of the source data to represent change (e.g. volumes or shoreline position). If the survey frequency is not regular the results are unlikely to be reliable.

Select a variable X. The variable can be any metric such as beach volume or shoreline position. Use the function:

`posneg_v_stats(x, t, 'VariableName')`, where x is a time series variable, such as beach volume of shoreline position, with associated time, t (defined by variable selection). The *VariableName* is optional but if used should be between single quote marks. The *VariableName* is used to label the plot axes.

The output comprises the following plots and dialogue box:



$$f(x|\mu) = \frac{1}{\mu} e^{-\frac{x}{\mu}} ; x \geq 0$$

N.B. posneg_dv_stats.m requires the statistics and machine learning toolbox.

4.16.4 Adding variables to peak and cluster time series

Peak and cluster time series are a subset of the source data set, saved as an independent record. For some workflows, there may be a need to add other variables with the same date-time as the subseries (e.g. to produce a polar plot of the cluster events, direction must be added to the timeseries, or to run a model (such as overtopping) period and direction may need to be added). This can be done using the Derive Output GUI, by selecting the subseries variable (e.g. wave energy clusters) as X (it must be X) and the source data set variable (e.g. wave period or direction) as the Y variable.

Then for the equation string, use a function of the form $Y.*(X./X)$

This simply adds the variable defined by Y at the intervals defined by X (assuming X is the subseries). The new variable is then named as required. However, if the new variable is *direction* and it is to be used in the plotting routines (e.g. wave roses) the Variable Name MUST begin with 'Dir' but this can be followed by any additional characters.

4.17 Simulation

Allows two simulation models to be run, which are given the shorthand names of YGOR and BMV models.

Warning: ***This is research code. It has been subject to limited testing and requires a detailed knowledge of the source literature and optimization methods to use effectively.***

The optimisation in the YGOR model uses constrained optimisation, such as simulated annealing, which requires the Matlab Global Optimization Toolbox. [YGOR model](#)

YGOR is named after the authors of the paper on which the model is largely based, namely Yates, Guza and O'Reilly (Yates *et al.*, 2009). Additional development was based on the work of (Castelle *et al.*, 2014) and (Villamarin, 2017).

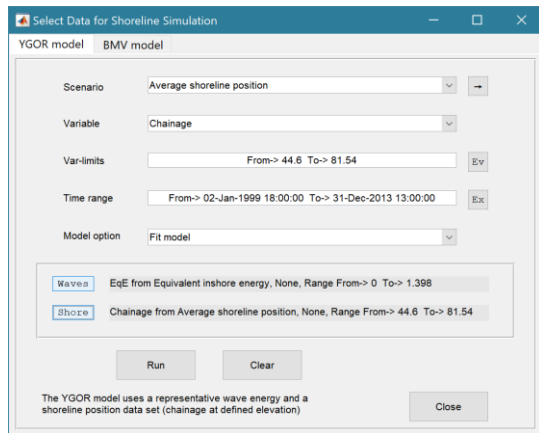
To generate a set of fit parameters, wave forcing and shoreline position data are required and this requires some pre-processing, as explained in Section 3.4.5. The output of the fitting procedure is a set of coefficients that can be used to make simulations (hindcasts or forecasts) using a wave timeseries. The fitting process generates a series of plots and the user is offered the option to adjust the fitting values being used. For each input dialogue box press 'OK' to make a change and 'Cancel' to accept the default values.

The fitting and hindcasting of shoreline position uses a derivative of the Kriebel and Dean model, as proposed by Yates *et al* (Yates *et al.*, 2009). An offset to the initial shoreline is included as an additional fit parameter, as proposed by Castelle *et al* (2014). The model relates nearshore wave energy, (represented by $\frac{H^2}{16}$), to shoreline position, Y . Nearshore wave data are typically obtained using plane bed refraction to account for refraction and shoaling from the wave data point to the closure depth of the beach profile, taking account of water levels variations due to the tide. Fitting the parameters is sensitive to the initial guess and search bounds. Two of the model parameters (a and b) are estimated by finding the line for $dY/dt = 0$ from the mean energy between surveys and the shoreline anomaly (dY) but only uses the points that are close to $dY/dt = 0$. The accretion and erosion rates, C^+ and C^- , are estimated from the mean positive and negative values of dY/dt respectively, and the initial offset is assumed to be zero. Parameter fitting is done using derivative free, constrained non-linear optimisation. Matlab functions for particle swarm, simulated annealing and simplex optimisation are provided as options.

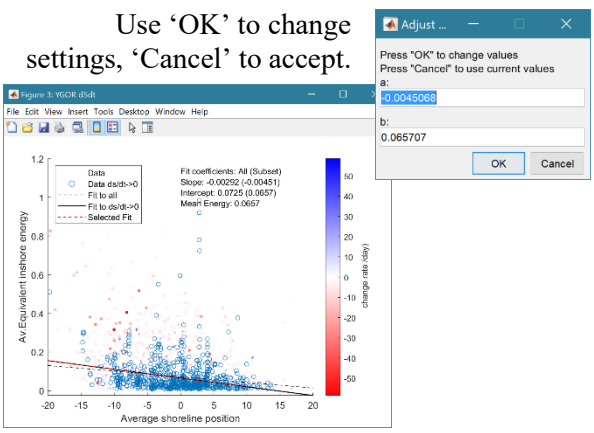
The following images show the steps in a typical workflow. After Step 1, an input dialogue prompts for the Averaging interval to use. This allows the only the defined interval prior to time of each shoreline position to be used as the wave forcing. The default (0) is to use the full wave time series and this generally works better. Once a fit has been accepted in Step 4, some simple regression statistics of model versus observed are generated (Step 5) and the user has the option to add a Taylor diagram (Step 6). If

the Local Skill Window have a value >0 , the local skill score is plotted, followed by the Taylor diagram. The ‘Case List’ button provides full details of the Taylor parameters and the global and local skill scores (if requested). See Section 4.19 for further details about the Taylor Diagram. These statistical tools can be used to help evaluate how well the model is performing.

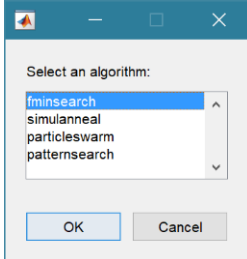
Step 1: make data input selection



Step 2: accept or adjust initial guess of a and b

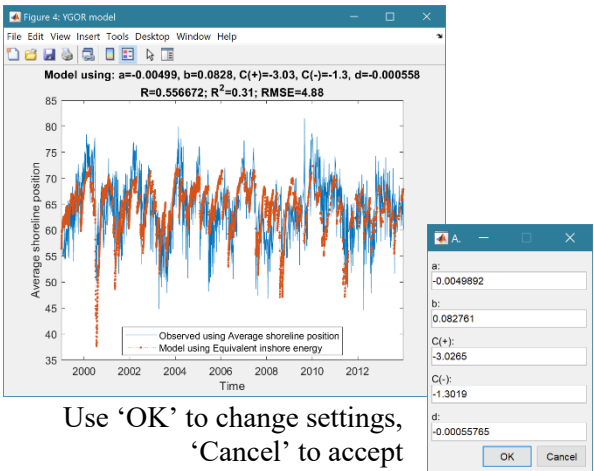


Step 3: select optimisation algorithm



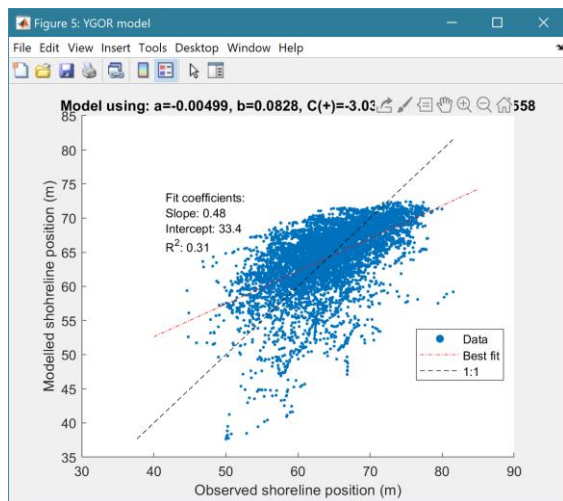
Note - in this UI:
 ‘OK’ selects the algorithm to be used
 ‘Cancel’ aborts the run

Step 4: accept or adjust fit coefficients

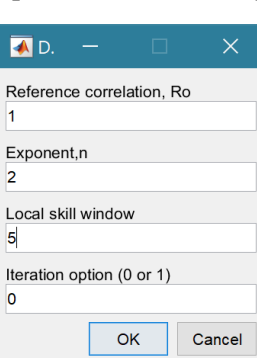


Use ‘OK’ to change settings,
 ‘Cancel’ to accept

Step 5: plot model performance statistics

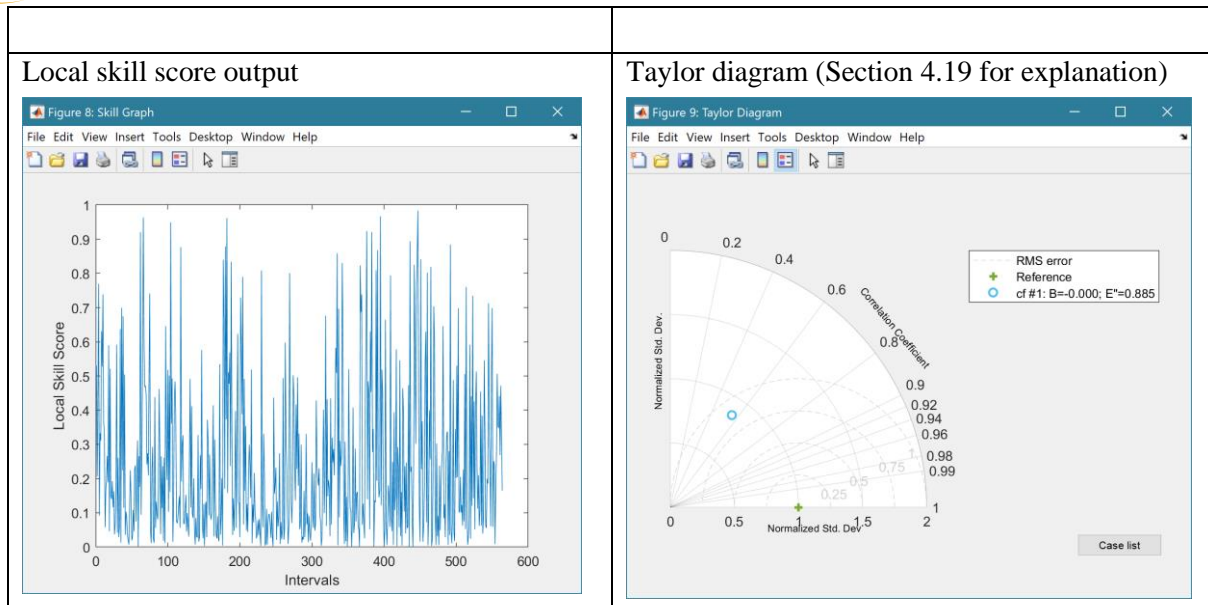


Step 6: optional Taylor diagram of model performance (see Taylor, 2001)



Maximum achievable correlation
 Exponent used in computing the skill score
 Number of grid cells used to define a local window. If $W=0$ (default) the local skill score is not computed.

