Preface

CSTmodel Matlab™ App to compute the mean tide level, high and low water levels, tidal velocity amplitude and river velocity along the length of a convergent estuary using the analytical model of Cai, Savenije and Toffolon (hence the CST model).

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. CSTmodel uses the muitoolbox and dstoolbox.

Resources

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

Cite as:

Townend, I.H., 2021, CSTmodel manual, CoastalSEA, UK, pp23, www.coastalsea.uk.

Bibliography

Cai H, 2014, A new analytical framework for tidal propagation in estuaries, TU Delft, TU Delft.

Acknowledgements

The provision of the Matlab™ code by Huayang Cai, SunYat-Sen university, China is gratefully acknowledged.

Revision history

Version	Date	Changes
2.0	Oct 2021	Established as an App using the muitoolbox
1.0	Jul.2018	First release via www.coastalsea.uk

Contents

1 Introduction

The CSTmodel App computes the mean tide level, high and low water levels, tidal velocity amplitude and river velocity along the length of a convergent estuary using the analytical model of Cai, Savenije and Toffolon (hence the CST model) as described in (Cai, 2014; Savenije, 2012).

2 Getting started

2.1 Configuration

CSTmodel 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: CSTmodel.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:

```
>> CSTmodel;
```

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

Documentation can be viewed from the 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.

2.2 Model Set-up

File>New to create a new project space.

Setup>Model Parameters: The UI requests data for the model variables. Once added the current set of variables can be viewed using the *Inputs* tab.

Setup>Run Parameters: The UI requests data for the model run time variables. Once added the current set of variables can be viewed using the *Inputs* tab.

Run> Run model

When the run has completed the user is prompted to provide a description of the model run (scenario).

The run is listed on the *Cases* tab and the tidal elevations for the most recent run can be viewed on the *Plot* tab.

Plot>Plot menu

The results from a run can be selected and plotted. By using the Add button additional model runs can be included on the plot, allowing different Cases to be compared.

2.2.1 Input parameters

In the model the CSA and width are defined by a convergent exponential and prismatic section defined by:

$$a = a_r + (a_0 - a_r)exp\left({}^{-x}/_{L'}\right)$$

where a0 is the CSA (or width) at the estuary mouth, a_r is the river CSA (or width) and L′ is the exponential convergence length.

The distance to the estuary-river switch provides the option to define different friction and storage ratios for the reaches up and downstream of this distance. For a single switch point there will be a single value for the ‘distance from the mouth to the switch point’ and 3 values for both the Manning friction coefficient and the Storage width ratio (one at the mouth, one at the switch point and one at the head). If more detailed mapping of friction is required, then the number of switch points should be 2 less than the number of friction and width ratio values specified. Eg if there are 3 switch points then there should be 5 values of friction and width ratio.

3 Application Menus

The UI comprises a series of drop down menus that provide access to a number of commonly used functions such as file handling, management of run scenarios, model setup, running and plotting of the results. In addition, Tabs are used to display set-up information of the Cases that have been run. In this manual text in *Red italic* refers to drop down menus and text in *Green italic* refers to Tab titles.

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 models are saved as *.mat files. User selects a model 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 *Cases* tab.

Tools>Clear all>Project: deletes the current project, including setup parameters and all Cases.

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

Tools>Clear all>Cases: deletes all cases listed on the *Cases* tab but does not affect the model setup.

3.3 Project

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

Project>Cases>Edit Description: select a scenario description to edit.

Project>Cases>Edit Data Set: edit a data set. Initialises a data selection UI to define 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).

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

Project>Cases>Delete: select the Case(s) to be deleted from the list of Cases and these are deleted (model setup is not changed).

Project>Cases>Reload: select a previous model run and reload the input values as the current input settings.

Project>Cases>View settings: display a table of the model input parameters used for a selected model run (Case).

Project> Import/Export>Import: load a Case class instance from a Matlab binary 'mat' file. Only works for data sets saved using Export.

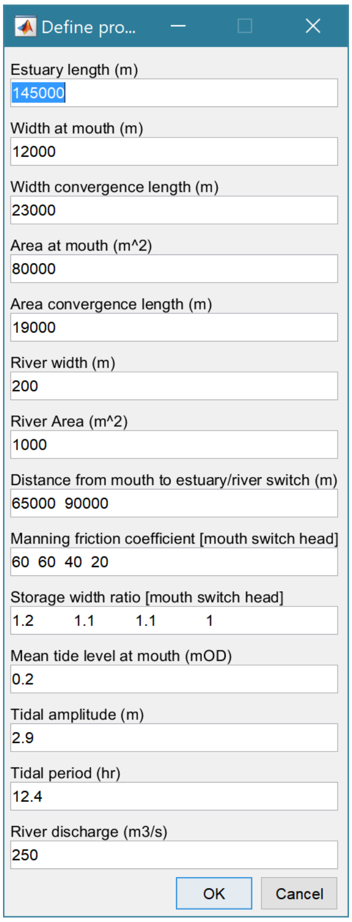
Project>Import/Export>Export: save a Case class instance to a Matlab binary 'mat' file.

These last two functions can be used to move Cases 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.

3.4 Setup

The setup menu provides a series of menus to enable different components of the model to be defined.



Setup>Model Parameters: calls UI to input or edit the model variables. Once added the current set of variables can be viewed using the *Inputs* tab.

Total length of channel to be represented

Mean tide width at the mouth

Rate of width convergence

Mean tide cross-sectional area at the mouth

Rate of CSA convergence

Width of the river channel

Cross-sectional area of the river channel

Distance from mouth to points at which the properties below are adjusted. These are supplemented by the mouth and head distances so the number of the following properties should be number of distances+2.

Manning friction coefficient as a set of linear along channel variations

The storage width ratio is the ratio of the width at high and low water and is also taken to vary linearly based on the defined values at the switch points.

