Oil Spill and Particle Track Toolbox

User Guide

PLEASE NOTE

1 Oil Spill Boom And Skimmer

This tool can be used to generate input files for the boom and skimmer feature of the extended DHI oil spill model. These input files describe the position of oil booms and attached skimmers in space and time

1.1 Setup

This page allows you to specify a name for the project/setup. You can use these settings to identify different boom specifications in your project.

1.2 Domain Settings

This page is used to define the spatial area used to define the boom/skimmer settings. You have the option to either load an independent domain mesh file or import the domain from an existing Mike21FM or Mike3FM project (files with the extensions *.m21fm, *.m3fm or *.mfm).

Importing a spatial domain from an existing setup will also set the start and end time of the boom and skimmer definition.

1.3 Time Settings

This page is used to define the start and end date/time and the time step size of the boom and skimmer definitions. Principally booms can be specified within a time span covered by the given range.

If the spatial domain was imported, the time settings are also imported and not user specified.

1.4 Boom Definition

This page is used to define one or multiple booms in space and time. It consists basically of two areas, the upper graphical domain view and the tabular view below.

The definition of a boom is an interactive process. The user has to specify the spatial extend of a boom by drawing line segments of the boom on the domain view. A boom consists of at least two (end-) points and the line segment connecting the points. The user can insert additional nodes and move or delete existing nodes of a boom using the context menu. The context menu offers also some zooming options. Details on the boom definition like the exact date/time and start- and end-point location can be edited in the lower table. This table offers also an edit dialog for the individual segment settings, i.e. the x-,y- positions of the inner nodes as well as the pump speci-

fications when pressing the edit button. Please note that some of the settings can not be modified (i.e. the start/endpoint location) using the edit dialog.

The active boom on the screen is indicated by a frame around it. The user can select to annotate the booms with the time stamp and the total boom length in [m].

1.5   Output File Selection

This page is used to specify the output file name of the file containing the boom specifications. This information can be used as input for the DHI extended oil spill model.

Note: In order for a boom to work in the DHI extended oil spill model, the special data file needs to be present in the simulation root folder, and the file name must be "boom_file.csv".

1.6   Status

The overview page presents you a summary of all the input parameters you have specified. You should inspect that the parameters are correct and if not go back and change them.

From the page you can either press "Finish" to save the setup which will transfer your setup definition to the current toolbox file or execute the setup to generate the a input file for the boom and skimmer feature.

2 DFS Reverse

This tool can be used to reverse dfs files in time, i.e. resort the data such that the first time step becomes the last and the last the first etc.

Additional multipliers can be specified to modify the data values.

The main usage is to provide reversed input fields for particle backtracking.

2.1 Select Input File

Please select the input file. This must be an existing dfs0, dfs2, dfs3 or dfsu data file.

2.2 File items and multipliers

Please provide a multiplier for each item in the data file.

To edit select the item/factor in the table and enter a new value. To keep the original item value (i.e. just reverse the time axis) use a multiplier value of 1.0; Velocity items will have a default factor of "-1.0", resulting in a reversed flow direction.

2.3 Output File Name

Please specify the output file name. This should be different form the input file name. Existing files will be overwritten without further notice.

2.4 Status

The overview page presents you a summary of all the input parameters you have specified. You should inspect that the parameters are correct and if not go back and change them.

From the page you can either Finish the setup which will transfer your setup definition to the current toolbox file or execute the setup.

2.5 Examples

2.5.1 Dfs0

This example shows a reversed time series. The values in the original series (blue) go from 100 to 1. As a multiplier with a value of 1.0 has been used the reversed series (red) goes from 1.0 to 100.
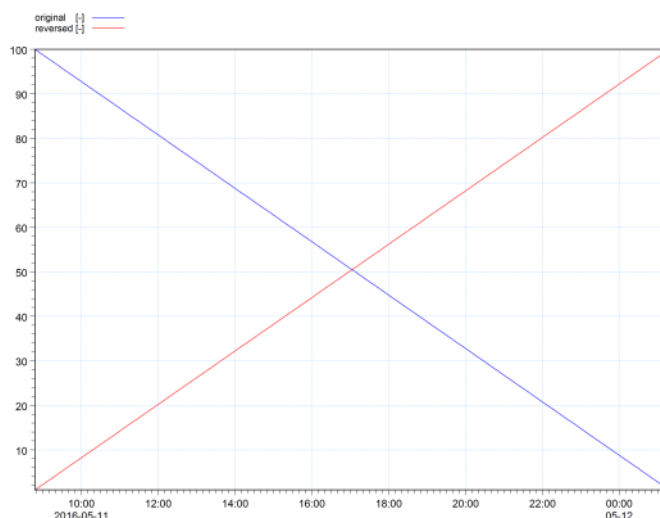
Figure 2.1   Original and resulting dfs0 data values

