Manual and documentation for the "Shoreline Evolution model"

June 22, 2018

Contents

1 Introduction

1.1 Origin and history

The shoreline evolution or 'Evo' model began as a graduate research project at the University of Queensland Huxley [2009], Huxley et al. [2010] with support provided by WBM BMT. Evo was subsequently developed as a consulting model at WBM BMT Huxley [2011], Teakle et al. [2013]. In essence, the Evo model is a synthesis of preexisting model theory from both longshore and cross-shore equilibrium type models into a hybrid model. Evo is also well suited for applications within statistical frameworks Gravois et al. [2016]. The University of Queensland elected to use the Evo model for the Bushfire & Natural Hazards Cooperative Research Center (BNHCRC) project 'clustered disasters on the coast - storm surge', with permission granted by WBM BMT. A collective decision was made by WBM BMT and The Univerisy of Queensland and to release the Evo model as an open source code and to provide a user guide manual, example model cases, and input/output scripts created throughout the BNHCRC project.

1.2 Purpose

Evo is a research tool designed to study the effects of waves and storm surges on beach erosion, in terms of both storm demand (citation) and long term shoreline/dune recession. This is achieved by calculating both the longshore sand transport gradients and cross-shore redistribution of sand from large waves or elevated tides that dominate the recession and storm demand processes respectively. The model also allows beach recovery caused by small accretionary waves and by artificial means (i.e. addition of sand from dredging). Seawalls and groyne structures can be included in the modelling, and also modified, to quantify the effects of coastal structures on beach evolution. The recommended spatial and temporal scales for Evo are beach lengths up to 100 km and simulation durrations up to 100 years.

2 Model Physics

2.1 Introduction

The Evo model is a hybrid cross-shore and longshore beach morphodynamic model that can incorporate coastal structures (groynes, seawalls), sediment sources and sinks (river entrance, dredging) and includes capabilities for

curvilinear grids. The underlying theory and equations used in Evo are described in the next sections.

2.2 Computational Grid

The Evo base is defined as consecutive easting and northings pairs of both base and end points. An alongshore chainage coordinate is then defined as the cumulative sum of distances between consecutive grid base points. Longshore transport, groyne structures and wave heights and directions all use the Evo grid as a basis. Next, profile elevation cross-sections are defined in between the Evo grid lines and serve as basis for cross-shore sand movement, seawalls, and sediment sources and sinks. An example Evo computational grid definition is given in 1.

Figure 1: Evample Evo Gird.

2.3 Cross-shore Theory

2.3.1 Overview

The equilibrium beach profile concept originally proposed by Bruun [1954] and further corroborated by Dean [1977], specifys a $Z = AX^{2/3}$ relationship

between the cross-shore beach profile vertical elevation Z and the horizontal distance of shore X. This theoretical profile shape was later combined with a Brunn rule type Bruun [1962] relation for equilibrium beach response to changes in waves and water levels Dean [1991], Dean and Dalrymple [2001]. These concepts were further extended to include the time rate of change between the antecedent and theoretical equilibrium profile Kriebel and Dean [1985, 1993], Miller and Dean [2004] from which the cross shore theory in Evo is based. In practice, the Evo model formulation defines a steady state (equilibrium) cross-shore profile shape for a given constant wave condition and mean water level. Further, the cross-shore profile will remain in this shape until the wave condition or water level is changed. The underlying assumption in this formulation is a conservation of total volume of sand in the cross-shore. A schematic diagram of the profile shape is given in Figure 2 and the definition of parameters are defined in the next sections.

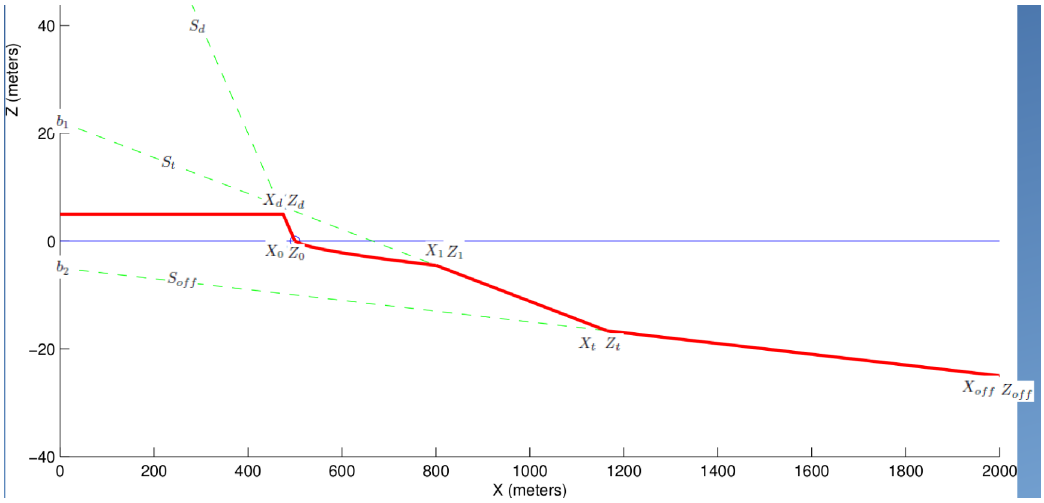

Figure 2: Evample Evo Profile.