Tidal period

River discharge

Setup>Run Parameters: calls UI to input or edit model run time variables. Once added the current set of variables can be viewed using the *Inputs* tab.

Setup>Estuary Properties: prompts the user to select a file containing the estuary form properties from a text file. The file format is a single header line followed by 5 columns of numeric data, for distance from mouth, area at mean tide level, width at high water, width at low water, and Mannings N. Further details on file format are provided in Section 5.1

Setup>Import data> Load data: prompts for file format to be loaded. The options available vary with Data type and then loads the data and prompts for a description (working title) for the data set. The UI first prompts the to load the Along-channel properties file. This is followed by the option to load a water levels file, with along-channel water levels at intervals over a tidal cycle, and a velocity file, with along-channel velocities at intervals over a tidal cycle. The properties file is required, whereas the water levels and velocities files are optional. Further details on the file formats used are given in Section 5.2 and 5.3.

Setup>Import data > Add data: prompts for file to be added (only one file at a time can be added) and the Case to use (if more than one Case). Only files with the format used to create the data set can be used to Add data to a data record and this is selected when the first file is loaded using the Load menu option. (NB: not used for Imported Data in this App).

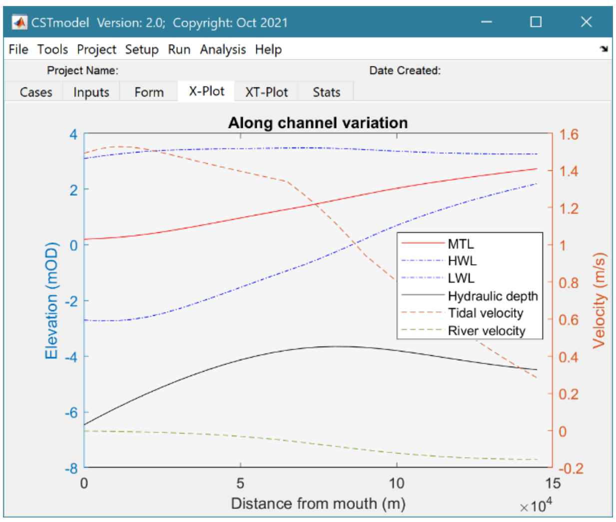
Setup>Import data > Delete data: prompts for Case from which some part of the data is to be deleted.

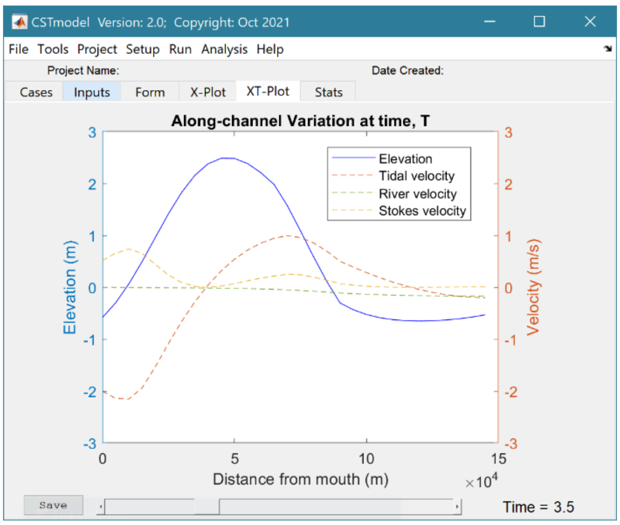
Setup>Import data > Data QC: runs a series of checks on the data. This is only available if defined for the specific data format.

Setup>Input Data>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.

3.5 Run

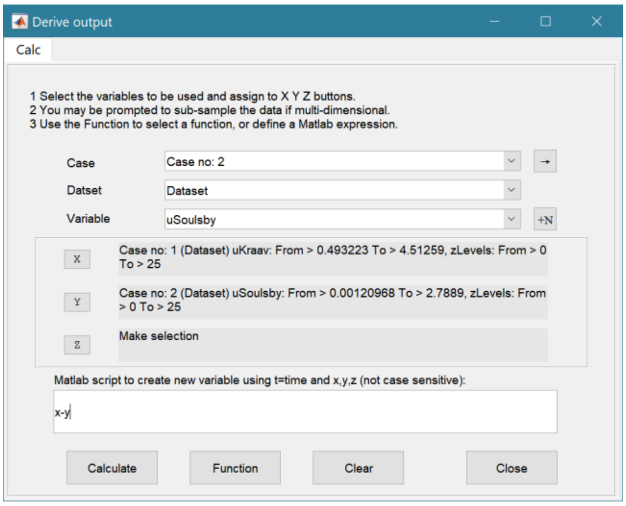
Run> Run Model: runs model, prompts for Case description which is added to the listing on the *Cases* tab and the results can be displayed and plotted from the X-Plot or XT-Plot tabs.





Use the '>Figure' button to create a stand-along figure of the X-Plot and the 'Save' button to save the animation to a file for the XT-Plot.

Run> Derive Output: data that has been added (either as data or modelled values) can be used to derive new variables. The UI allows the user to select data and use a chosen selection of data/variable/range to define either a Variable, XYZ dimension, 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 3.9 for details of how this works. An equation is then defined in the text box below using the x, y, z or t variables[1]. 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. Comments can be used to pass additional instructions, such as the inclusion of the RowNames in the output to be saved as a new dataset, using either %time or %rows.