2.5.2   Dfsu 2D

This example shows a reversed dfsu file with 2D flow information. The original series (Figure 2.2) is taken from the Øresund particle drift example. For simplicity the centre time step is shown in both the original (Figure 2.2) and the reversed series (Figure 2.3), i.e. the images are directly comparable.
To demonstrate the reversed time axis, time series were extracted from both files at the same coordinates, showing the U component of the flow over time, see Figure 2.2 and Figure 2.3, respectively.

As seen from Figure 2.3 vectors demonstrate the reversed current flow in the center time step. The time series, besides the negated U component, demonstrate the reversed time axis.
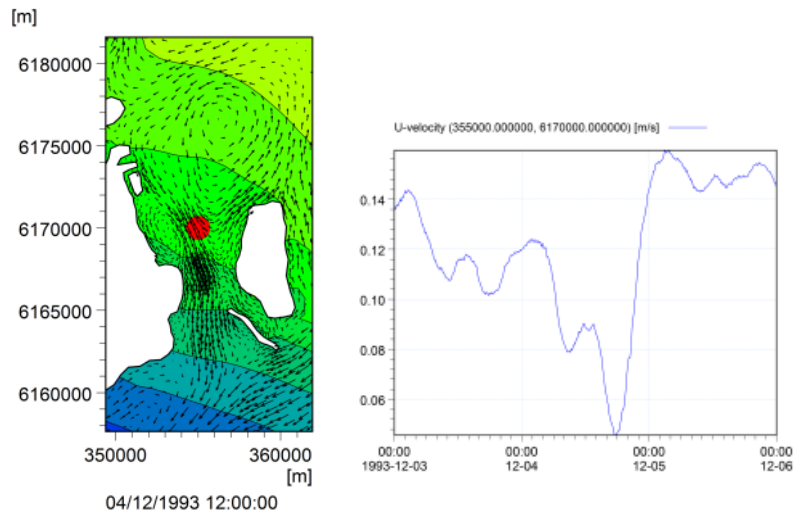
Figure 2.2   Original data.
Left: instant flow field, Right: extracted time-varying U-velocity in point
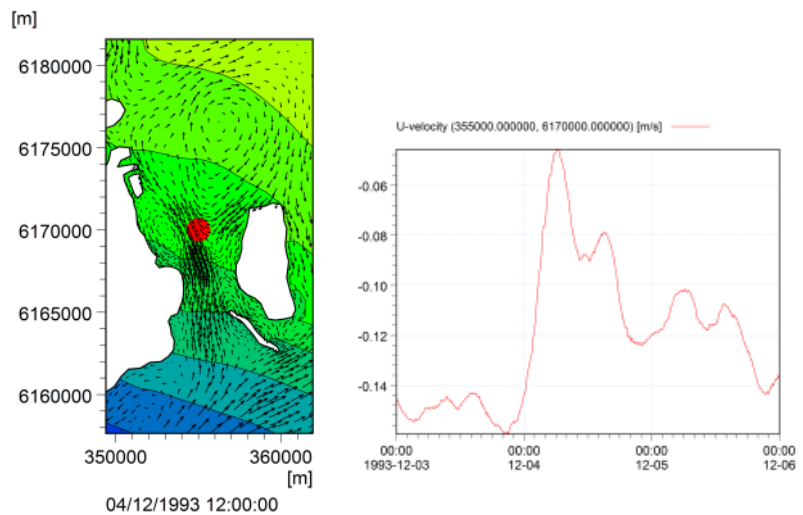
Figure 2.3   Reversed flow.
Left: instant flow field, Right: extracted time-varying U-velocity in point

3 Particle Track Tool

The Particle Tracking module and MIKE ECO Lab ABM/Oilspill module can store particle tracks in various DHI track format. The DHI XML track format is described in detail in the MIKE ECO Lab reference manual, chapter "The XML track format". The XML format offers two sub-formats. In uncompressed tracks all information is stored as separate XML element for each variable, in compressed tracks all variable values are stored as large CDATA element. The other available format is a binary track format. The Particle Track Tool can be used to convert tracks from one format into another track format or export data into a standard dfs0 or ASCII data files. Further the tool can be used to calculate some statistics (min/max, average, std dev etc.) on of included variables and the track length (in 2D and 3D). It can also be used to calculate the number of active particles per time step and can list a general information header on the included classes and variables.