Local skill score is computed using a sampling window around every time step ($=1$), or computed for all non-overlapping windows ($=0$)



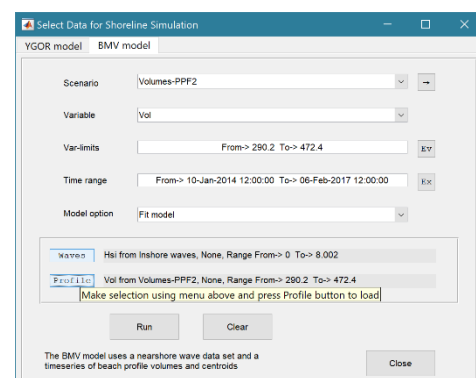
When the fitting of the YGOR model is complete, the option is provided to save the fit coefficients. This updates *Setup>Model parameters>YGOR simulation parameters* UI, which defines the fit coefficients to be used in YGOR simulations (see Section 3.3.2). The current values can be viewed on the *Site/Simulation* tab.

Alternatively, YGOR simulations can be made using any set of coefficients defined in the *Model parameters>YGOR simulation parameters* menu option - see Section 3.3.2. The model then uses the wave timeseries to generate a set of shoreline positions. The wave timeseries can cover any required simulation period. However, the shoreline position variable used to generate the fit coefficients must also be selected for Simulations (this is used to impose any trend applied during the optimisation). The resulting estimates of shoreline position are plotted as a function of time and if the period of the observed data used for fitting the model overlaps the simulation period these are also plotted. Once the run is complete, there is an option to save the output as Shoreline model timeseries.

For an example of the model application see Montano *et al.* (2020).

4.17.2 BMV model

BMV is named after the authors of the papers on which the model is largely based, namely Bernabeu, Medina, and Vidal To generate a set of fit parameters, wave forcing and beach profile volume data are required and this requires some pre-processing, as explained in Section 3.4.5. The output of the fitting procedure is a set of coefficients that can be used to make simulations (hindcasts or forecasts) using a wave timeseries.



The initial model uses an equilibrium profile for the cross-section below high water, as proposed by Bernabeu et al (2003b), Figure 5. Above this there is a runup profile from high water to the maximum level of wave runup, with a slope estimated using the concept of least action (Reis and Gama, 2010). Under some conditions there is a berm before the beach slopes up to the crest of the dune. In some profiles, there is no berm and there is a continuous slope

from the limit of runup to the dune toe. In others, the runup zone extends to an almost vertical cliff at the toe of the dune (yellow profile in Figure 5).

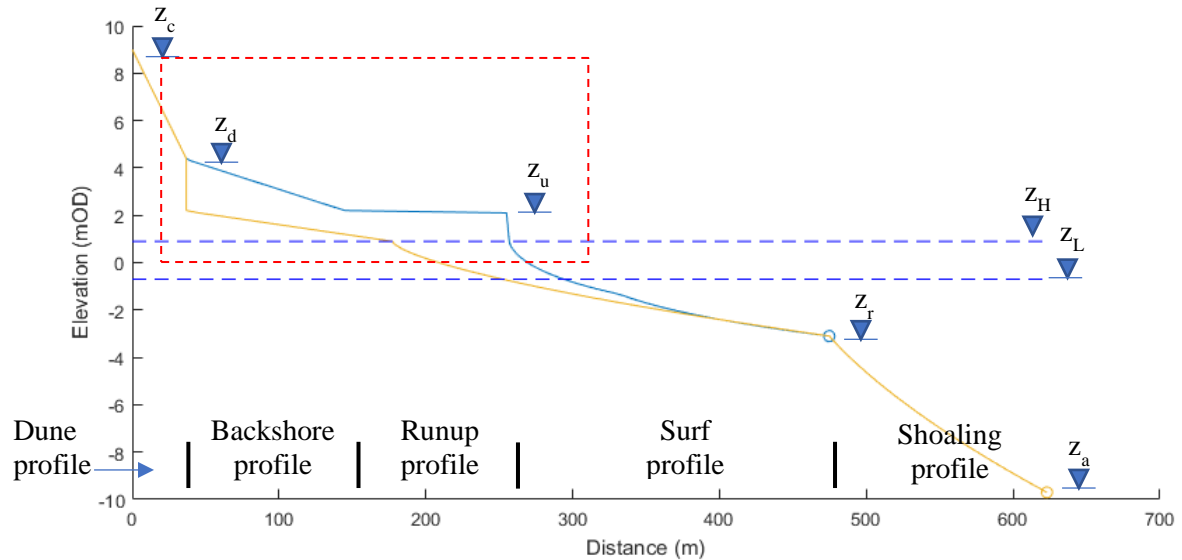
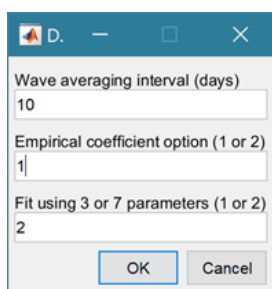


Figure 5 – Definition sketch showing the relevant levels used in the model and illustrating the “minimum profile” based on the maximum wave condition in yellow and the profile for a typical wave condition in blue. Levels of the profile sections shown relate to the blue profile. Red dashed box indicates the control box used when calculating the volume of the profile (see Section 4.13).

Except for the dune and backshore berm/slope the profile can be determined using the incident wave conditions. The hypothesis is that the distance of the equilibrium profile from observed profile at a given point in time represents the anomaly and hence the gradient of the restoring potential, If the same wave condition were to prevail for a period of time, the anomaly would reduce to zero.

After initiating the Run, the user is given the opportunity to make modifications to the run-time options for the model. These parameters are defined in *Setup>Model parameters>BMV simulation parameters*: (see Section 3.3.2



The period (in days) prior to the date of a measured profile over which wave data are averaged.

Empirical coefficient option:

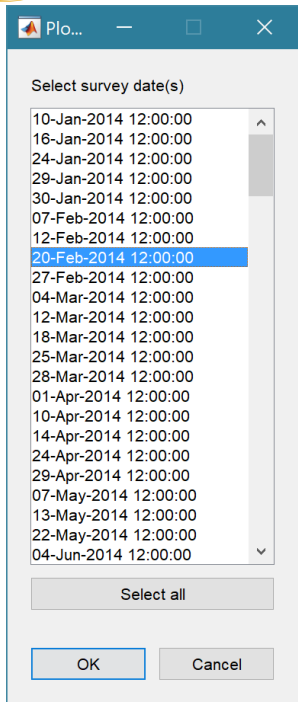
1. Using equilibrium profile coefficients defined in Bernabeu *et al* (2003b)
2. Using equilibrium profile coefficients defined in Bernabeu *et al* (2003a)

Fit using 3 or 7 parameters:

1. Using the parameters for the “dune” profile: i.e. setup coefficient, runup coefficient, dune berm factor defined above.

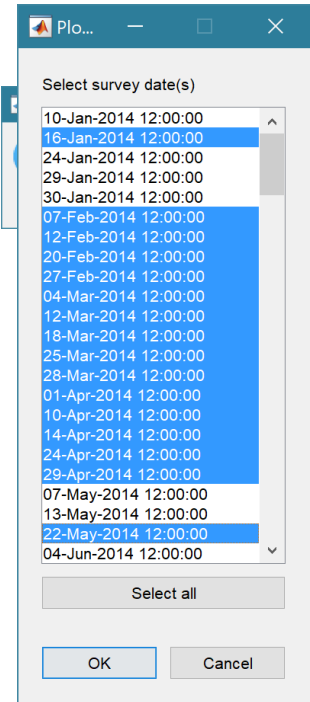
2. Using the upper “dune” profile parameters, as in option 1, and the

equilibrium profile coefficients. The initial estimates defined by the option selected for the Empirical coefficients are relaxed to fit the entire profile.



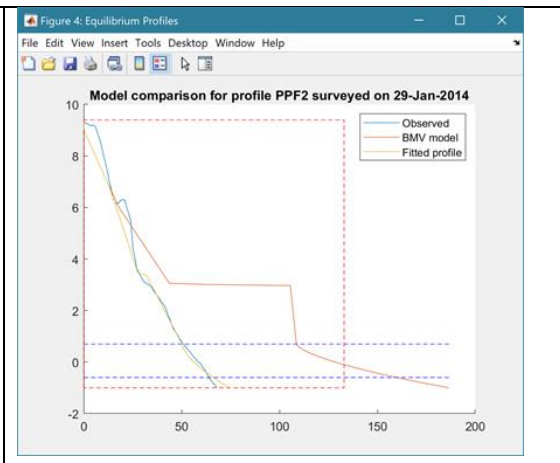
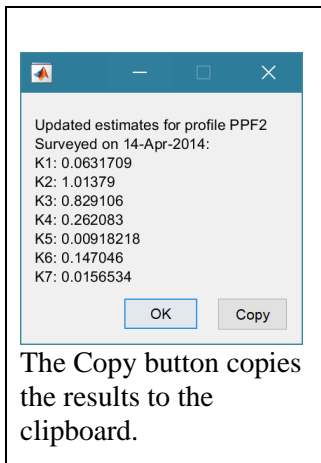
Two workflows are possible using the Fit model option. In one the user selects an individual profile and the parameters can be adjusted and fit re-examined. Once acceptable, the nonlinear curve-fitting (data-fitting) function `lsqcurvefit`, which searches for the best fit in a least-squares sense. The resultant fit parameters are returned as a time series and plotted as shown below

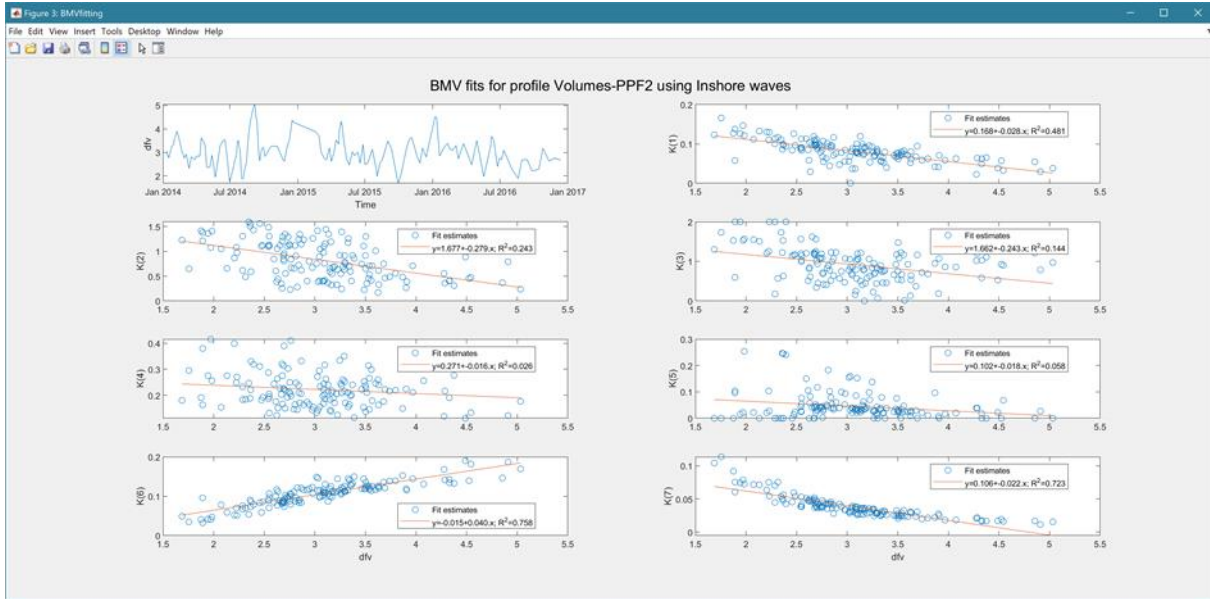
Again, the profile Selection UI re-appears for further analysis, or press ‘Cancel’ to exit.