For example, any of the following could be entered into the equation box:

```
x.^2+y   %time
myfunction1(x,y,t,'usertext')
myfunction2(x,mobj)
```

The output from function calls can be figures or tables, a single numeric value, or a dataset to be saved (character vector or array). External functions must return output as a cell array with the new variable in the first cell and data to be used to define RowNames in the second cell. If the %time or %rows instruction is included in the call, row data are added providing that the length of the input dataset matches the output dataset. If there is no output to be passed back the function should return a cell array containing the string 'no output' to suppress the message box, which is used for single value outputs. For expressions that return a result that is the same length as one, or more, of the variables used in the call, there is also the option to add the variable to one of the input datasets as a new variable. In all there are three ways in which results can be saved:

1. Expression or function returns a result that is the same number of rows as one or more of the input datasets. Option to (a) add as a new variable to an existing data set, or (b) create a new dataset with no assignement to the RowNames property.
2. As (1) with the comment of %time or %rows in the call. Attempts to use the RowNames property of one of the inputs to define RowNames in a new dataset. Requires the input variables to have same number of rows. (Plan to add interpolation so this may change).
3. Expression or function returns a result with a new variable and time time in a 2-element cell array. The variable is saved as a new 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 can be selected but they are defined in the call using dst, for example:

[1] 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.

myfunction3(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 cell array containing one or more data sets. The options for saving the data are the same, with the additional option that when a dstable, or struct of dstables, is returned, these are saved directly and it is assumed that the dsproperties have been defined in the function called.

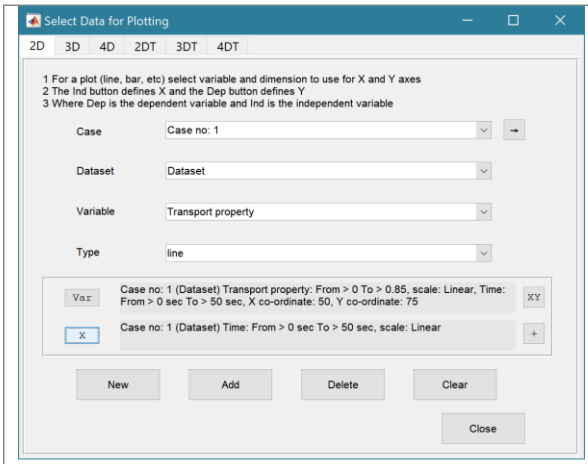
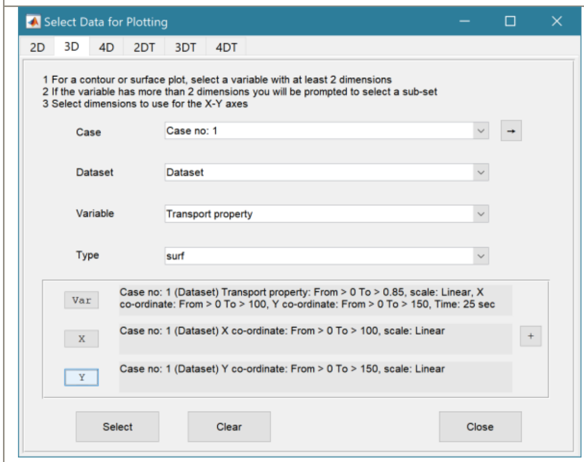
Some further details on using this option and the '**Function**' library available are provided in Section 4.2.

3.6 Analysis

Plotting and Statistical Analysis both use the standard Data selection UI. These both require Case, Dataset and Variables to be selected from drop-down lists and assigned to a button. Further details of how this works are given in Section 3.9.

3.6.1 Plotting

Analysis>Plot menu: initialises the Plot UI to select variables and produce several 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. There are then buttons to create a New figure, or Add, or Delete variables from an existing figure for 2D plots, or simply a Select button for 3D and 4D plots. The following figures illustrate the options available.

	**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) <u>Control Buttons</u>: → : 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. <u>Control Buttons</u>: see 2D plot above.

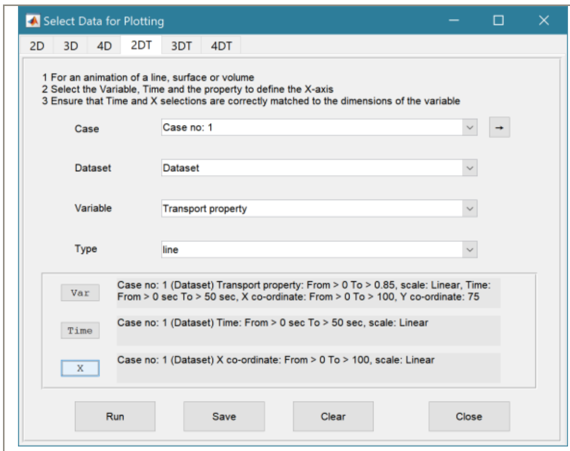
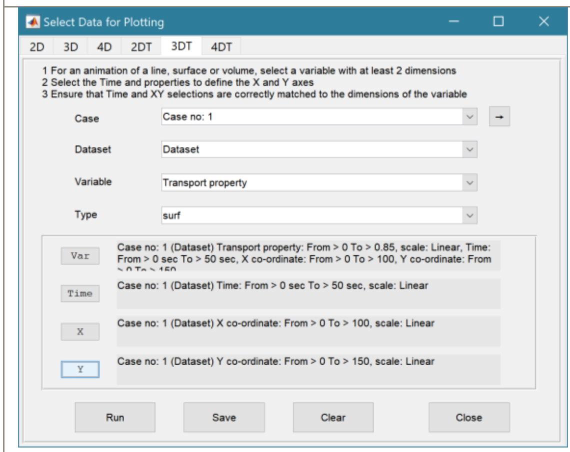
For all plot types, when the data has more dimensions than the plot or animation the user is prompted to sub-select from the data (by selecting sampling values for the dimensions that are not being used).

Animations follow a similar workflow. There are buttons at the bottom of each tab to:

Run the selection and create an animation,

Save the animation to a file (the animation needs to have been run first) . There is also an option to save on the bottom left of the animation figure.

Clear the current selection.