3.1 Track Tool

This tool can be used to convert tracks stored in various formats. It can also filter some sub-items and compute some general statistics like min, max, average, std. deviation and track length.

3.2 Input Selection

Select the input file. The drop-down list covers the supported particle track file formats. Table 3.1 shows the supported particle track formats[1] that are currently supported to read particle data.

Table 3.1   Supported particle track formats for input

Format	Extension
Uncompressed DHI XML track	.xml
Compressed DHI XML track	.xml
DHI particle track	.track

3.3 Job Selection

Select the process you want to perform. You have the following four options:

● Track Conversion
Save the track as a new format, including the option to filter some class/variables

---

1 The supported particle track formats in MIKE Zero may be extended by plugins

- Track Statistics
  Compute statistics on track variables

- Particle Counting

- Extract Class/Item Header Information
  Generate some general header information on included track class/variables

3.4 Track Conversion

The tool can convert/save particle track data in various formats[1]. Table 3.2 shows the formats that are currently supported to write particle data.

Table 3.2   Supported formats for writing particle data

Format	Extension
Uncompressed DHI XML track	.xml
Compressed DHI XML track	.xml
DHI particle track	.track
Keyhole Markup Language Track (Google Earth)	.kml
ASCII table	.txt
DHI dfs0 time series (one per particle)	.dfs0

3.4.1 Output options

If tracks are stored in the DHI XML format, an XLST style sheet can be generated to facilitate track viewing inside a standard web browser.
Please note that this option for performance reasons should only be used with smaller tracks as the transformation will require the browser to read and manipulate the complete track data in memory.

If tracks are converted using the dfs0 format, the tool will create a separate file for each included particle. The class and particle id will be appended to the specified file name.

Tracks may also be reversed in time, i.e. resorted so that the first time step becomes the last and the last the first etc. This option is primary intended to be used in combination with particle back-tracking.

If desired, the positions stored in the track may be transformed into another geographic projection. The projection has to be given as full specified projection string.

If data is intended to be written out as ASCII data, several formatting options can be specified. This includes the data separator and the width and preci-

---

1   The supported particle track formats in MIKE Zero may be extended by plugins

sion of the value format. You can use escape sequences to represent non printable ASCII characters as data separator, see also ASCII data separator *(p. 16)*.

3.5   Track Statistics

This tool can calculate statistics on variables for both individual particles and summary statistics for all particles of a class. This includes the min, max, average, standard deviation, count and sum of individual variables as well as the initial (first non-delete value), final (last valid position) value and the occurrence time step of the maximum, minimum, initial and final value (respective particle ID in class summary). Please check the file header for detailed information on the specific value order.

If the track includes valid position data (i.e. variables with the variable codes "x", "y" and "z") the tool will also compute the total track length (i.e. the summed distance between consecutive positions) as well as the movement length (i.e. the distance between the initial and final position) in 2D and 3D.

The statistics will be written out as ASCII data file. Several formatting options can be specified. If the "ASCII-Tree" format is used, some additional characters are included to simplify reading. Generally one line per variable will be written. If required, all data of one particle can be written as a single line. The options includes also the data separator and the width and precision of the value format. You can use escape sequences to represent non printable ASCII characters as data separator, see ASCII data separator *(p. 16)*.

3.6   Particle Counting

This tool can be used to count the number of valid particles per time step based on the value of the first included variable, usually the x-position (i.e. a particle is counted if the variable value is a non-delete value).

The count data can be saved as dfs0 file (one dfs0 item per class) or as ASCII table.You can use escape sequences to represent non printable ASCII characters as data separator, see ASCII data separator *(p. 16)*.

3.7   Extract Class/Item Header Information

The particle tool can be used to extract a summary header of the included class and variables. The information will be stored as simple ASCII data file. You can use escape sequences to represent non printable ASCII characters as data separator, see below.

3.8   Particle Filter

By default all stored particle classes and variables are used.

If desired, individual class/variables can be selected by defining proper class/item filters.