Output when fitting an individual profile:

Output when multiple profiles are selected:





The Simulation model option computes the volume, and centroid position (x, z) for each wave condition in the time series. The resultant time series is plotted and can be saved as a new data set for more detailed comparison with similar data from the measured profiles.

The BMVmodel simulation option is still under development.

4.18 Beach Vulnerability Index

There are two options accessible from the Run menu, namely a Single site version and a version that examines a beach based on a set of beach profiles (see Section 3.4.6). The basis of these two models are summarised in the following sections. These models make use of the Matlab™ Statistical Toolbox.

4.18.1 Single site BVI (*Run > Vulnerability > BVI site*)

Uses a set of models that are run independently to provide data on wave conditions, runup, littoral drift, cross-shore transport and shoreline position change rates. For further details of these models, see:

- Nearshore waves: Section 4.2
- Runup: Section 4.6
- Littoral drift: Section 4.7
- Cross-shore transport: Section 4.8
- Rate of Shoreline change: Section 4.14

Different scenarios can be considered by generating multiple cases of the above models and then selecting different combinations in the Vulnerability UI, see Section 3.4.6.

Three statistics are computed for four indicators, which are then combined as an arithmetic mean to give a summary statistic. The three statistics are for the relative magnitude index (RMI), coefficient of variation index (CVI) and the beach vulnerability index BVI (a slightly modified version of the form used by Alexandrakis (2014)). The four indicators relate to drift potential, cross-shore exchange, runup, and shoreline change.

The relative magnitude is defined as:

$$RMI = \frac{\mu}{\mu + \sigma}, \text{ where } \mu \text{ is the mean and } \sigma \text{ is the standard deviation.}$$

For the variables that can be positive and negative (cross-shore transport, erosion/accretion), the data are divided into positive and negative values. The positive values are used to compute the index as defined above and the negative values contribute to the index based on:

$RMI = 1 - RMI_{(x < 0)}$. The final index is then the average of the two contribution.

The coefficient of variation is defined as:

$$CVI = \sigma / \mu, \text{ where } \mu \text{ is the mean and } \sigma \text{ is the standard deviation.}$$

Again, the data are divided for variables that can be positive and negative. Using absolute values of the negative component, the two values are averaged.

The BVI is defined for each indicator are as follows:

$$\begin{aligned} \text{Drift:} & \quad QsI = |\bar{Q}_s| / Q_s^{\max} \\ \text{Cross-shore:} & \quad QxI = |Q_x^{on} - Q_x^{off}| / Q_x^{\max} \\ \text{Runup:} & \quad RuI = \bar{zRu} / zCrest \\ \text{Shoreline change:} & \quad \partial xI = N(\overline{\Delta x^+} - \overline{\Delta x^-}) / W_B \end{aligned}$$

where an overbar indicates a mean value and subscript *max* the maximum value. *Q* are rates of littoral or cross-shore transport, the overbar indicates a time mean value, the vertical bars denote absolute values, and the superscripts *on* and *off*, refer to onshore and offshore transport. For runup, *zRu* is the 2% runup elevation (Stockdon *et al.*, 2006), taking into account variations in still water level, and *zCrest* is the specified crest elevation (as defined in Site properties – see Section 3.3.2).

The shoreline change indicator uses rates of shoreline change based on the regression slope of the shoreline position, Δx m/yr (which depends on the elevation used when running the Shoreline model – see Section 4.14) averaged across all the profiles included in the Shoreline model case. The difference of the positive and negative mean rates of change gives the net rate of change. To make this a “representative” distance it is multiplied by 1 year and this is divided by the beach width, W_B . This reflects the potential of losing the beach over this time interval. An upper bound value of 1 is set, should there be the potential to lose the beach in a shorter period. The beach width is computed from indicative high and low water levels (95% percentiles using the still water levels) and mean slope of the profiles at the defined shoreline position.

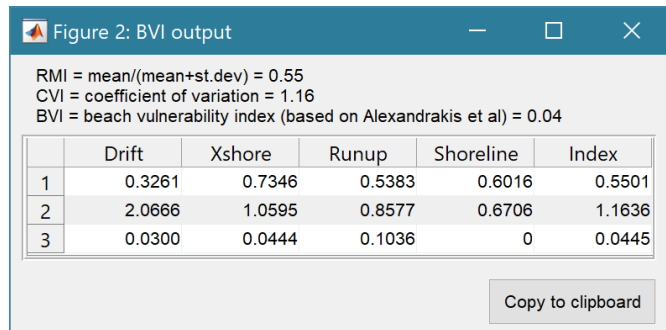
NB: there are no checks in the model for data validity. It is therefore important to examine the contributing data sets and model outputs, to ensure that the data are representative of the conditions and have been suitably quality checked.

Table rows are:

- 1 – RMI = mean/(mean+st.dev)
- 2 – CVI = st.dev/mean
- 3 – BVI = beach vulnerability indicators

4.18.2 Multi-profile BVI (*Run> Vulnerability>BVI profile set;*)

The multi-profile BVI also computes 4 indices and a resultant BVI based on the arithmetic mean of the constituent indices. The method used defines a set of variables that seek to represent the dominant impacts on UK beaches. The definition of vulnerability used to define these variables is as follows:



“Vulnerability is defined as the potential of a coastal system to be harmed by the impact of a storm event and it is quantified by comparing the magnitude of the impact with the adaptation capacity of the system. In simple terms, the impact is characterised by the intensity of storm-induced coastal processes, whereas the adaptation ability is defined by the beach physical properties determining the capacity to cope with considered impacts” (Bosom and Jiménez, 2011).

Following this well-posed definition, data on wave, water level and sediment movement are used to characterise the potential impact. The ability of the coastal system to cope with the induced impacts is measured in terms of beach variables that provide an indication of the ability of the system to contain the impact. The indices are computed using historic data and so represent a “state” of the system over the duration of the data, rather than a projection (e.g by including projected changes in sea level). The indices are defined as summarised in the following table. The indices are computed and tabulated for each profile in the selected beach “set”.

	Measure of impact	Measure of capacity	Index
Inundation	Runup elevation, using waves and water levels (10% exceedance used)	Mean beach crest (Minimum could also be used)	$I_{Ru} = zRu_{10\%} / zCrest$
Beach erosion	Rate of erosion ¹	Mean width of intertidal (HW-LW) and backshore (HW to crest)	$I_{dX} = \frac{ dx/dt _{(<0)}}{\bar{W}_{int} + \bar{W}_{back}} \cdot N$
Beach steepening	Rate of beach rotation (intertidal width narrowing ¹)	Mean width of intertidal (HW-LW)	$I_{dR} = \frac{ dW_{int}/dt _{(<0)}}{\bar{W}_{int}} \cdot N$
Beach mobility	Littoral and cross-shore transport	Dependence on supply ²	$I_{Sm} = \frac{\sum Q_s + \sum Q_x}{\sum Q_s + \sum Q_x }$

Notes:

1 – accretion and widening of the beach are treated as countering vulnerability and result in 0 impact.

2 – when the net transport alongshore and on-offshore tends to zero the beach is less dependent on a supply, Where the exchanges are more unidirectional then the mobility and hence vulnerability is higher.

The variables in the above equations are:

I_{Ru} : uses $zRu_{10\%}$ - the 10% exceedance level of wave runup, $zCrest$ – the average level of the beach crest over the data record. The runup based on the method of Stockdon et al (2006) is added to the water levels to get the runup elevation, see Section 4.6.

I_{dX} : uses dx/dt – rate of horizontal change of the beach based on the average of estimates at HW, MTL and the lowest level on the measured profiles (usually just above LW). Only values that are negative, indicating erosion, are used to measure the index W_{int} – width of the intertidal between HW and LW. W_{back} – width of backshore between HW and the beach crest. Overbars indicate averages over the duration of the data record. N is the number of years for the “exposure” period. If this is large (e.g. 10 or more years) the index can exceed one. In this case the value of the index reported is the number of years for the beach width to be lost, whilst an index value of 1 is used to calculate the BVI. NB: if the reported value of the indicator is 0, this indicates that the profile is accreting and does not therefore contribute to the vulnerability.

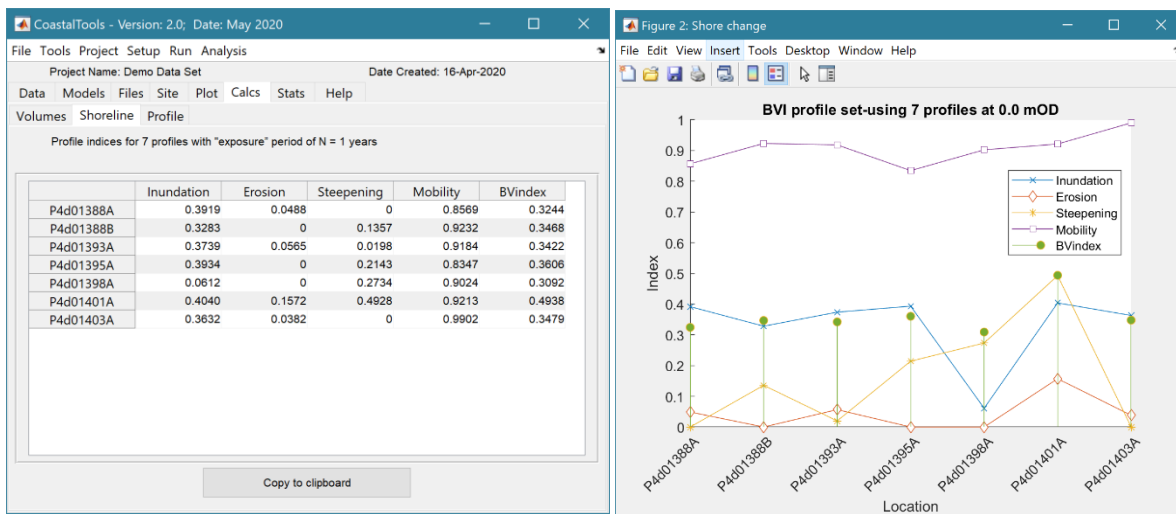
I_{dR} : uses dW_{int}/dt – rate at which the beach width between HW and the lowest level on the measured profiles (usually just above LW) is determined by taking the difference of dx/dt_{HW} and $dx/dt_{z_{min}}$. This is scaled relative to the intertidal beach width, W_{int} . N is the number of years for the “exposure” period, which is applied in the same way as explained above for I_{dX} . NB: if the reported value of the indicator is 0, this indicates that the profile is flattening and does not therefore contribute to the vulnerability.

I_{Sm} : uses the littoral drift, Q_s and the cross-shore transport, Q_x . For details of how these are computed see Sections 4.7 and 4.8.

The Beach Vulnerability Index, BVI, is then given by:

$$BVI = \frac{I_{Ru} + I_{dX} + I_{dR} + I_{Sm}}{4}$$

Tabulated results for each profile are presented on the *Calcs>Shoreline* tab and can be plotted using the *Run> Vulnerability>BVI set plot* menu option. In addition, the results for the Mean Tide Level (0mOD) shoreline change used in the analysis can be viewed in the same way as other Shore change results using the *Run> Beach properties> Shore change>Change plot* and *Rates plot* menu options.