	**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) <u>Control Buttons</u>: → : 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. <u>Control Buttons</u>: see 2DT plot above.

Selection of User plot type

Calls the user_plot.m function, where the user can define a workflow, accessing data and functions already provided by the particular App or the muitoolbox. The sample code can be found in the psfunctions folder and illustrates the workflow to a simple line plot using x-y data from the 2D tab and a surface plot using x-y-z data from the 3D tab.

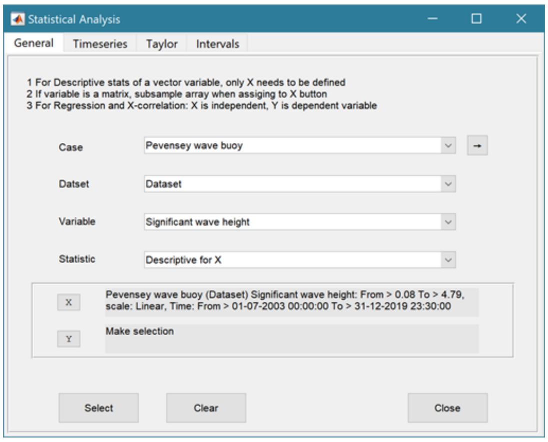
3.6.2 Statistics

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.

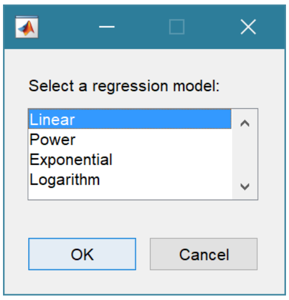
General tab

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

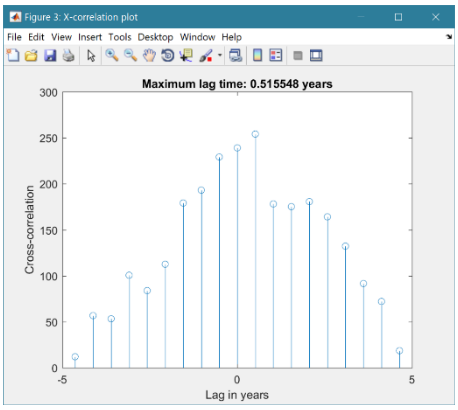
1) **Descriptive for X**: general statistics of a variable (mean, standard deviation, minimum, maximum, sum and linear regression fit parameters). Only X needs to be defined. The range of the variable can be adjusted when it is assigned to the X button (see Section 3.9). If the variable being used is 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 regresion plot of the dependent variable, Y, against the independed 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-corrleation 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.

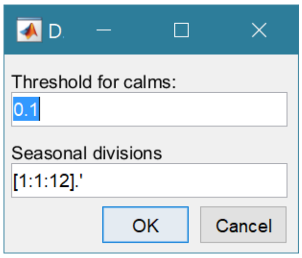
4) **User**: calls the function user_stats.m, in which the user can implement their own analysis methods and display results in the UI or add output to the project Catalogue. Currently implements an analysis of clusters as detailed for Timeseries data below.

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.

   For option 3, the separation between peaks ('*tint*') is also be defined in the pop-up gui. 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) **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. 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.

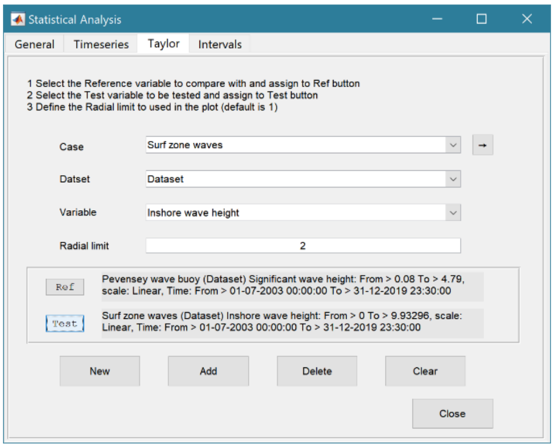
Taylor tab

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).

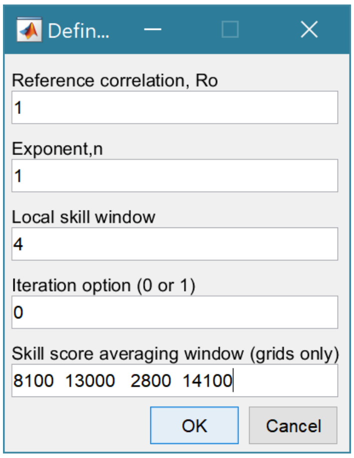
[*The ModelSkill App provides additional tools to test data and the ModelSkill App manual provides further details of the methods used*.]

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).

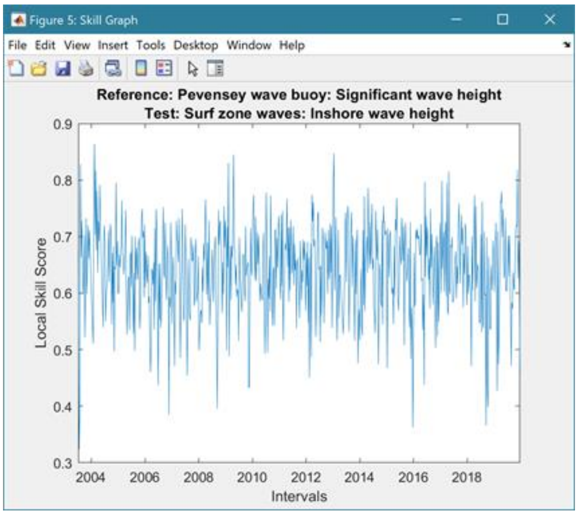
Exponent used in computing the skill score.

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.

