

Coastal Morphodynamic Models: Generic Guidelines for Model Developers

Embedding ICOASST into practice deliverable

January 2017

Outline

This document aims to provide a useful guide for model developers when writing a user manual for coastal morphodynamic models. It is based on lessons learnt from the evaluation of a suite of 6 models and 2 model compositions. It includes a generic model user manual structure together with recommendations of valuable content that should be included. The appendix includes recommendations based on the composition evaluations, summarising the risk associated with model composition development.

The document has been written with the proposed structure for the User Manual, so that the headings and subheadings are the ones proposed to be followed. This is just a suggested contents table, which the developer might decide to structure differently; however it is recommended that the issues described in each of the sections are taken into account in order to aid potential users in how to use their model appropriately.

Table of Contents

Outline	1
1. Introduction	2
1.1. Model description	2
1.1.1. Key features	2
1.1.2. History	2
1.2. Typical application	2
2. Naming convention	3
3. Model Basics	3
3.1. Technical detail	3
3.2. Boundary conditions	3
3.3. Assumptions and limitations	3
3.3.1. Assumptions	3
3.3.2. Limitations	3
3.3.3. Known instabilities	3
4. Running Model	3
4.1. How to run the model	3
4.2. Input files	4

4.3. Steering file	4
4.4. Other file	4
4.5. Errors	4
5. Running Model with supplied input	5
5.1. Downloading and running	5
5.1.1. Pre-requisites	5
5.1.2. Download	5
5.1.3. File structure	5
5.1.4. Running with example files	5
6. Output Files	6
7. References	6
Appendices	8
A. <Insert appendix heading>	Error! Bookmark not defined.

1. Introduction

This section should be used to deliver a clear but detailed model description, as well as information regarding its typical application.

1.1. Model description

The model description is suggested to outline how the model works, the type of model, and its main assumptions and limitations. It is suggested to include a basic diagram in the form of an image depicting the model elements and/or processes.

1.1.1. Key features

A list of key features of the model and definitions should be included in this section, with any relevant formula (for example: equilibrium, exchange of sediment between the elements, drivers etc.).

1.1.2. History

A brief history of the model or description of predecessor versions with references may be useful to a user or developer who would like to understand more about the models evolution or requires technical detail.

1.2. Typical application

The models typical application provides detail of its use and why it was built, in particular what are its objectives within coastal morphology. Other suggested applications are valuable but not always necessary.

2. Naming convention

This is a useful section to include if the model has a set naming convention for the input/output files and steering files.

3. Model Basics

This section can be used to expand on the basic model description in the introduction (section 1).

3.1. Technical detail

Include some technical detail and extra formula the model is based on. Diagrams, if not included already, are useful within this section to add an extra level of understanding of the processes within the model. If the model has been adapted or developed especially for a particular use, this section should include detail of what has been changed and why.

3.2. Boundary conditions

List and describe the boundary conditions, including how they are calculated or estimated.

3.3. Assumptions and limitations

3.3.1. Assumptions

List and describe all assumptions within the model, in order of importance.

3.3.2. Limitations

List and describe all known limitations, in order of importance.

3.3.3. Known instabilities

Use this section to highlight if the user could configure the model in a way that causes an instability.

4. Running Model

To run the model the user needs to understand all the inputs files, input variables and how they are changed. If there are more than one input file, there should be a description for each of them, including all the variables used for each. Therefore, it is necessary for the user manual to contain a full list of inputs names, definitions, range of values and units per input file.

4.1. How to run the model

Specify the code language, compilers, versions and requirements to run the model.

4.2. Input files

Specify the amount of compulsory files needed to run the model, together with any optional files that can also be used and when to use them. A description of the different input files, their format and their variables will then follow in each subsection.