4.19 Taylor diagram

The basis of the Taylor Diagram and associated skill score is explained in Taylor (2001) and summarised here.

4.19.1 Taylor diagram theory

For a variable, X we denote:

Bias as:
$$\bar{E} = \bar{X}_{model} - \bar{X}_{obs}$$

Centred Root Mean Square Difference (or Error) as:

$$E' = \sqrt{\frac{\sum \{ (X_{model} - \bar{X}_{model}) - (X_{obs} - \bar{X}_{obs}) \}^2}{N}}$$

to give the total Mean Square Difference as;
$$E^2 = \bar{E}^2 + E'^2$$

As $E' \rightarrow 0$, patterns become similar, so it is not possible to determine how much of the error is due to a difference in structure and phase and how much is simply due to a difference in the amplitude of the variations.

Standard deviations for model and observed data sets are given by:

$$\sigma_{model} = \sqrt{\frac{\sum (X_{model} - \bar{X}_{model})^2}{N}}$$

$$\sigma_{obs} = \sqrt{\frac{\sum (X_{obs} - \bar{X}_{obs})^2}{N}}$$

and the Correlation coefficient is given by:

$$R = \frac{\sum(X_{model} - \bar{X}_{model})(X_{obs} - \bar{X}_{obs})}{N\sigma_{model}\sigma_{obs}}$$

These measures are related as follows:

$$E'^2 = \sigma_{model}^2 + \sigma_{obs}^2 - 2\sigma_{model}\sigma_{obs}R$$

And by defining R as cos(ϕ) this has the form of the law of cosines equation, where the mean square error and two standard deviations form the sides of a triangle with angle ϕ between the two sides defined by standard deviations.

When comparing different metrics, it can be convenient to use the above in a normalised form, where the normalised variables are denoted by a hat.

Normalised RMS difference:
$$\hat{E}' = \frac{E'}{\sigma_{obs}}$$

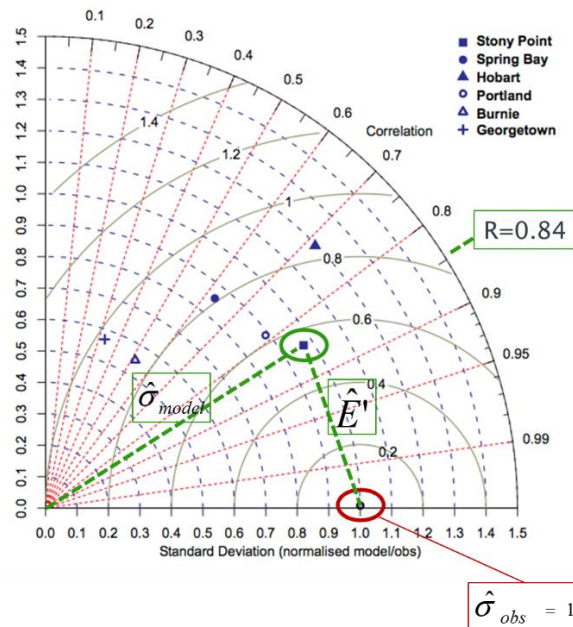
Normalised Standard deviations:
$$\hat{\sigma}_{model} = \frac{\sigma_{model}}{\sigma_{obs}}$$

$$\hat{\sigma}_{obs} = 1$$

Leading to:
$$\hat{E}'^2 = \hat{\sigma}_{model}^2 + \hat{\sigma}_{obs}^2 - 2\hat{\sigma}_{model}\hat{\sigma}_{obs}R$$

whence:
$$\hat{E}'^2 = 1 + \hat{\sigma}_{model}^2 - 2\hat{\sigma}_{model}R$$

The figure below shows the form of the resultant Taylor diagram for one data set.



4.19.2 Skill score

Taylor defines the basis of a good skill score as:

“For any given variance the score should increase monotonically with increasing correlation, and for any given correlation the score should increase as the modelled variance approaches the observed

variance. Traditionally, skill scores have been defined to vary from zero (least skilful) to one (most skilful).”

He then proposes a skill score that achieves this without being too complicated as:

$$S = \frac{4(1 + R)}{(\hat{\sigma}_{\text{model}} + 1/\hat{\sigma}_{\text{model}})^2 (1 + R_0)}$$

where R_0 is the maximum correlation attainable and $S=0$ -poor skill; $S=1$ -good skill. This option weights the pattern variance. However, this is just a model and the weighting can be varied based on what is considered to be most important for the application. E.G. to increase penalty for low correlation, the equation of the following form could be adopted:

$$S = \frac{4(1 + R)^4}{(\hat{\sigma}_{\text{model}} + 1/\hat{\sigma}_{\text{model}})^2 (1 + R_0)^4}$$

Bosboom et al (2014a; 2014b) use similar variations on this basic structure.

In ModelSkill this is implemented as:

$$S = \frac{4(1 + R)^n}{(\hat{\sigma}_{\text{model}} + 1/\hat{\sigma}_{\text{model}})^2 (1 + R_0)^n}$$

The exponent, n , and the maximum correlation attainable, R_0 , are defined in Run Parameters.

The foregoing provides an estimate based on values (mean, standard deviation, etc) for the full grid domain; a global skill score, ‘Sg’. Bosboom and Reniers (2014) also make use of spatially local estimates of the skill score, ‘Sl’, which they obtain by applying a weighting based on distance from each point, when computing the mean, standard deviation and correlation. They then average the local skill score estimates, to obtain an average local skill score. When this is done based on the grid points for the surrounding cells (where the window, W , is the number of grids cells either side of the central point that are to be included) and the weighting is constant across this sub-grid, this is the same as simply sub-sampling the grid. A similar one-dimensional window can be applied to a vector when comparing timeseries data. This is the approach adopted in Coastaltools. The average skill score, Sl, is added to the metadata for the test point in the Taylor diagram and can be viewed by selecting the ‘Case List; button on the Taylor Diagram figure

5 User functions

To allow the user to add other data sets, models or statistical routines, without disrupting the underlying model structure, selected functions have been made external to the class definitions. Copies of the sample files can be found in the `..\muitoolbox\muitemplates` folder. For User Data and User Model these can be copied to a working folder and renamed to suit the application. For User Statistics and User Plots the files in the templates folder can be used to replace the version in the `..\muitoolbox\psfunctions` folder. The `user_plots` and `user_stats` functions are called from the respective UIs, hence the names should not be changed (unless the code in the calling class is also edited).

5.1 User Data

This option allows the user to load data that is different to the ones provided. The class is `muiUserData` and the class file handles the loading and amending of records but requires the file format and meta-data for the variables being loaded to be defined. The file `dataimport_format_template.m`, in the `..\muitoolbox\muitemplates` folder, provides an example of the code needed to load a bespoke file format. This is called from *Setup > Import data > User Data* and loads the data into the generic `muitoolbox muiUserData` class. The key functions that need to be edited in the file are `getData` and `setDSproperties`. The `getData` function defines how to read the data and assign it to a single `dstable` or a named struct array of `dstables`, where each field name provides a reference to the dataset in the table. The `setDSproperties` function defines the metadata needed to describe the variable and any associated dimensions. Examples of the code to load a range of different formats can be found in the `...\muiAppCoastalClasses\FormatFiles` folder.

5.2 User Models

The User Model option (*Run > User Model* - see Section 3.4.7) calls the `user_model` function in the `..\muitoolbox\psfunctions` folder. The user is prompted to define the classname and the function call to run the model (default is `runModel`). This then calls the chosen model and adds any results to the App database (depending on the actions defined in the user model class. The file `Model_template.m`, in the `..\muitoolbox\muitemplates` folder, provides a template for creating a model class. Any existing model classes can also be called using this method.

5.3 User Statistics

On the General and Timeseries tabs of the Statistics UI there is a Statistic list dialogue box. The User option in this list calls the `user_stats.m` function, which can be found in the `..\muitoolbox\psfunctions` folder. This allows the user to define their own workflow, accessing data and functions already provided by the CoastalTools App. The sample code illustrates the workflow for timeseries data to produce a clusters plot. If called from the General statistics tab the code simply returns a warning message. The code could be added to provide some alternative function when called from the General stats.

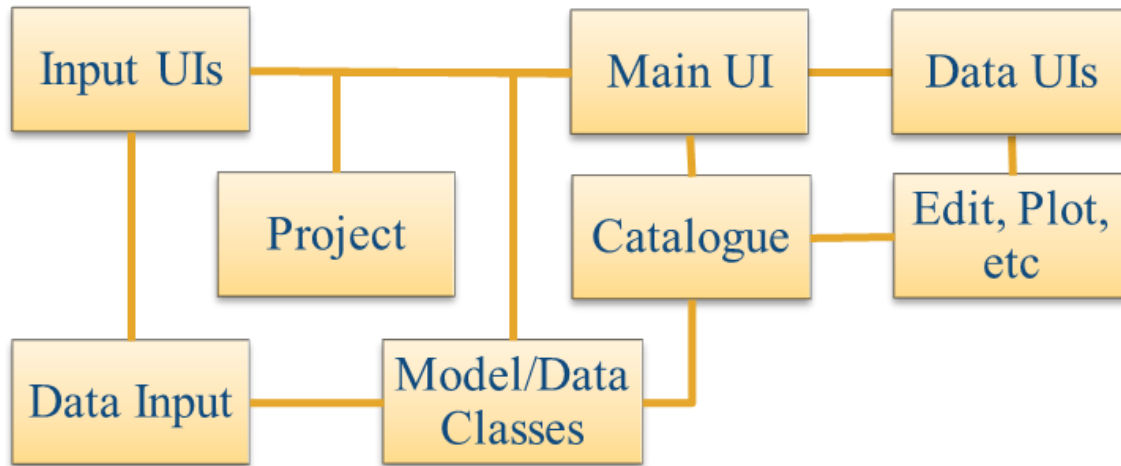
5.4 User Plots

On the Plots UI there is a Plot type list dialogue box. Select User and assign the required variables to the buttons for the selected tab. This passes the variables selected to the `user_plot.m` function, which can be found in the `..\muitoolbox\psfunctions` folder. This allows the user to set up their own plotting function. The demonstration function provided plots a line or surface plot depending on whether a variable has been assigned to the z-dimension.