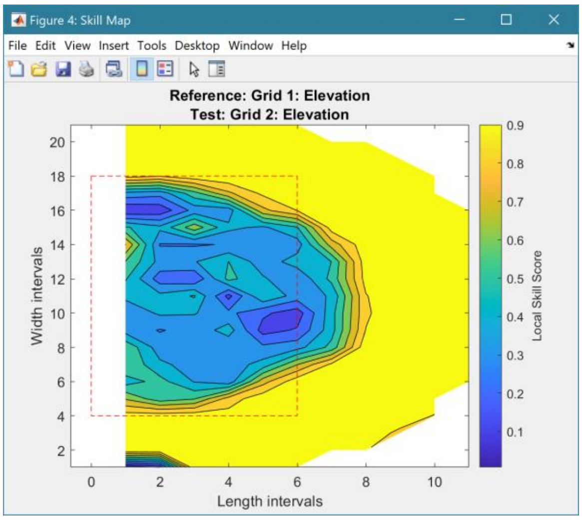
Local skill score is computed for window around every grid cell (=1), or computes score for all non-overlapping windows (=0)

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

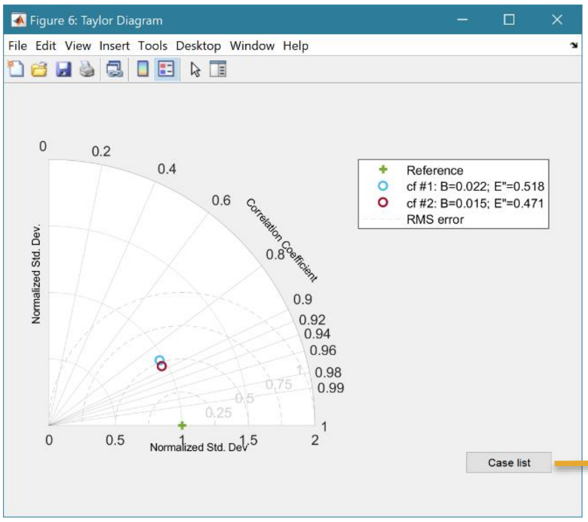
(a) time series skill score plot



(b) grid skill score plot

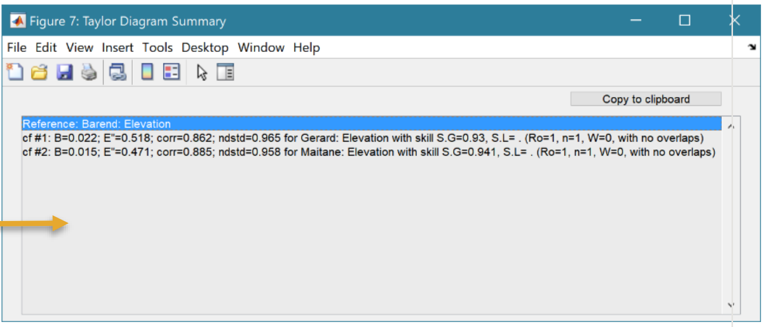


The Taylor Plot shows the Reference point as a green cross and the Test points as coloured circles. The legend details the summary statistics and the Case List button generate a table figure listing all the results. These can be copied to the clipboard.



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, Sg, and the average local skill score, Sl.



3.7 Help

The help menu initialises the App documentation in the Matlab™ Supplemental Software documentation.

3.8 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.

Cases: lists 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 and any source information (files input or models used).

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

Form: plot the estuary form properties defined for the model and/or imported from a user file.

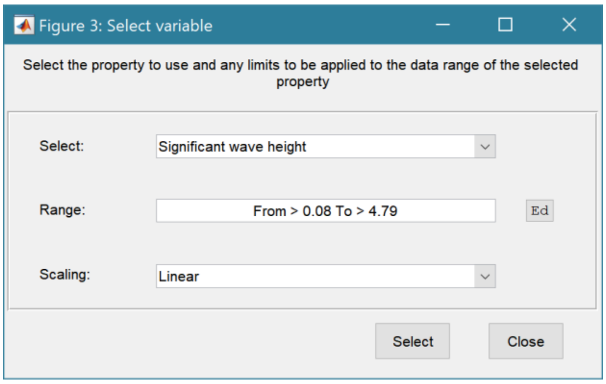
X-Plot: displays a plot of the along channel output properties for the selected case (display only).

XT-Plot: displays a plot of the time varying properties at a selected distance along the estuary. (display only).

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

3.9 UI Data Selection

Functions such as Derive Output (3.5), Plotting (3.6.1) and Statistics (3.6.2) 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 (eg 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) is also selected on this UI. The selection is defined in two steps:



Step 1.

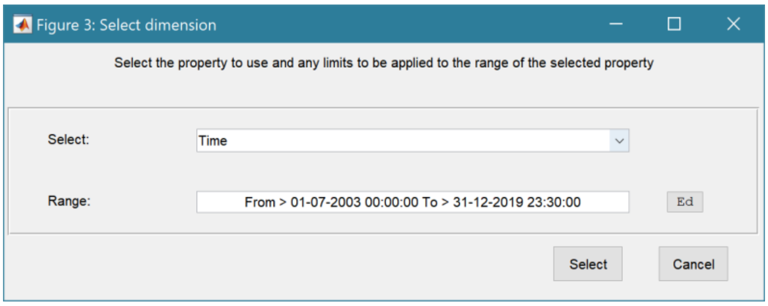
Select the attribute to use. This can be the variable or any of its associated dimensions, which are listed in the drop-down list.

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.

