Preface

SedTools is a prototype toolbox for various forms of sediment analysis

Requirements

The model is written in MatlabTM and provided as Open Source code (issued under a GNU General Public License). The model requires the ModelUI package.

Resources

The ModelUI zip file contains folders for the core ModelUI package and can be downloaded from [www.coastalsea.uk](http://www.coastalsea.uk). The SedTools zip file is provided as a plug-in package for ModelUI. In addition, the sub-folder contains the user manual and the software licence.

Bibliography

# Acknowledgements

Settling analysis is based on the Matlab function cal\_settling developed by Urs Neumeier, 2005

Revision history

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 2.0 | Feb 2021 | Ported to work in muitoolbox |
| 1.0 | Feb.2019 | First release via www.coastalsea.uk |

Contents

[Acknowledgements i](#_Toc2178549)

[1 Introduction 1](#_Toc2178550)

[2 Getting Started 1](#_Toc2178551)

[2.1 Configuration 1](#_Toc2178552)

[2.2 Model Set-up 1](#_Toc2178553)

[3 Menus and Tabs 2](#_Toc2178554)

[4 Background to Analysis Methods 3](#_Toc2178555)

[5 Program Structure 3](#_Toc2178556)

[6 Bibliography 5](#_Toc2178557)

# Introduction

SedTools is a toolbox of for the analysis of sediments …..

This currently includes functions to:

* analyse the data from a settling column and derive sediment grain-size distributions

# Getting Started

## Configuration

SedTools model makes use of the ModelUI package. There are several ways of setting this up, including:

1. Place the ModelUI and SedTools folders under the same root folder (e.g MUImodels). Add the path for the ModelUI and MUIfunctions folder to the search path:

>>addpath(‘..\ModelUI’,*‘..\MUIfunctions’*)

and with the SedTools folder as the working folder, run the model:

*>>SedTools;*

The file runsedtools.m contains the above as a script, which is run from the prompt line:

*>> runsedtools;*

1. From the Matlab **Home** tab, in the **Environment** section, click **Set Path**. The Set Path dialog box appears, and you can add the required paths.
2. Put the ModelUI and Channel Form model files in the same working folder (not recommended)

If the model is setup using (ii) or (iii) then run the model using:

*>>SedTools;*

To remove ModelUI and other folders from the search path when you have finished you can use the following:

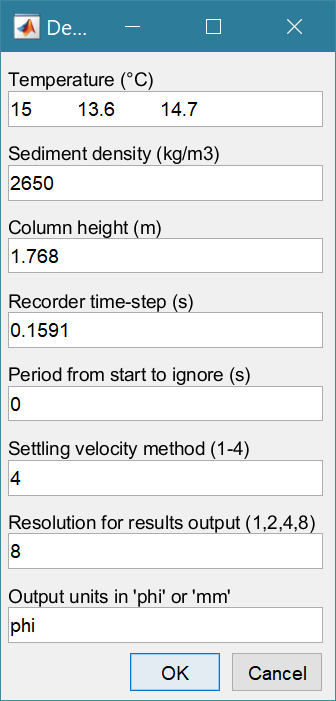
*>> rmpath(‘<ModelUI path>’, ‘<MUIfunctions path>’’);* where <ModelUI path> is the path to the ModelUI folder, etc.

Using the above the model must be run with the SedTools folder as the working folder. If you plan to run other models and move folders while working, there is a utility in the Utilities sub-folder called addModelUIpaths.m[[1]](#footnote-1). This sets the required paths for all ModelUI models. These are removed at the end of each MatlabTM session, or can be removed using the utility delModelUIpaths.m, which is also in the Utilities sub-folder. To make these permanent path changes, see “Add Folders to the MATLAB Search Path at Startup” in the MatlabTM help documentation.

## Model Set-up

*File>New* to create a new project space.

*Setup>Settling Parameters*: Allows the inputs required for the settling column analysis to be defined along with various options to control the output. Once added the current set of variables can be viewed using the *Settling* tab.



Temperature during the settling experiment (\*).

Density of the sediment used (\*).

Height of the settling column (\*).

The time step used to record the settling data

The period from the start of the recording to omit from the analysis

Computation method for settling velocity (+)

The resolution to be used in the output (values are 1/phi)

Units to display output

(\*) If multiple data sets are being averaged, multiple values can be entered, with the number of values equal to the number of input files. Otherwise the first value is replicated for all data sets.

(+) Methods: 1 = Gibbs et al. (1971); 2 = Baba and Komar (1981); 3 = van Rijn (1993); 4 = Soulsby (1997).