6 Program Structure

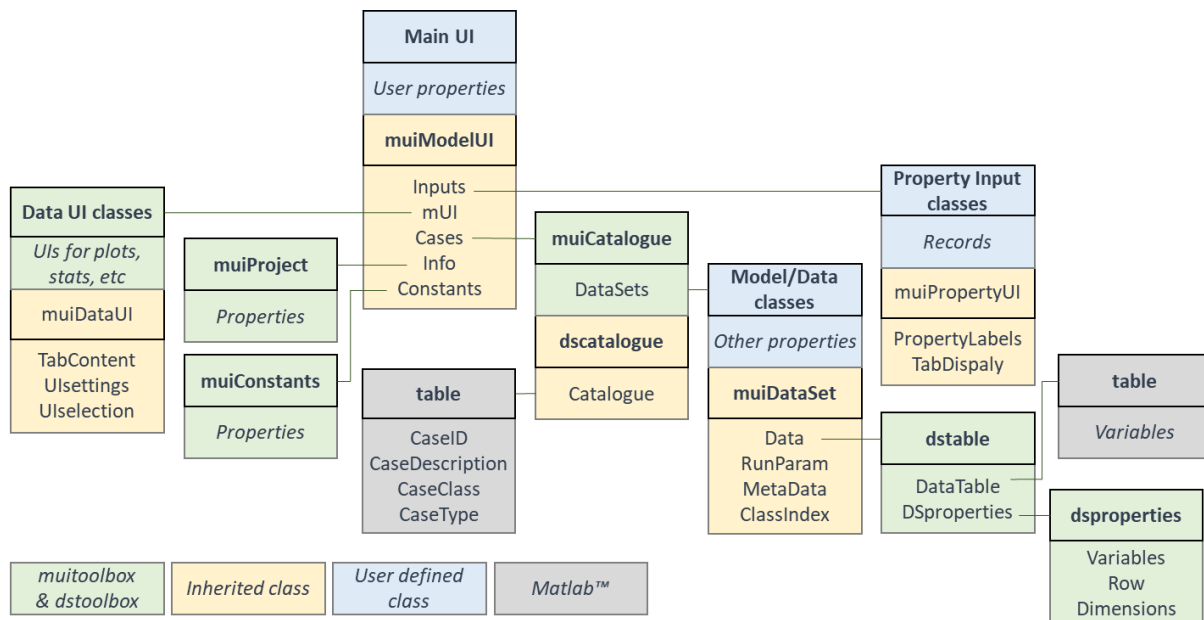
The overall structure of the code is illustrated schematically in Figure 6. This is implemented through several classes that handle the graphical user interface and program workflows (Main UI) and several classes that handle the data manipulation and plotting (Input UIs and Data UIs).

Figure 6 -- High level schematic of program structure



The interfaces and default functionality are implemented in the CoastalTools App using the following mui toolbox classes depicted in Figure 7, which shows a more detailed schematic of the program structure. See the mui toolbox and dstoolbox documentation for more details.

Figure 7 – schematic of program structure showing how the main classes from mui toolbox and dstoolbox are used



In addition, the CoastalTools App uses the following classes and functions:

In .../CoastalTools/

CoastalTools – main UI

CT_BeachAnalysis – tools to analyse beach profile data

CT_PlotsUI – UI for CoastalTools plots (alternative to muiPlotsUI class)

CT_Plots – inherits muiPlots and adds additional plotting options



CT_SimUI – UI for shoreline and profile simulation models

CT_WaveModels – models using wind, wave and water level data

Sim_BMV – runs the BMV beach profile simulation model

Sim_BMVinput – data input for the BMV model

Sim_BMVmodel – setup and running of BMV simulation

Sim_YGOR – runs the YGOR shoreline simulation model

Sim_YGORinput – data input for the YGOR model

Functions:

ct_beachvulnerability – function to compute Beach Vulnerability Index for a set of beach profiles

ct_data_cleanup – set of functions to help clean up timeseries and beach profile data

ct_sitevulnerability – function to compute Beach Vulnerability Index for a single site

simBMVfitting – functions to fit the BMV model to a profile data set

simYGORmodel – functions to run the YGOR model

DemoDataSet.mat – sample file of wave, water level and beach profile data

In .../muiAppCoastalClasses/

ctBeachProfileData – handle beach profiles as Chainage + Elevation and/or Easting, Northings + Elevation.

ctBlueKenueData – Blue Kenue file formats for data a point

ctHindcastParameters – input parameters for the wind-wave model

ctShorelineData – shoreline position data (e.g. distance to mean tide level)

ctStructureInput – definition of a beach or structure used to estimate overtopping quantities

ctTidalAnalysis – analyse water level data to extract tidal constituents and use constituents to construct tidal elevation timeseries.

ctWaterLevelData – import water level data

ctWaveData – import wave data

ctWaveModel – model of wave propagation to a nearshore or deep-water locations

ctWaveParameters – input parameters for the wave models

ctWindData – import wind data

ctWindWaveModel – model wind-wave generation over varying fetch lengths

In .../muiAppCoastalClasses/FormatFiles/

Folder contains a range of format files for loading a number of different data formats and some generic QC functions.

SFcodes – beach profile survey feature codes

In .../muiAppCoastalFcns

addslrtotides – add a linear, or exponentially varying, rate of sea level rise to a tidal record.

beachtransportratio - compute the ratio of onshore to alongshore wave action given by $\tan(\alpha_p)$.

celerity - calculate the wave celerity using Hunt's equation.

coeff_AB - function called by overtopping functions *otop_Q* and *otop_C*.

deanbeachprofile - find the bed slope across the surf zone the profile is based on a user defined slope between Hw and SWL (0mOD) and a Dean profile below this level.

eff_fetch - compute the effective fetch for each mean direction based on the directional distribution function.

get_profile_times - get the composite time intervals for all profiles.

getalp - find the angle between the wave crest and the bed contour.

hallermeier_zones - computes the limits of the surf and shoal zones and returning the surf zone limit and the closure depth or shoal limit.

hb_break - wave height after breaking for given water depth.

hs_break - significant wave height after breaking for given water depth.

hs_surf - calculate the inshore wave height at the edge of the surf zone.

iribarren - function to calculate the Iribarren number which characterise the wave breaker type.

littoraldrift - sediment transport drift rates for sand and shingle.

littoraldriftstats - estimate net drift over selected period from the time series of drift rates.

otop_Q - function to calculate the overtopping discharge for a simple sloping structure based on the Method proposed by Owen.

posneg_dv_stats - computes the rate of change of variable and plot histograms for positive and negative components.

profileslope - calculate the bed slope at some depth within the surf zone.

readfetchfile - read the file that contains the fetch lengths as a function of direction.

refraction - plane bed wave refraction and shoaling using linear wave theory.

rmswaveorbitalvelocity - calculate the root mean square wave orbital velocity.

runup - wave runup magnitude.

runup_slope - calculate runup beach slope using Reis A H and Gama C, 2010.

section_centroid - find centroid and area of a cross-section defined by points (xi,zi).

settling_velocity - calculate the settling velocity using Soulsby equation.

shoaling - plane bed wave shoaling using linear wave theory.

shore_orientation - find the orientation for a series of coordinates that define a line.

shore_profile_order - sort the profile order based on the E,N of the base point - min(Chainage).

simple_tide - function to compute a tidal water level time series using main constituents scaled to required tidal amplitude.

slope_points - find point and slope on line (eg a shore profile nearest to zlevel).

sortENdata2line - sort the eastings and northings into an order that makes the best continuous line

tau_crit - calculate the critical erosion shear stress and erosion rate for sand, mud or mixed sediments.

tma_spectrum - calculate the TMA spectrum for waves that are depth limited.

tidalrange - compute tidal range from a water level time series.

ucrit - compute the critical flow velocity for a given critical shear stress and wave conditions in the wave-current case.

ut_constants.mat - binary file of constituents used by **ctTidalAnalysis** class.

ut_reconstr - reconstruct superposed harmonics (code from Matlab™ Forum).

ut_solv - execute harmonic tidal analysis (code from Matlab™ Forum).



waterlevelfreqplots - create various water level exceedance and duration plots.

wave_energyflux - function to calculate the wave energy flux (wave power) using linear wave theory.

wave_friction - compute the wave friction factor for rough and smooth turbulent conditions and smooth laminar conditions.

wave_scatter - plot H-T scatter diagram that uses the wave celerity based on water depth to add contours of wave steepness, plot a 3D contoured histogram or various combinations of height, period, depth and steepness.

wave_steepness - compute the wave steepness for given wave height, wave period and water depth.

xshore_bailard - computes the cross-shore transport for given wave and beach conditions.

7 Bibliography

Abramov V and Khan M K, 2017, A Practical Guide to Market Risk Model Validations (Part II - VaR Estimation). 70, <http://dx.doi.org/10.2139/ssrn.3080557>.

Alexandrakis G and Poulos S, 2014, An holistic approach to beach erosion vulnerability assessment. *Sci Rep*, 4, 6078, 10.1038/srep06078.

Antoniades I P, Brandi G, Magafas L and Di Matteo T, 2021, The use of scaling properties to detect relevant changes in financial time series: A new visual warning tool. *Physica A: Statistical Mechanics and its Applications*, 565, 10.1016/j.physa.2020.125561.

Bailard J A, 1981, An energetics total load sediment transport model for a plane sloping beach. *Journal of Geophysical Research: Oceans*, 86 (C11), 10938-10954, 10.1029/JC086iC11p10938.

Bailard J A, 1982, Modelling on-offshore sediment transport in the surf zone, *International Conference on Coastal Engineering*, pp. 1419-1438, ASCE.

Bailard J A and Inman D L, 1981, An energetics bedload model for a plane sloping beach: Local transport. *Journal of Geophysical Research: Oceans*, 86 (C3), 2035-2043, 10.1029/JC086iC03p02035.

Bernabeu A M, Medina R and Vidal C, 2003a, A morphological model of the beach profile integrating wave and tidal influences. *Marine Geology*, 197 (1-4), 95-116, 10.1016/s0025-3227(03)00087-2.

Bernabeu A M, Medina R and Vidal C, 2003b, Wave reflection on natural beaches: an equilibrium beach profile model. *Estuarine, Coastal and Shelf Science*, 57 (4), 577-586.

Bosboom J and Reniers A J H M, 2014, Scale-selective validation of morphodynamic models, 34th *International Conference on Coastal Engineering*, pp. 1911–1920, Seoul, South-Korea.

Bosboom J, Reniers A J H M and Luijendijk A P, 2014, On the perception of morphodynamic model skill. *Coastal Engineering*, 94, 112-125, <https://doi.org/10.1016/j.coastaleng.2014.08.008>.

Bosom E and Jiménez J A, 2011, Probabilistic coastal vulnerability assessment to storms at regional scale – application to Catalan beaches (NW Mediterranean). *Natural Hazards and Earth System Science*, 11 (2), 475-484, 10.5194/nhess-11-475-2011.

Bouws E, Günther H, Rosenthal W and Vincent C L, 1985, Similarity of the wind wave spectra in finite depth water: Part 1 - spectral form. *Journal of Geophysical Research*, 90 (C1), 975-986.

Bouws E, Günther H, Rosenthal W and Vincent C L, 1987, Similarity of the wind wave spectrum in finite depth water: Part 2 - statistical relations between shape and growth stage parameters. *Dt.hydrogr.Z*, 40, 1-24.

Brandi G and Di Matteo T, 2021, On the statistics of scaling exponents and the multiscaling value at risk. *The European Journal of Finance*, 1-22, 10.1080/1351847x.2021.1908391.

Carter D J, 1982, Estimation of wave spectra from wave height and period, MIAS Publication, Report No: 4, pp. 1-21, Godalming.

Castelle B, Marieu V, Bujan S, Ferreira S, Parisot J-P, Capo S, Sénéchal N and Chouzenoux T, 2014, Equilibrium shoreline modelling of a high-energy meso-macrotidal multiple-barred beach. *Marine Geology*, 347, 85-94, 10.1016/j.margeo.2013.11.003.