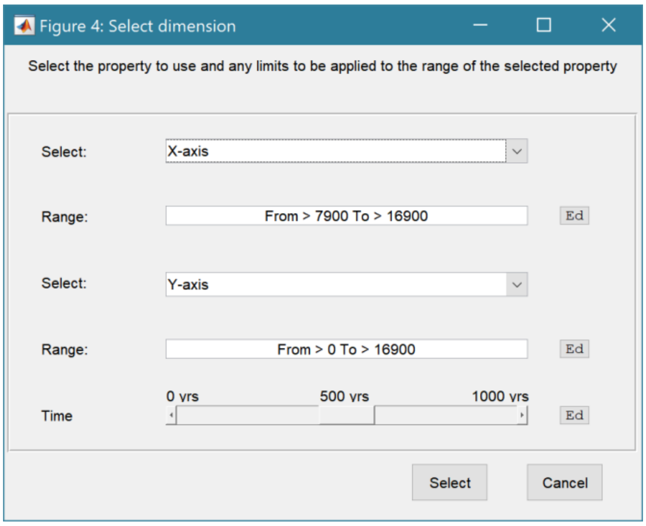
The number of variables listed on the UI depends on the dimensions of the selected variable. For each one Select the attribute to use and the range to be applied.



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.

4 Supporting Information

4.1 Basis of the model

The model adopted is the solution proposed by Cai, Savenije and Toffolon, which is well suited and has already been applied to several estuaries (Cai, 2014). The model accounts for the influence of river discharge on tidal wave propagation and the along channel variation of mean water level. The analytical solutions are obtained by solving a set of four implicit equations, i.e. phase lag equation, scaling equation, celerity equation, and damping equation, derived from the conventional 1-dimensional conservation of mass and momentum equations (Savenije, 2012). In addition, the resultant residual water level, due to nonlinear friction, has been taken into account by adopting an iterative procedure. The model reproduces the main tidal dynamics (i.e. tidal amplitude, velocity amplitude, wave celerity, and phase lag between elevation and velocity) along the estuary axis. This is referred to here as the CST model. For a full description of the model, readers can refer to Cai et al. (2014) and related papers (Cai, 2014; Cai et al., 2012a; Cai and Savenije, 2013; Cai et al., 2016; Cai et al., 2012b; Cai et al., 2013). An application with a more detailed numerical model is provide by Zhang et al (2016).

4.2 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 3.5). 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,utext,mobj) ',usereqn]); %handle to anonymous
function
var = heq(t,x,y,z,utext,mobj);
```

This function is then evaluated with the defined variables for *t, x, y,* and *z* and optionally *utext* and *mobj*. *utext* allows text string to be passed (any string enclosed in single quotes, e.g. 'Test') and *mobj* passes the handle to the main UI.

Some functions may alter the length of the reference co-ordinates (x, y, z, t), or return more than one variable. These can be handled by passing a comment appended to the function definition e.g.

`subsample(x,t, thr,mobj) %time`. In the current version of the toolbox, only '*time*' is handled as a key word and this allows functions to derive a new timeseries with different time input values to those of the input variable(s).

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 3.5). The variables selected are sub-sampled based on any adjustments made when each variable is assigned to a button. The dimensions of the vector or array with these adjustments applied need to be dimensionally correct for the function being called.

Equations can use functions such as 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 derived 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.

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.

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.

4.2.1 Calling an external function

This interface 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 CSTmodel are detailed in the next Section.

The input variables for the function must match the syntax used for the call from the Derive Output UI. In addition, functions must return output as a cell array. This can be a single value, a new variable vector or array or a time series with one or more variables. If *time* is included then this should be in the first cell with variables in the subsequent cells.

If the %time or %rows instruction is included in the call, row data are added providing that the length of the input dataset matches the output dataset. If there is no output to be passed back, the function should return a cell array containing the string 'no output' to suppress the message box which is used for single value outputs.

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 but they are defined in the call using dst, for example:

`mufunction3(dst, 'utext', mobj)` where '*utext*' and *mobj* are options as defined above.

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 cell array containing one or more data sets.

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.

4.2.2 Pre-defined functions

The following examples are provided within CSTmodel, 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 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 ca 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. ***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.

4. ***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).

5 Input file formats

All input functions use ASCII text files.

5.1 Along-channel Model Input Properties

Properties to define the along-channel variation in cross-sectional area and width are loaded using *Setup>Estuary Properties* with a file that 1 header line and 4 or 5 columns (including the Manning coefficient is optional). The columns are: x - Distance from mouth, Amtl - Area at mean tide level, Whw - Width at high water, Wlw - Width at low water, Manning_N - Mannings N (optional).

Humber_raw.txt | **demo_estuary_props.txt** | demo_Xobs.txt | demo_Uobs.txt

```
Humber 2000 measured: x Amtl Whw Wlw Manning_N(optional)
0	84244.29	16002.29132	10478.37793
1498.7	92307.06	10568.45376	6373.531979
4190.59	77367.74	11332.31847	6425.803482
6623.75	64617.04	11647.56352	6340.4899
9850.53	54393.19	8182.219804	6260.121396
11320.93	49358.8	5843.379911	4931.790771
13579.51	43051.88	4104.755249	3292.223828
16112.08	35487.28	3976.445614	3056.229749
19042.15	34605.09	4097.990966	2875.337093
21475.31	31697.93	4332.578626	3128.883136
24127.11	27350.87	4313.519826	3083.735757
```

5.2 Along-channel Observed Properties

Observed data to be used to evaluate model results is loaded using *Setup>Import Data>Load* menu option. The along-channel properties file format has a 1 line header and 5 columns of data for: dist - along channel distance, msl - mean water level, z-amp - tidal amplitude, u-amp - velocity amplitude, hyd-depth - hydraulic depth.

Humber_raw.txt | demo_estuary_props.txt | **demo_Xobs.txt** | demo_Uobs.txt