*Run>Run Settling Analysis*: Runs the settling analysis. The user is prompted to name the output, the default being the input file name.

*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 scenarios to be compared.

*Plot>Output Definitions*: provides the ability to edit the Output Definitions used to define output variable names and labels for each model or data set. This edits the model or data set class property ResDef within the project but does not change the class definitions.

The *Plot* tab provides a summary plot of the output as marginal and cumulative frequency curves, along with a statistical summary table. The statistical data can be copied to the clipboard using the button below the table.

# Menus and Tabs

# Background to Analysis Methods

Cal\_settling was written to process the data of the settling column of the School of Ocean and Earth Science, Southampton Oceanographic Centre, University of Southampton.

* The input data must be the readings of the scale (units do not matter) recorded with a fixed time-step. The time-step MUST be written as a constant in the routine. The input data can either be given as a text file with one weight record per line (and nothing else), or as Matlab array.
* The routine can use 4 different formulae for converting the settling velocity into grain size: Gibbs et al. (1971), Baba and Komar, (1981), van Rijn (1993) and Soulsby (1997). The used formula is selected by a constant in the routine. It is recommended to use the formula of Soulsby (1997).
* The results are given in Phi units, by default with 1/8 Phi resolution, but other resolutions can be selected.
* It is possible to average together several analyses, for example replicates produced from sub-samples, by giving the multiples filenames in a cell array of strings. Different temperatures or settling heights can be specified. It is strongly recommended to FIRST process each replicate separately and to examine the quality and the similarity of the results BEFORE averaging the replicates of one sample.
* Cal\_setting produces a figure representing the grain-size frequencies and the cumulative frequency distribution (see example below). The statistical parameter (either from the graphical method or the moment method, selection through a constant in the routine) can be printed on this figure. The figure drawing can be deactivated with an option. This figure can be printed from Matlab, or copied-pasted from Matlab into another program.
* With a particular option, the routine copy the numerical results in the system clipboard, so that they can be pasted into another program like Excel.
* The routine can also return the settling velocities for a given temperature and grain-size density.
* The settling column in Southampton often records a slight mass increase in the first seconds, which does not correspond to settling grains but represent some noise. This produces a peak in the coarsest grain-size class. To remove this effect, use the option "ixx", which ignore the xx first seconds (example: "i08" for 8 seconds).

See source of core program and further details at: <http://neumeier.perso.ch/matlab/cal_settling.html>

# 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 GUI) and several classes that handle the data manipulation and plotting (Data GUIs). The interfaces and default functionality are implemented in the ModelUI package using the following classes:

|  |
| --- |
| *Figure 1 – High level schematic of program structure* |
|  |
| ***GUIinterface*** – basic functionality of Main GUI  **ModelUI** – implements GUIinterface for the ModelUI application  ***DataGUIinterface*** – basic functionality for Data selection  **ModelPlots** – implements DataGUIinterface to plot results  **DataEdit** – implements DataGUIinterface for editing data sets  **DataManip** – implements DataGUIinterface to derive new data sets from existing ones  **DataSelect** - implements DataGUIinterface for data selection in stand-alone functions  **DataStats** – implements DataGUIinterface to provide a range of statistical analysis functions  ***PropertyInterface*** – basic functionality for handling data input and display  ***DataSet*** - basic functionality for handling timeseries data  ***DSDataSet*** – additional functions to import and handle table datasets. Inherits DataSet.  ***TSDataSet*** – additional functions to import and handle timeseries datasets. Inherits DataSet.  **ModelSpecification** – handle model specifications for different models for default ModelUI  **Project** – details of project definition (name and date) and path and file for project  **ConstantData** – defines constants used in model  **Results** – store model run details as scenarios and handle saving scenarios to an excel file.  **PlotFig** – defines a range of different plot types  **InputFile** – handles the input file list for reference data and imported timeseries data  **RunModel** – template for models that require extensive pre and post processing as part of the run  **NumInputdlg -** Modifies inputdlg Matlab function to handle numeric input and output  **Tabledlg** - Defines a table with rows and columns with the option to edit variables defined in the call to Tabledlg. |

In addition, SedTools uses the following classes and functions:

SedTools – UI for SedTools

SettlingParams – data entry for settling column analysis

SettlingAnalysis – settling column results and plotting

settling\_column – implements cal\_settling by Urs Neumeier, 2005

# Bibliography

1. When new models are added as sub-folders to the MUImodels folder, these are also added to the path when addModelUIpaths is run. [↑](#footnote-ref-1)