- Codiga D L, 2011, Unified Tidal Analysis and Prediction Using the UTide Matlab Functions, Report No: 2011-01,p. 59, Graduate School of Oceanography, University of Rhode Island, Narragansett, RI.
- Coles S, 2001, An Introduction to Statistical Modeling of Extreme Values, Springer-Verlag, London.
- Dean R G, 1977, Equilibrium beach profiles: US Atlantic and Gulf coasts, Report No: Ocean Engineering Report No 12, Newark, Delaware.
- Di Matteo T, Aste T and Dacorogna M M, 2003, Scaling behaviors in differently developed markets. *Physica A: Statistical Mechanics and its Applications*, 324 (1-2), 183-188, 10.1016/s0378-4371(02)01996-9.
- Donelan M A, Hamilton J and Hui W H, 1985, Directional Spectra of wind-generated waves. *Phil.Trans.R.Soc.Lond.A.*, 315 (1534), 509-562.
- Hallermeier R J, 1981, A profile zonation for seasonal sand beaches from wave climate. *Coastal Engineering*, 4 (3), 253-277.
- Hsu J R-C, Yu M J, Lee F C and Benedet L, 2010, Static bay beach concept for scientists and engineers: A review. *Coastal Engineering*, 57 (2), 76-91.
- Hsu J R C and Evans C, 1989, Parabolic bay shapes and applications. *Proceedings - Institution of Civil Engineers.Part 2.Research and theory*, 87, 557-570.
- Hsu J R C, Silvester R and Xia Y M, 1989, Applications of headland control. *Journal of Waterway, Port, Coastal and Ocean Engineering*, 115 (3), 299-326.
- Hughes S A, 1984, The TMA shallow-water spectrum description and applications, Coastal Engineering Research Centre, Report No: CERC-84-7,pp. 1-39, Washington DC.
- Ihlen E A, 2012, Introduction to multifractal detrended fluctuation analysis in matlab. *Front Physiol*, 3, 141, 10.3389/fphys.2012.00141.
- Kamphuis W J, 1991, Alongshore sediment transport rate. *ASCE, Journal of the Waterway, Port, Coastal and Ocean Division*, 117 (6), 624-640.
- Kitaigorodskii S A, Krasitskii V P and Zaslavskii M M, 1975, On Phillips theory of equilibrium range in the spectra of wind-generated gravity waves. *Journal of Physical Oceanography. Journal of Physical Oceanography*, 5 (3), 410-420.
- Masselink G, Austin M, Tinker J, O'Hare T and Russell P, 2008, Cross-shore sediment transport and morphological response on a macrotidal beach with intertidal bar morphology, Truc Vert, France. *Marine Geology*, 251 (3-4), 141-155, 10.1016/j.margeo.2008.01.010.
- Medina R, Bernabeu A M, Vidal C and González M, 2004, A general equilibrium beach profile formulation, In: *SASME Book of Abstracts, Series*, pp. 1-3.
- Montano J, Coco G, Antolinez J A A, Beuzen T, Bryan K R, Cagigal L, Castelle B, Davidson M A, Goldstein E B, Ibaceta R, Idier D, Ludka B C, Masoud-Ansari S, Mendez F J, Murray A B, Plant N G, Ratliff K M, Robinet A, Rueda A, Senechal N, Simmons J A, Splinter K D, Stephens S, Townend I, Vitousek S and Vos K, 2020, Blind testing of shoreline evolution models. *Sci Rep*, 10 (1), 2137, <https://doi.org/10.1038/s41598-020-59018-y>.

- Morales R, Di Matteo T, Gramatica R and Aste T, 2012, Dynamical generalized Hurst exponent as a tool to monitor unstable periods in financial time series. *Physica A: Statistical Mechanics and its Applications*, 391 (11), 3180-3189, <https://doi.org/10.1016/j.physa.2012.01.004>.
- Owen M W, 1980, Design of seawalls allowing for wave overtopping, Report No: EX 924, Wallingford.
- Pacheco J C R, Román D T and Vargas L E, 2008, R/S Statistic: Accuracy and Implementations, 18th International Conference on Electronics, Communications and Computers (conielecomp 2008), IEEE, 10.1109/conielecomp.2008.14.
- Reis A H and Gama C, 2010, Sand size versus beachface slope — An explanation based on the Constructal Law. *Geomorphology*, 114 (3), 276-283, 10.1016/j.geomorph.2009.07.008.
- Silvester R, 1985, Natural headland control of beaches. *Continental Shelf Research*, 4 (5), 581-596.
- Silvester R and Hsu J R C, 1997, *Coastal Stabilization*, World Scientific Publishing, Singapore.
- Soulsby R, 1997, *Dynamics of marine sands*, Thomas Telford, London.
- Stockdon H F, Holman R A, Howd P A and Sallenger A H, 2006, Empirical parameterization of setup, swash, and runup. *Coastal Engineering*, 53 (7), 573-588, 10.1016/j.coastaleng.2005.12.005.
- Sutcliffe J, Hurst S, Awadallah A G, Brown E and Hamed K, 2016, Harold Edwin Hurst: the Nile and Egypt, past and future. *Hydrological Sciences Journal*, 61 (9), 1557-1570, 10.1080/02626667.2015.1019508.
- Taylor K E, 2001, Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research - Atmospheres*, 106 (D7), 7183-7192, 10.1029/2000JD900719.
- Tucker M J, Carr A P and Pitt E G, 1983, The effect of an offshore bank in attenuating waves. *Coastal Engineering*, 7 (2), 133-144, [http://dx.doi.org/10.1016/0378-3839\(83\)90009-1](http://dx.doi.org/10.1016/0378-3839(83)90009-1).
- USACE, 1984, *Shore Protection Manual*, US Army Corps, Vicksburg.
- Villamarin B C, 2017, *Beach Cross-shore dynamics: Applicability of Equilibrium Shoreline models*. Masters thesis, Universidad del Pais Vasco, Plentzia.
- Wright L D and Short A D, 1984, Morphodynamic variability of surf zones and beaches: A synthesis. *Marine Geology*, 56 (1-4), 93-118, [http://dx.doi.org/10.1016/0025-3227\(84\)90008-2](http://dx.doi.org/10.1016/0025-3227(84)90008-2).
- Yates M L, Guza R T and O'Reilly W C, 2009, Equilibrium shoreline response: Observations and modeling. *Journal of Geophysical Research*, 114 (C9), 10.1029/2009jc005359.

Appendix A – Input Data File Formats

Existing file formats are detailed in this section. For details of how to add a file format to an existing data class, or add a new data class see Appendix B.

Waves

1) Channel Coastal Observatory (CCO) format: <http://www.channelcoast.org/>

Date/Time (GMT)	Latitude	Longitude	Flag	Hs (Hm0) (m)	Hmax (m)	Tp (s)	Tz (Tm) (s)	Dirp (degrees)	Spread (deg)	SST (deg C)
01-Jan-2004 00:00:00	50.73422	-0.49434	3	1.510	2.600	4.8	3.9	183	28	9999.0
01-Jan-2004 00:30:00	50.73412	-0.49400	3	1.630	2.580	5.0	4.0	176	27	9999.0
01-Jan-2004 01:00:00	50.73412	-0.49400	3	1.940	3.300	5.3	4.3	176	19	9999.0
01-Jan-2004 01:30:00	50.73409	-0.49406	3	2.030	3.900	5.6	4.3	177	23	9999.0
01-Jan-2004 02:00:00	50.73410	-0.49402	3	2.270	3.420	5.6	4.5	183	18	9999.0
01-Jan-2004 02:30:00	50.73410	-0.49404	3	2.360	3.330	5.9	4.6	176	21	9999.0
01-Jan-2004 03:00:00	50.73410	-0.49402	3	2.410	3.740	6.3	4.8	184	15	9999.0
01-Jan-2004 03:30:00	50.73410	-0.49400	3	2.340	3.780	6.3	4.9	186	18	9999.0
01-Jan-2004 04:00:00	50.73417	-0.49402	3	2.600	3.830	6.7	4.9	191	15	9999.0
01-Jan-2004 04:30:00	50.73418	-0.49406	3	2.560	3.740	6.3	5.0	194	17	9999.0

2) Simple wave record

Format is defined using Matlab script in the header line. The format shown here is date, time (hours and minutes only), Hs, Tp, Dir all as real numbers. This allows different date formats to be easily handled. The utility `add_file_header.m` (located in MUIfunctions folder) can be used to add a file header to multiple files.

1	%(dd/MM/yyyy)D	%(HH:mm)D	%f	%f	%f
2	01/01/1980	00:00	0.274	9.35	217
3	01/01/1980	03:00	0.256	9.26	211.5
4	01/01/1980	06:00	0.252	9.26	211
5	01/01/1980	09:00	0.25	9.17	210.6
6	01/01/1980	12:00	0.248	9.17	209.9
7	01/01/1980	15:00	0.248	9.09	208.9
8	01/01/1980	18:00	0.246	14.08	207.7
9	01/01/1980	21:00	0.278	13.89	203.3
10	02/01/1980	00:00	0.378	13.7	179.3
11	02/01/1980	03:00	0.402	13.7	258.9
12	02/01/1980	06:00	0.414	13.51	251.3
13	02/01/1980	09:00	0.418	13.33	224
14	02/01/1980	12:00	0.43	13.16	211.9
15	02/01/1980	15:00	0.468	12.99	217.6
16	02/01/1980	18:00	0.436	12.99	203.4
17	02/01/1980	21:00	0.442	13.16	200.6

3) ShoreCast file format (date and time are in vector format followed by variables)

Year	Month	Day	Hour[NZST]	Min	Sec	Hs[m]	Tp[s]	Tm01[s]	Tm02[s]	Dir[°]
1979	1	1	12	0	0	1.860000	12.093000	11.615000	11.303000	53.447000
1979	1	1	15	0	0	1.860000	12.097000	11.625000	11.338000	53.866000
1979	1	1	18	0	0	1.859000	12.108000	11.636000	11.365000	54.076000
1979	1	1	21	0	0	1.851000	12.091000	11.626000	11.365000	54.219000
1979	1	2	0	0	0	1.823000	12.027000	11.598000	11.345000	54.491000
1979	1	2	3	0	0	1.771000	11.900000	11.505000	11.186000	54.856000
1979	1	2	6	0	0	1.690000	11.758000	11.301000	10.744000	55.202000
1979	1	2	9	0	0	1.593000	11.617000	11.158000	10.719000	55.567000
1979	1	2	12	0	0	1.499000	11.475000	11.001000	10.672000	55.958000
1979	1	2	15	0	0	1.420000	11.316000	10.826000	10.563000	56.397000

Water Levels

1) Date-Level format

Format is defined using Matlab script in header line. Format shown here is record number (not used), date, time, elevation (mOD). This allows different date formats to be easily handled. The utility `add_file_header.m` (located in MUIfunctions folder) can be used to add a file header to multiple files.



```
%*s % {yyyy/MM/dd}D % {HH:mm:ss}D %q
262921) 2000/01/01 00:00:00 1.599
262922) 2000/01/01 00:15:00 1.604
262923) 2000/01/01 00:30:00 1.633
262924) 2000/01/01 00:45:00 1.669
262925) 2000/01/01 01:00:00 1.704
262926) 2000/01/01 01:15:00 1.750
262927) 2000/01/01 01:30:00 1.804
262928) 2000/01/01 01:45:00 1.868
262929) 2000/01/01 02:00:00 1.926
262930) 2000/01/01 02:15:00 1.986
262931) 2000/01/01 02:30:00 2.038
```