```
dist	msl	z-amp	u-amp	hyd-depth
0.00	0.217	2.869	1.059168694	13.59436663
1.50	0.280	2.830	0.878607182	8.112766743
4.19	0.283	2.848	0.859929251	7.222529873
6.62	0.308	2.867	0.8908874	6.373117664
9.85	0.325	2.875	0.909520327	7.80950323
11.32	0.328	2.887	0.99013511	9.276226273
13.58	0.334	2.896	1.044568372	8.598338326
16.11	0.321	2.931	1.201749586	9.370815949
19.04	0.372	2.912	1.161558969	8.969696734
21.48	0.386	2.922	1.166814125	9.309230543
24.13	0.408	2.935	1.229040224	8.152271237
```

5.3 Velocity and Elevation Observations

The format for the Elevation and Velocity files are loaded using *Setup>Import Data>Load* menu option when loading the Along-channel properties. The files have a 1 line header, 1 column for time, and N columns for elevation or velocity at each time interval, where N is the number of distances included in the along-channel properties file (see above). An example of the first 1.5 hours at 0.25-hour intervals is shown below where there are 32 columns (text is wrapped to fit in display window).

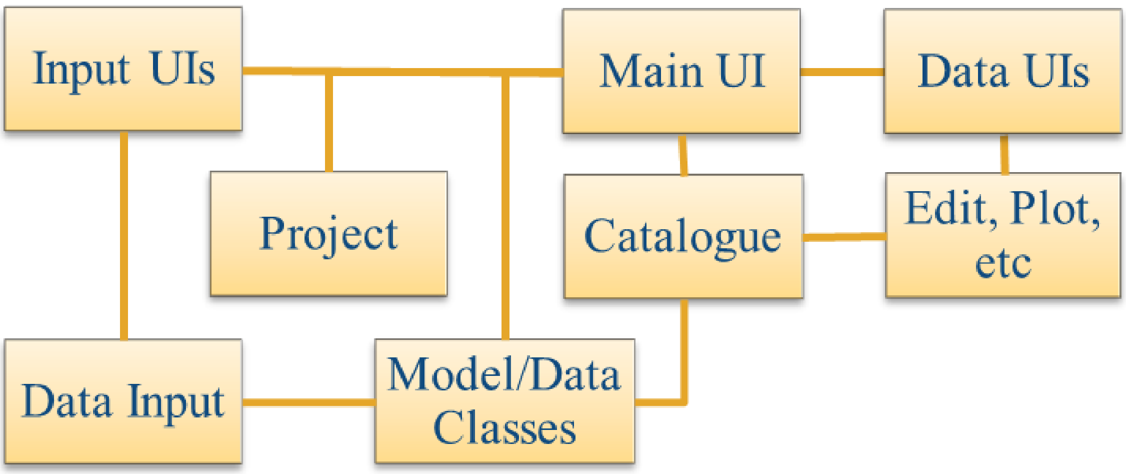
Humber_raw.txt | demo_estuary_props.txt | demo_Xobs.txt | demo_Uobs.txt