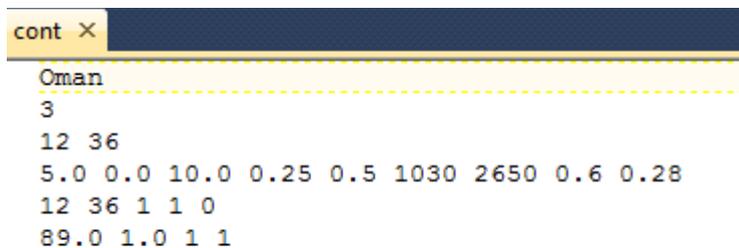
4.3. Steering file

The steering file format would be described here, together with all the variables in it, which should be fully defined (ideally within a table, see Table 1 below for example), along with their range of values and units

Table 1: Example of a Table containing definition of variables in a given file

Variable name (units)	Description	Default value and Range of values
Dfifty(m)	Value for the sediment D50, which represents the main sediment size in the area applied.	The default value is 0.0002m. D50 needs to be between 0.1mm and 1.5mm for the applied formulation to be valid.
...		

It is recommended that there is an example steering file in the form of an image, see Figure 1 for example.



```

cont x
Oman
3
12 36
5.0 0.0 10.0 0.25 0.5 1030 2650 0.6 0.28
12 36 1 1 0
89.0 1.0 1 1

```

Figure 1: Example of a control file image

4.4. Other file

A similar description to the steering file above is recommended for each of the files (optional or compulsory) used within the model. Ideally, each will include a table describing the variables used and an image so that the format is easily understood.

4.5. Errors

Within this section or the next typical errors should be dealt with. For example, it is suggested to include typical error messages and what problem they refer to. Sometimes this is not so simple and providing a contact email will suffice.

5. Running Model with supplied input

This section can be merged with section 4, although, it is suggested that these are separate to avoid confusion. The information supplied in this section should make it simple for a user to download and run the model using the supplied inputs.

5.1. Downloading and running

5.1.1. Pre-requisites

List all pre-requisites that are needed to run the model, for example, the computing system, programmes and their version.

5.1.2. Download

Tell the user where the model and example inputs can be downloaded from.

5.1.3. File structure

It is recommended to include the file structure of the download to check that all the components have been downloaded and any supporting documents are defined.

5.1.4. Running with example files

From previous user manual evaluations, it is seen to be beneficial for the model user to have a step by step guide to follow when running the example with the supplied inputs and steering file. This should be accompanied by an image depicting a successful run (screen output example in figure 2) or text describing how a user can tell if the run was successful.

```
Timeframe:
From year 1015 output every 1 years
Until year 2015
Run duration: 706706 tides

Assigning wave point numbers
Structure control files loaded
look-up tables loaded
C:\workshop\28_UserManualExample Year: 1015
```

Figure 2: Example of successful model run screen output

6. Output Files

All output options, files and variables are listed in this section. The variables should be defined and units given. An image of what the output file looks like is necessary, see figure 3 for an example. It is recommended to include an example of how to process the output data along with an example plot.

```
Results for time:
SECTION      X
1           0.0000
2          100.0000
3          200.0000
4          300.0000
```

Figure 3: Example of data in output file.

7. References

List of references used throughout the document.

Appendices

A. Model Composition Development

The guidelines in this document were drawn out from the review of the models and models' user manuals in the CCO iCOASST webpage. Additionally, the compositions were also reviewed, and some conclusions on the development are summarised here, emphasising the risks associated with model composition development.

Model composition development is specific to the models being linked and the software these models rely on. This makes it difficult to write general recommendations on the development of model compositions. There are, however, two major issues that should be borne in mind when developing a composition.

It is understood that model linking can be quite an onerous task, especially if the models were developed a long time ago and therefore their structure was not designed for model linking. The models will tend to require extensive code development to get the right structure and expose the right variables to be ready for the other models. Another difficulty comes from the software requirements of the different components and possible incompatibilities from the different models to be linked. Thirdly, there might be proprietary software dependencies which are difficult to overcome when such compositions need to be made public.

The problems encountered within both the Liverpool site and the Suffolk site compositions all emanate from these difficulties when linking models. Moving forwards, it will be advisable when writing new software to design it with a structure that would make it easy to link to other models in the future. Moreover, there are two fundamental issues to be borne in mind when linking models:

- **Portability:** it is recommendable that the model compositions are tested on a different environment to try and assure portability of the composition. The need for full paths in OpenMI does not help with this issue, although it should be considered as much as possible.
- **Software compatibility:** when the models are dependent on software it is difficult to foresee possible updates to the software that might make the composition fail when these updates are carried out in the future. However, it is desirable if possible to reduce the software dependencies and try and write the linkages in a generic form to try and overcome this problem.