2) Channel Coastal Observatory (CCO) format: <http://www.channelcoast.org/>

Date/Time (GMT)	Tide_OD	Tide_CD	Residual (m)	Flag
04-Jul-2008 14:50:00	-0.262	2.388	0.328	1
04-Jul-2008 15:00:00	-0.190	2.460	0.307	1
04-Jul-2008 15:10:00	-0.098	2.552	0.309	1
04-Jul-2008 15:20:00	-0.006	2.644	0.311	1
04-Jul-2008 15:30:00	0.095	2.745	0.320	1
04-Jul-2008 15:40:00	0.168	2.818	0.299	1
04-Jul-2008 15:50:00	0.298	2.948	0.330	1
04-Jul-2008 16:00:00	0.429	3.079	0.355	1
04-Jul-2008 16:10:00	0.498	3.148	0.310	1
04-Jul-2008 16:20:00	0.620	3.270	0.311	1
04-Jul-2008 16:30:00	0.711	3.361	0.271	1
04-Jul-2008 16:40:00	0.858	3.508	0.279	1
04-Jul-2008 16:50:00	0.975	3.625	0.250	1
04-Jul-2008 17:00:00	1.124	3.774	0.247	1
04-Jul-2008 17:10:00	1.297	3.947	0.264	1
04-Jul-2008 17:20:00	1.470	4.120	0.279	1
04-Jul-2008 17:30:00	1.631	4.281	0.283	1
04-Jul-2008 17:40:00	1.784	4.434	0.283	1
04-Jul-2008 17:50:00	1.920	4.570	0.273	1

3) British Oceanographic Data Centre format: <http://www.bodc.ac.uk/>

3(a) BODC NTSLF format

```
Port: P008
Site: Portsmouth
Latitude: 50.80256
Longitude: -1.11175
Start Date: 01JAN1991-00.00.00
End Date: 31DEC1991-23.00.00
Contributor: National Oceanography Centre, Liverpool
Datum information: The data refer to Admiralty Chart Datum (ACD)
Parameter code: ASLVBG02 = Surface elevation (unspecified datum) of the water body by bubbler tide gauge
Cycle Date Time ASLVBG02 Residual
Number WWW mm dd hh mi ss f f
689) 1991/01/29 16:00:00 0.618 -0.227
690) 1991/01/29 17:00:00 1.076 -0.155
691) 1991/01/29 18:00:00 1.634 -0.119
692) 1991/01/29 19:00:00 1.946 -0.138
693) 1991/01/29 20:00:00 2.508 -0.133
694) 1991/01/29 21:00:00 3.436 -0.117
695) 1991/01/29 22:00:00 4.291 -0.079
696) 1991/01/29 23:00:00 4.592 -0.046
697) 1991/01/30 00:00:00 4.455 -0.046
698) 1991/01/30 01:00:00 4.151 -0.050
699) 1991/01/30 02:00:00 3.214 -0.049
700) 1991/01/30 03:00:00 1.734 -0.157
```

3(b) BODC ODV format

```
//Data documentation at https://www.bodc.ac.uk/data/documents/series/537651/
//If you find any issues with this data please report it to enquiries@bodc.ac.uk
//SDN_parameter_mapping
//<subject>SDN:LOCAL:Chronological Julian Date</subject><object>SDN:F01::CJDY1101</object><units>SDN:P06::UTAA</units>
//<subject>SDN:LOCAL:SeaLvl_bubbler</subject><object>SDN:F01::ASLVBG01</object><units>SDN:P06::ULAA</units>
//<subject>SDN:LOCAL:SeaLvl_bubbler2</subject><object>SDN:F01::ASLVBG02</object><units>SDN:P06::ULAA</units>
//<subject>SDN:LOCAL:HTSeaLvl</subject><object>SDN:F01::ASLVBH01</object><units>SDN:P06::ULAA</units>
//
Cruise Station Type WWW-MM-ddThh:mm:ss.ggg Longitude [degrees_east] Latitude [degrees_north] LOCAL_CDI_ID EDMO_code Bot. Depth [m]
Chronological Julian Date [days] QV:SEADATANET SeaLvl_bubbler [m] QV:SEADATANET SeaLvl_bubbler2 [m] QV:SEADATANET HTSeaLvl [m] QV:SEADATANET
unspecified 988/1996/172 * 1996-06-20T10:00:17.000 -1.8749 50.7143 537651 43 0 2450255.416858 1 1.862 1 1.862 1 1.872 1
2450255.427179 1 1.877 8 1.869 8 1.880 8
2450255.437500 1 1.901 1 1.892 1 1.901 1
2450255.447917 1 1.920 1 1.913 1 1.920 1
2450255.458333 1 1.920 1 1.913 1 1.920 1
2450255.468750 1 1.904 1 1.895 1 1.901 1
2450255.479167 1 1.898 1 1.890 1 1.896 1
2450255.489583 1 1.884 1 1.882 1 1.888 1
2450255.500000 1 1.860 1 1.859 1 1.864 1
2450255.510417 1 1.831 1 1.828 1 1.834 1
2450255.520833 1 1.799 1 1.796 1 1.802 1
2450255.531250 1 1.765 1 1.763 1 1.765 1
2450255.541667 1 1.731 1 1.726 1 1.736 1
2450255.552083 1 1.713 1 1.707 1 1.718 1
2450255.562500 1 1.705 1 1.698 1 1.705 1
```

4) ShoreCast file format (date and time are in vector format followed by variables)

```
Year Month Day Hour[NZST] Min Sec tide[m]
1968 1 1 2 0 0 -0.796000
1968 1 1 3 0 0 -0.462000
1968 1 1 4 0 0 -0.009000
1968 1 1 5 0 0 0.454000
1968 1 1 6 0 0 0.820000
1968 1 1 7 0 0 1.003000
1968 1 1 8 0 0 0.963000
1968 1 1 9 0 0 0.717000
1968 1 1 10 0 0 0.228000
```

Winds

Three formats are currently implemented.

1) Date-Record format

The format read statement is given as the header of the data. The default definition assumes date, time, wind speed (m/s) direction (degTN). A typical header would then be:

```
%{yyyy-MM-dd}D %{HH:mm:ss}D %f%f
```

This is similar to the wave and water level date-record format files.

Note: the order of the year-month-date format needs to match that used in the file. This allows different date and record formats to be handled. The utility `add_file_header.m` (located in MUIfunctions folder) can be used to add a file header to multiple files.

```
#{dd/MM/yyyy}D #{HH:mm}D %f %f
01/01/1980 00:00 7.1 80
01/01/1980 03:00 7.2 75
01/01/1980 06:00 7.5 80
01/01/1980 09:00 7.1 80
01/01/1980 12:00 7 80
01/01/1980 15:00 5.9 70
01/01/1980 18:00 5.1 80
01/01/1980 21:00 5.3 80
02/01/1980 00:00 6.2 90
02/01/1980 03:00 999 80
02/01/1980 06:00 4.8 80
02/01/1980 09:00 6.44 80
02/01/1980 12:00 6.6 80
02/01/1980 15:00 7 80
```

Note that if the variables in the record change, then a new file format will be needed to define the variables (see Appendix B).

2) CEDA MIDAS format

This data is available from:

<https://catalogue.ceda.ac.uk/data.ceda.ac.uk/badc/ukmo-midas-open/data/uk-mean-wind-obs/dataset-version-201908/>

The data is delivered as a csv file that can be opened in Excel and loaded directly into CoastalTools. The files have a very large header, with a lot of useful meta-data, so an image is not reproduced here.

3) Hong Kong data format



You may need to edit the wind file to replace the header as shown below:

1	Date	hour	Dir10min	Speed10min	Dir1hr	Speed1hr
2	20180913	01	80	7.1	80	7.2
3	20180913	02	80	7.1	80	6.7
4	20180913	03	80	7.1	80	7.5
5	20180913	04	80	7.7	80	7.1
6	20180913	05	80	6.3	80	7.0
7	20180913	06	80	5.4	80	5.9
8	20180913	07	70	6.0	80	5.1
9	20180913	08	80	5.8	80	5.3
10	20180913	09	80	6.3	80	6.2
11	20180913	10	90	2.9	80	5.0
12	20180913	11	80	6.1	80	4.8
13	20180913	12	80	7.4	80	6.4
14	20180913	13	80	6.6	80	6.6
15	20180913	14	80	6.9	80	7.0
16	20180913	15	80	6.5	80	7.0
17	20180913	16	90	5.7	80	6.7
18	20180913	17	90	5.9	90	6.2
19	20180913	18	90	5.7	90	5.7
20	20180913	19	90	5.5	90	5.4

Wind-wave Hindcast Fetches

For each location on the coast you will need a fetch file. The file Test_Fetch_Lengths.txt is an example file. This defines the fetch lengths at the site in 10-degree intervals measured from true north (defined from 0 to 350 degrees even if zero).

1	Dir	Fetch
2	0	0
3	10	0
4	20	0
5	30	1500
6	40	2300
7	50	16000
8	60	17500
9	70	23000
10	80	22000

Beach Profiles

1) Channel Coastal Observatory (CCO) format: <http://www.channelcoast.org/>

Easting	Northing	Elevation_OD	Chainage	FC	Profile	Reg_ID
488072.856	95988.852	6.364	130.396	G	112	4d01403
488075.556	95986.645	6.381	133.882	G	112	4d01403
488078.392	95984.552	6.446	137.406	G	112	4d01403
488081.390	95982.129	6.501	141.260	G	112	4d01403
488084.150	95979.926	6.573	144.791	G	112	4d01403
488087.123	95977.767	6.619	148.463	G	112	4d01403
488089.877	95975.476	6.691	152.044	G	112	4d01403
488092.331	95973.597	6.659	155.134	G	112	4d01403
488094.608	95971.872	4.963	157.991	G	112	4d01403
488095.918	95970.882	4.810	159.632	G	112	4d01403
488096.593	95970.270	4.808	160.541	G	112	4d01403
488097.447	95969.618	5.052	161.615	G	112	4d01403
488098.132	95969.069	5.117	162.493	G	112	4d01403

2) Chainage profiles listed as one, or more, profile with one, or more, surveys (dates) per file

1	Site	ProfileID	Date	Chainage	Elevation	Flag
2	NARRA	PF6	1976-04-27	0	6.29	EMERY
3	NARRA	PF6	1976-04-27	10	1.68	EMERY
4	NARRA	PF6	1976-04-27	20	0.1	EMERY
5	NARRA	PF6	1976-04-27	30	-0.95	EMERY
6	NARRA	PF8	1976-04-27	0	3.42	EMERY
7	NARRA	PF8	1976-04-27	10	2.65	EMERY
8	NARRA	PF8	1976-04-27	20	2.24	EMERY
9	NARRA	PF8	1976-04-27	30	1.9	EMERY
10	NARRA	PF8	1976-04-27	40	1.42	EMERY
11	NARRA	PF8	1976-04-27	50	1.18	EMERY
12	NARRA	PF8	1976-04-27	60	0.88	EMERY
13	NARRA	PF8	1976-04-27	70	0.36	EMERY
14	NARRA	PF8	1976-04-27	80	-0.19	EMERY
15	NARRA	PF8	1976-04-27	90	-0.51	EMERY
16	NARRA	PF2	1976-04-27	0	9.25	EMERY
17	NARRA	PF2	1976-04-27	10	7.92	EMERY
18	NARRA	PF2	1976-04-27	20	6.75	EMERY
19	NARRA	PF2	1976-04-27	30	3.72	EMERY
20	NARRA	PF2	1976-04-27	40	2.54	EMERY
21	NARRA	PF2	1976-04-27	50	2.15	EMERY
22	NARRA	PF2	1976-04-27	60	0.66	EMERY
23	NARRA	PF2	1976-04-27	70	0.18	EMERY
24	NARRA	PF2	1976-04-27	80	-0.09	EMERY