```
Velocity
0.25	0.769134245	0.619739443	0.591014743	0.314295733	1.186268646	0.772932472	0.911687966	
0.529385972	0.669711082	1.005391114	0.749394437	0.77485393	0.675135358	0.380945355	1.150127099	
0.931788939	0.734149877	0.631133899	0.79351887	0.592535633	0.602870319	0.632055715	0.548954874	
0.817613133	0.684149789	0.517716359	0.654057518	0.706195211	1.165749941	1.213062554	0.571339612	
0.58537505
0.5	0.769521232	0.620076179	0.590375421	0.31434531	1.186606945	0.772763129	0.911819985	
0.529438516	0.66966019	1.006660841	0.749712369	0.774744024	0.676383967	0.381574838	1.150667477	
0.932271371	0.734577195	0.631982355	0.794392247	0.59301854	0.602878067	0.632008003	0.548369041	
0.817019028	0.683867156	0.516233529	0.65116105	0.703580557	1.160423004	1.20634187	0.568370242	
0.581948554
0.75	0.769423397	0.620080642	0.590821908	0.314353088	1.186990398	0.773004454	0.911848868	
0.529349001	0.669656609	1.006383621	0.750956726	0.775090002	0.674615762	0.381188842	1.150915177	
0.932537238	0.734696219	0.631659366	0.794500146	0.593267538	0.60275928	0.63165755	0.547943647	
0.815702294	0.681566213	0.51444867	0.648675833	0.699901843	1.154127265	1.199605215	0.565158225	
0.579418627
1	0.769460421	0.620548032	0.590853878	0.314515143	1.186595396	0.772880557	0.912385215	
0.52933236	0.669596678	1.006787563	0.750173526	0.775059037	0.67618953	0.381571526	1.151558261	
0.933059707	0.735286877	0.63232233	0.79436725	0.593019361	0.60206134	0.631286859	0.546575204	
0.812799778	0.679482878	0.512403334	0.645753337	0.696976442	1.149651599	1.19368492	0.563349464	
0.577512698
1.25	0.769586735	0.620659495	0.591606516	0.314442207	1.186968943	0.77321451	0.912346165	
0.52939756	0.669317663	1.007119232	0.750898872	0.77515075	0.676977567	0.38196426	1.15235593	
0.933456334	0.735306438	0.631924765	0.794559442	0.592713397	0.601606286	0.629558234	0.545558903	
0.810902009	0.676768962	0.510296028	0.643297759	0.694204671	1.145695127	1.188122789	0.561655479	
0.576599869
1.5	0.77008247	0.620430378	0.590745681	0.314792525	1.187145307	0.773593043	0.912654135	
0.529711095	0.670032678	1.007466137	0.751489213	0.775737316	0.676520854	0.381879268	1.152395031	
0.932548952	0.735554082	0.632009711	0.794028033	0.591698776	0.600172974	0.628351109	0.543582715	
0.807755197	0.674762871	0.50897984	0.641265136	0.691505437	1.141327947	1.18648271	0.560170805	
0.575633976
```

6 Program Structure

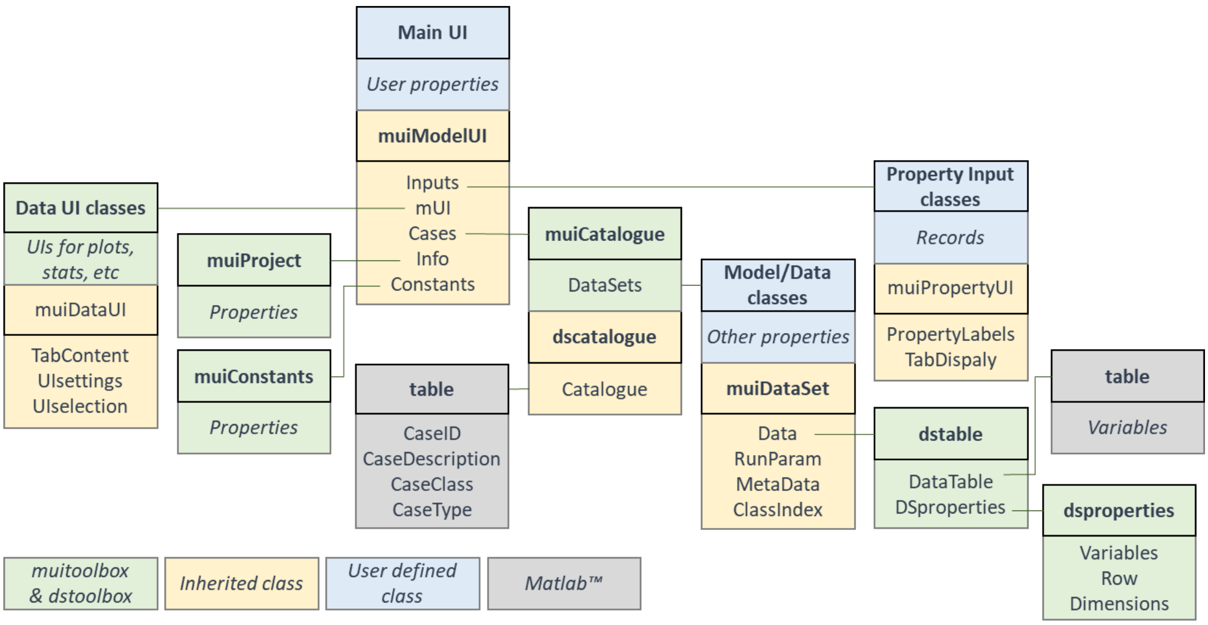
The overall structure of the code is illustrated schematically in Figure 1. This is implemented through several classes that handle the graphical user interface and program workflows (Main UI) and several classes that handle the data manipulation and plotting (Input UIs and Data UIs).

Figure 1 - – High level schematic of program structure



The interfaces and default functionality are implemented in the CSTmodel App using the following muitoolbox classes depicted in Figure 2, which shows a more detailed schematic of the program structure. See the muitoolbox and dstoolbox documentation for more details.

Figure 2 – schematic of program structure showing how the main classes from muittoolbox and dstoolbox are used



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

CSTparameters – class to handle input for model parameters

CSTrunparams – class to handle input for model run time parameters

CSTformprops – class to load estuary form properties from a text file

CSTrunmodel – class to handle call to model, tab plot and output definition *cst_model* – CST model code, provided by Cai HuaYang wit the following functions:

f_new_2012 – analytical solution for tidal dynamics proposed by Cai, H., H. H. G. Savenije, and M. Toffolon, 2012, Journal of Geophysical Research, 117, C09023

f_toffolon_2011 – analytical solution for tidal dynamics proposed by Toffolon, M., and H. H. G. Savenije (2011), J Geophys Res-Oceans, 116,

findzero_new_discharge_river – analytical solution dimensionless parameters for river flow

findzero_new_discharge_tide – analytical solution dimensionless parameters for tide

newtonm – Newton Rapshon root finder

7 Bibliography

Cai, H., 2014. A new analytical framework for tidal propagation in estuaries. PhD, TU Delft, TU Delft, 187 pp.

Cai, H., Savenije, H., Yang, Q., Ou, S., Lei, Y., 2012a. Influence of River Discharge and Dredging on Tidal Wave Propagation: Modaomen Estuary Case. Journal of Hydraulic Engineering, 138(10), 885-896.

Cai, H., Savenije, H.H.G., 2013. Asymptotic behaviour of tidal damping in alluvial estuaries. Journal of Geophysical Research: Oceans, DOI 10.1002/2013JC008772.

Cai, H., Savenije, H.H.G., Jiang, C., Zhao, L., Yang, Q., 2016. Analytical approach for determining the mean water level profile in an estuary with substantial fresh water discharge. Hydrology and Earth System Sciences, 20(3), 1177-1195. [10.5194/hess-20-1177-2016]

Cai, H., Savenije, H.H.G., Toffolon, M., 2012b. A new analytical framework for assessing the effect of sea-level rise and dredging on tidal damping in estuaries. Journal of Geophysical Research, 117(C09023).

Cai, H., Savenije, H.H.G., Toffolon, M., 2013. Linking the river to the estuary: influence of river discharge on tidal damping. Hydrol.Earth Syst.Sci.

Coles, S., 2001. An Introduction to Statistical Modeling of Extreme Values. Springer Series in Statistics. Springer-Verlag, London.

Savenije, H.H.G., 2012. Salinity and tides in alluvial estuaries. Elsevier, Amsterdam, Netherlands.

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]

Zhang, M., Townend, I.H., Zhou, Y.X., Cai, H.Y., 2016. Seasonal variation of river and tide energy in the Yangtze estuary, China. Earth Surface Processes and Landforms, 41, 98-116. [10.1002/esp.3790]