3.8.1   Class/item filter

Here you select the variables and classes that are to be included in the output.

Please note that the class order is given but the order of the variables inside the class is specified by the order of their selection. All variables of a class can be selected/deselected by clicking the class name directly.

3.9   Output file name

Please specify the output file name. This should be different from the input file name. Existing files will be overwritten without further notice.

3.10   Status

The overview page presents you a summary of all the input parameters you have specified. You should inspect that the parameters are correct and if not go back and change them.

From the page you can either Finish the setup which will transfer your setup definition to the current toolbox file or execute the setup.

3.11   ASCII data separator

If ASCII data files are written you can use the escape sequences listed in Table 3.3 to represent non printable ASCII characters as data separator.

Table 3.3   Escape sequences to represent data separators

Sequence	Description	Encoding
\'	single quote	byte 0x27 in ASCII encoding
\"	double quote	byte 0x22 in ASCII encoding
\?	question mark	byte 0x3f in ASCII encoding
\\	backslash	byte 0x5c in ASCII encoding
\a	audible bell	byte 0x07 in ASCII encoding
\b	backspace	byte 0x08 in ASCII encoding

Table 3.3   Escape sequences to represent data separators

Sequence	Description	Encoding
\f	form feed - new page	byte 0x0c in ASCII encoding
\n	line feed - new line	byte 0x0a in ASCII encoding
\r	carriage return	byte 0x0d in ASCII encoding
\t	horizontal tab	byte 0x09 in ASCII encoding
\v	vertical tab	byte 0x0b in ASCII encoding

4 Connectivity Tool

This tool can be used to calculate the connectivity between grid cells, based on the position of particles stored in a track and mapped to a user defined Cartesian grid. It can be used to calculate the "downstream" or "upstream" connectivity, i.e. determine where all particle originating from a defined cell drift to (downstream connectivity) or where all particles that ended in the cell had started (upstream connectivity).

4.1 Select Input

Please select an input file in a DHI particle format. If needed the geographic projection inside the track can be updated/changed to match the given projection.

The dropdown list covers the supported particle track file formats. Table 4.1 shows the supported particle track formats[1] that are currently supported to read particle data.

Table 4.1 Supported particle track formats for input

Format	Extension
Uncompressed DHI XML track	.xml
Compressed DHI XML track	.xml
DHI particle track	.track

4.2 Mapping Definition

The tool can be used to extract from /append data to a previous generated connectivity database or to start with a new database created form a particle track.

4.3 Grid Definition

If you start a new connectivity database you must specify the definition of the spatial mapping grid. The mapping grid is a regular, Cartesian grid defined by its lower left and upper right point and the width and height of a grid cell. The number of cells in j/k directions are computed automatically ($n=(max-min)/dx$) and the maximum may be adjusted ($max=min+(n*max)$) so that the grid covers the specified area.

---

1 The supported particle track formats in MIKE Zero may be extended by plugins

If it is intended to output of the connectivity information as a standard dfs2 data file, the origin may be adjusted automatically by moving the lower left point a half grid cell to the left. This option is active by default.

Particle positions with coordinates outside the specified mapping grid will not be considered for the construction of the connectivity matrix.

4.4   Input Data Selection

Although the primary information for the connectivity mapping is derived from the coordinates stored in the particle track, you have to specify the particle class id to be used (usually 1, but depend on how many classes had been defined in the setup and written to the track) and a variable code. The variable code will be "Mass" for tracks generated by the standard particle tracking module. For tracks generated by the MIKE ECO Lab module (ABM/ Oilspill) it will depend on the data written to the track. In doubt use the ParticleTool to extract information on the available class/variable header.

If desired the particle may be filtered, i.e. only particles where the selected variable is within the specified range will be considered for the connectivity mapping. By default, no such filter will be applied (all valid particle positions will be used).

4.5   Mapping Mode

The mapping mode defines which particle positions will be used for the mapping to define a connection between grid elements. Only positions within the area covered by mapping grid will be considered, particle positions outside this area are invalid. The user can choose to select the

- release and last position
    - i.e. the start- and endpoint of the particle if both are inside the mapping grid

- release and current position
    - i.e. representing the movement from the release along the "track"

- the previous and current position
    - similar to the last but not bound to the release position, i.e. grid cells on the track

The mapping mode can use any combination of these.

4.6 Output Definition

Here one or more outputs can be added or removed. To add outputs use the "add" button and define the output settings in the following dialog. To remove outputs form the list, select an output and press the "del" button.