3) Chainage profiles listed as one survey (date) with multiple profiles per file

1	Site	ProfileID	Date	Chainage	Elevation	Flag
2	DUCK	P-100	1998-01-12	0.00	6.30	GPS
3	DUCK	P-100	1998-01-12	12.00	5.85	GPS
4	DUCK	P-100	1998-01-12	24.00	3.62	GPS
5	DUCK	P-100	1998-01-12	36.00	2.25	GPS
6	DUCK	P-100	1998-01-12	48.00	1.13	GPS
7	DUCK	P-100	1998-01-12	60.00	-0.01	GPS
8	DUCK	P-100	1998-01-12	72.00	-1.19	GPS
9	DUCK	P-100	1998-01-12	84.00	-1.86	GPS
10	DUCK	P-100	1998-01-12	96.00	-2.14	GPS
11	DUCK	P-100	1998-01-12	108.00	-2.29	GPS
12	DUCK	P-100	1998-01-12	120.00	-2.26	GPS
13	DUCK	P-100	1998-01-12	132.00	-2.17	GPS
14	DUCK	P-100	1998-01-12	144.00	-2.00	GPS
15	DUCK	P-100	1998-01-12	156.00	-1.88	GPS
16	DUCK	P-100	1998-01-12	168.00	-2.07	GPS

The CCO format includes a column ‘FC’ for Feature Codes, which are defined as follows:

1	ZZ	Unknown
2	S	Sand
3	M	Mud
4	MS	Mud and Sand
5	G	Gravel
5	VG	Vegetated gravel
6	GS	Gravel and Sand
7	GM	Gravel and Mud
8	X	Mixture
8	MX	Mixture
9	SH	Shell
10	BO	Boulders
11	R	Rock
12	SM	Saltmarsh
13	HW	High water mark
14	SD	Sea defence
15	W	Water body
16	G	Grass
17	D	Dune
18	DV	Vegetated dune
19	F	Forested
20	FB	Obstruction
21	CT	Cliff top
21	CL	Cliff line
22	CE	Cliff edge
23	CF	Cliff face
24	SP	Start point of line
24	ZCP	Start point of line

For further details see <http://www.channelcoast.org/>



Shoreline Position

Format is as used for the 2018 ShoreCast exercise (Montano *et al.*, 2020)

Header defines columns (NB date and time are 6 columns unlike most other formats provided for).

1	Year	Month	Day	Hour[NZST]	Min	Sec	Average[n
2	1999	1	2	9	0	0	56.399675
3	1999	1	3	7	30	0	57.672548
4	1999	1	4	6	0	0	58.945422
5	1999	1	4	11	0	0	58.861482
6	1999	1	4	18	0	0	57.973684
7	1999	1	5	7	0	0	56.324917
8	1999	1	5	11	0	0	56.583697
9	1999	1	6	7	0	0	61.708058
10	1999	1	6	12	0	0	62.101505
11	1999	1	7	10	30	0	61.209084
12	1999	1	8	9	0	0	60.316662
13	1999	1	8	14	0	0	60.458741
14	1999	1	9	10	0	0	60.972464
15	1999	1	10	2	40	0	60.536215
16	1999	1	10	19	20	0	60.099966
17	1999	1	11	12	0	0	59.678588
18	1999	1	11	16	0	0	59.593717
19	1999	1	12	12	40	0	59.155216
20	1999	1	13	9	20	0	58.716715



BlueKenue

See Blue Kenue manual for details of format:

http://www.nrc-cnrc.gc.ca/eng/solutions/advisory/blue_kenue_index.html

1) Time series at a point *.ts1

```
#####
:FileType ts1 ASCII EnSim 1.0
# Canadian Hydraulics Centre/National Research Council (c) 1998-2012
# DataType Type 1 Time Series
#
:Application BlueKenue
:Version 3.3.4
:WrittenBy Ian
:CreationDate Fri, Feb 17, 2017 03:04 PM
#
-----
:Name Resampled FREE SURFACE (X 618062.01 Y 5633731.79)
#
:AttributeUnits 1 M
#
:LocationX 618062.009194
:LocationY 5633731.789242
#
:StartTime 2010/01/01 12:00:00.000
:DeltaT 0:15:00.000
#
:EndHeader
1.500000
1.500000
1.500000
1.500000
1.500539
```

2) Time series at a point *.ts2

Can be velocity and direction (MAGDIR), as shown here, or two velocity components (UV):

```
#####
:FileType ts2 ASCII EnSim 1.0
# Canadian Hydraulics Centre/National Research Council (c) 1998-2012
# DataType Type 2 Time Series
#
:Application BlueKenue
:Version 3.3.4
:WrittenBy Ian
:CreationDate Fri, Feb 17, 2017 04:43 PM
#
-----
:SourceFile Solent1.res
:ExtractPointSet Sample point
#
:Name VELOCITY UV (X 618062.01 Y 5633731.79)
:Title VELOCITY UV: Solent1
#
:AttributeUnits 1 M/S
#
:LocationX 618062.009194
:LocationY 5633731.789242
#
:DataDefinition MAGDIR
#
:StartTime 2010/01/01 12:00:00.000
:DeltaT 0:15:00.000
#
:EndHeader
0.000000 0.000000
0.000000 248.689590
0.000000 104.149414
0.000000 345.505493
0.000574 333.952698
0.005124 332.008178
```


3) Time series at a point *.ts3

```
#####
:FileType ts3 ASCII EnSim 1.0
# Canadian Hydraulics Centre/National Research Council (c) 1998-2012
# DataType Type 3 Time Series
#
:Application BlueKenue
:Version 3.3.4
:WrittenBy Ian
:CreationDate Fri, Feb 17, 2017 03:17 PM
#
-----
:Name Value(1) (X 621304.24 Y 5630374.26)
#
:AttributeUnits 1 M
#
:LocationX 0.000000
:LocationY 0.000000
#
:EndHeader
2010/01/01 12:00:00.000 1.500000
2010/01/01 12:15:00.000 1.500000
2010/01/01 12:30:00.000 1.500000
2010/01/01 12:45:00.000 1.500013
2010/01/01 13:00:00.000 1.509651
2010/01/01 13:15:00.000 1.686642
2010/01/01 13:30:00.000 1.480736
2010/01/01 13:45:00.000 0.595087
2010/01/01 14:00:00.000 -0.250650
2010/01/01 14:15:00.000 -0.681176
```

4) Time series at a point *.ts4

Can be velocity and direction (MAGDIR), or two velocity components (UV) as shown here:

```
#####
:FileType ts4 ASCII EnSim 1.0
# Canadian Hydraulics Centre/National Research Council (c) 1998-2012
# DataType Type 4 Time Series
#
:Application BlueKenue
:Version 3.3.4
:WrittenBy Ian
:CreationDate Sun, Feb 19, 2017 11:03 AM
#
-----
:Name VELOCITY UV (X 618291.73 Y 5633143.44)
:Title VELOCITY UV: Solent1 SERAFIN
#
:AttributeUnits 1 M/S
#
:LocationX 618291.732987
:LocationY 5633143.444948
#
:DataDefinition UV
#
:EndHeader
2010/01/01 12:00:00.000 0.000000 0.000000
2010/01/01 12:15:00.000 -0.000000 -0.000000
2010/01/01 12:30:00.000 -0.000000 0.000000
2010/01/01 12:45:00.000 -0.000000 0.000000
2010/01/01 13:00:00.000 -0.000466 0.000647
2010/01/01 13:15:00.000 -0.034492 0.048785
2010/01/01 13:30:00.000 -0.113091 0.158124
2010/01/01 13:45:00.000 0.256879 -0.387983
2010/01/01 14:00:00.000 0.847288 -1.242919
```

Appendix B – Data set properties (DSproperties)

Data are stored in a *dstable*, which extends a Matlab *table* to hold more comprehensive metadata for multi-dimensional data sets. This makes use of a *dsproperties* class object to hold the metadata. The *dstable* and *dsproperties* classes are part of the *dstoolbox*. When loading data or saving model results the DSproperties can be defined, loaded and saved when creating a new Case within the application. These data are used in the application to provide descriptions of variables and dimensions in UIs, define units and formats and generic labels which are used for plots and analysis outputs. Further details of the *dstable* and *dsproperties* classes can be found in the Matlab Supplemental Software Documentation for the *dstoolbox*. An example of the code to load DSproperties for a time series and a variable with 2 spatial dimensions are shown below.

DSproperties for a set of timeseries variables.

```
dsp = struct('Variables', [], 'Row', [], 'Dimensions', []);

dsp.Variables = struct(...
    'Name', {'AvSpeed', 'MaxSpeed', 'Dir'}, ...
    'Description', {'Mean wind speed', 'Maximum wind speed', 'Mean wind direction'}, ...
    'Unit', {'m/s', 'm/s', 'deg'}, ...
    'Label', {'Wind speed (m/s)', 'Wind speed (m/s)', 'Wind direction (deg)'}, ...
    'QCflag', repmat({'raw'}, 1, 3));

dsp.Row = struct(...
    'Name', {'Time'}, ...
    'Description', {'Time'}, ...
    'Unit', {'h'}, ...
    'Label', {'Time'}, ...
    'Format', {'dd-MM-yyyy HH:mm:ss'});

dsp.Dimensions = struct(...
    'Name', {''}, ...
    'Description', {''}, ...
    'Unit', {''}, ...
    'Label', {''}, ...
    'Format', {''});
```

DSproperties for a 2D variable with 2 spatial dimensions.

```
dsp = struct('Variables', [], 'Row', [], 'Dimensions', []);
|
dsp.Variables = struct(...    %cell arrays can be column or row vectors
    'Name', {'u'}, ...
    'Description', {'Transport property'}, ...
    'Unit', {'m/s'}, ...
    'Label', {'Transport property'}, ...
    'QCflag', {'model'});

dsp.Row = struct(...
    'Name', {'Time'}, ...
    'Description', {'Time'}, ...
    'Unit', {'s'}, ...
    'Label', {'Time (s)'}, ...
    'Format', {'s'});

dsp.Dimensions = struct(...
    'Name', {'X', 'Y', 'Z'}, ...
    'Description', {'X co-ordinate', 'Y co-ordinate', 'Z co-ordinate'}, ...
    'Unit', {'m', 'm', 'm'}, ...
    'Label', {'X co-ordinate (m)', 'Y co-ordinate (m)', 'Z co-ordinate (m)'}, ...
    'Format', {'-', '-', '-'});
```



Appendix C - Date and Time Locale

The system was developed to provide rapid access to data from the Channel Coastal Observatory. It was therefore developed to use British date and time formats. On some computers, with a difference system language, the date and time formats will be different and this can result in some problems with the “Locale” settings. One way to get around this issue is to change the date and time settings in the “Time & Language” section of Settings, or in the “Clock and Region” section in the Control Panel. Change the “Country or region” to United Kingdom. And then click “Additional date, time & regional settings” under “Related settings”; in the “Clock and Region” dialog box select “Change date, time or number formats”, and set the format to “English (United Kingdom)” and other date/time formats are as shown in the picture below.

[Language preferences](#)

Date and time formats	
Short date:	dd/MM/yyyy
Long date:	dd MMMM yyyy
Short time:	HH:mm
Long time:	HH:mm:ss
First day of week:	Monday