For outputs in ASCII format some common options like the used delimiter and the width and precision of numbers can be specified. If "Include default values" is enabled, values will be written for all elements, otherwise empty elements will be skipped leading to smaller ASCII output files.

4.6.1 Output specification

On adding an output you have to specify its details, i.e. the file name, type and format in the following dialog.

If a connectivity matrix/database output is selected the user can choose to save it as binary database or plain ASCII representation.

If downstream or upstream connectivity outputs are selected, the user must also specify the coordinates of a reference point. All connectivity data will refer to information relative to the cell in the mapping grid where this reference point is located. Connectivity outputs will be saved as DHI dfs2 files, either as ASCII Dfs2 or binary dfs2 data.

For each output the user can select to normalise it (i.e. the output will contain relative instead or absolute values). Additionally the user has to specify if the origin (cell [0/0]) of the output is the upper left corner (normal rows) or the lower left corner (reversed rows). The later represents the normal dfs2 order and is enabled by default.

4.7 Overview

The overview page presents you a summary of all the input parameters you have specified. You should inspect that the parameters are correct and if not go back and change them.

From the page you can either Finish the setup which will transfer your setup definition to the current toolbox file or execute the setup.

4.8 General Description of Connectivity

The connectivity of a grid is defined as the information how cells are "linked" together. To specify such a link, two cells in the grid must be selected and their connection must be described. If the link is directed, a "downstream" and an "upstream" connectivity can be separated (otherwise the connection is undirected). In the example shown in Figure 4.1 the blue cell is linked to the

red one (indicated by stars). As the link is directed, the blue element is con-
nected "downstream" to the red one. This equivalents to the red cell being
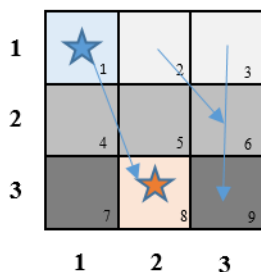"upstream" connected to the blue one.

Figure 4.1   Example for defining terms for the connectivity of grid cells
              Small integers defines the grid cell number

In the tool the two grid cells are selected by mapping two particle positions
(see Mapping Mode) onto a user defined grid.

The information on the connectivity of all grid cells will be stored in a large
matrix, called connectivity matrix. This matrix will note the connection and link
strength of each element to any other elements (i.e. the matrix will have the
dimension $(n * m)^2$ if the mapping grid is n * m). The link strength corresponds
to the number of particles moving from one cell to the other (i.e. 0 represents
no connection, 1 exactly one particle and 10 ten particles etc.). Once build,
the connectivity matrix represents all information needed to derive the con-
nectivity for any cell. A row in this matrix represents the downstream connec-
tivity whereas a column can be used to construct the upstream connection.
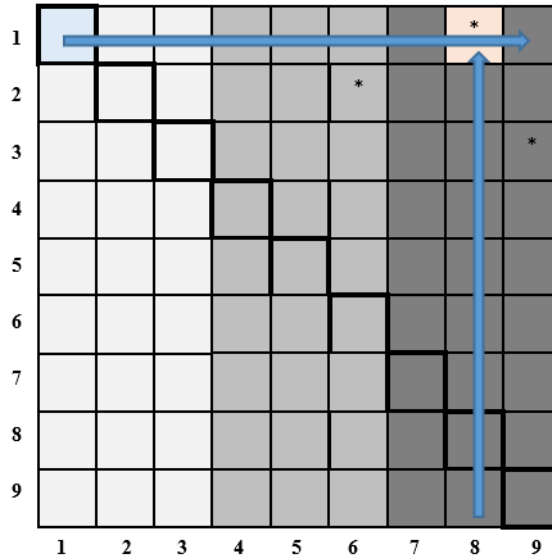
Figure 4.2   Example of connectivity matrix

Figure 4.2 shows an exemplary connectivity matrix for the simple example shown in Figure 4.1. A link is denoted by "*", i.e. there are links between cells 1/8, 2/6 and 3/9. The row k represents the downstream connectivity of the grid cell with the linear number k whereas a column j the upstream connectivity of the jth cell. j and k depend on the linearization of the mapping grid (i.e. how the 2D mapping grid is linearized into 1D by numbering the elements). Just in this case (where the mapping grid is symmetric) j, k correspond to the main diagonal.

INDEX