2.3.2 Profile Paramters

The cross-shore profile within Evo is calculated by a combining the grid file, xpar file (cross-shore parameter) and icond file (initial condition) inputs parameters using the following equations. Reference to the respective input file sections and Figure (2) for definitions. Reconstructing the profile from these condensed set of cross-shore parameters allows for a decrease in model run time, as well as size requirements for model input and output files.

- Dune Slope Intercept (b_0)

$$b_0 = Z_0 - S_d * X_0 \tag{1}$$

- Cross Shore Distance Dune Top (X_d)

$$X_d = \frac{(Z_d - b_0)}{S_d} \tag{2}$$

- Profile Slope Paramter (A)

$$A = \frac{(Z_0 - Z_1)}{(X_1 - X_0)^{2/3}} \tag{3}$$

- Cross Shore Distance Increment (dX)

$$dx = \frac{(X_1 - X_0)}{NA} \tag{4}$$

- Transition Slope Intercept (b_1)

$$b_1 = Z_1 - S_t * X_1 \tag{5}$$

- Offshore Slope Intercept (b_2)

$$b_2 = Z_{off} - S_{off} * X_{off} \tag{6}$$

- Cross Shore Distance Transition(X_t)

$$X_t = \frac{(b_2 - b_1)}{(S_t - S_{off})} \tag{7}$$

- Transition Elevation (Z_t)

$$Z_t = S_t * X_t + b_1 \tag{8}$$

$$Z_t = S_{off} * X_t + b_2 \tag{9}$$

2.3.3 Equilibrium Profile

In Evo simulations, a cross-shore equilibrium profile is calculated at each profile location at every model time step. The breaking wave height and mean water level entirely define the shape and extent of the 'active profile' region. This formulation is quite similar to that described in Dean [1991], Dean and Dalrymple [2001], Miller and Dean [2004] with the omission of the small contribution of wave set down at the breaking point. Refer to the section on wave transformation (2.5) regarding calculation of the breaking wave height at each profile.

- Height of breaking waves (H_b), breaking depth(h_b), and breaking parameter (γ)

$$H_b = \gamma h_b \tag{10}$$

- Wave set-up(η_{su})

$$\eta_{su} = \frac{K h_b}{1 - K} \tag{11}$$

- Wave setup parameter (K)

$$K = \frac{3\gamma^2/8}{1 + 3\gamma^2/8} \tag{12}$$

The active profile elevations (Z_0) and (Z_1) are then defined between the breaking depth (h_b) and wave set-up(η_{su}) superimposed on the mean water level (wl).

$$Z_0 = \eta_{su} + wl \tag{13}$$

$$Z_1 = -h_b + wl \tag{14}$$

Now based on the specified profile shape paramter A and the $Z = AX^{2/3}$ relation, the active profile width ($X_0 - X_1$) is calculated as.

$$(X_0 - X_1) = \left(\frac{(Z_0 - Z_1)}{A} \right)^{(3/2)} \tag{15}$$

Notice that at this stage, only the active profile width ($X_1 - X_0$) has been calculated and not yet the absolute positions of (X_0) and (X_1). Applying the (2.3.2) relations, an iterative root finding algorithm is solved to determine the absolute locations where profile volume is conserved. The transition between the current profile and the theoretical equilibrium profile shape does not occur instantaneously, rather at a specified rate following the exponential decay model described in the following section.

2.3.4 Profile change rate

Each of the active profile shape parameters moves in a linear fashion from the present state towards the equilibrium state following an exponential decay, namely $f(t) = f_{eq} + (f_{initial} - f_{eq}) \exp(-Kt)$ where (f) is a dummy variable substitution for the active profile parameters (X_0, X_1, Z_0, Z_1). In practice this formulation implies that the further away the equilibrium is from the present state, the faster it will evolve towards. These rates of change are highly dependent on whether erosion or accretion is occurring, erosion (K_e) is a much quicker process compared to accretion (K_a) with typical values expressed in time constants of approximately 2.5 days and 3 years respectively.

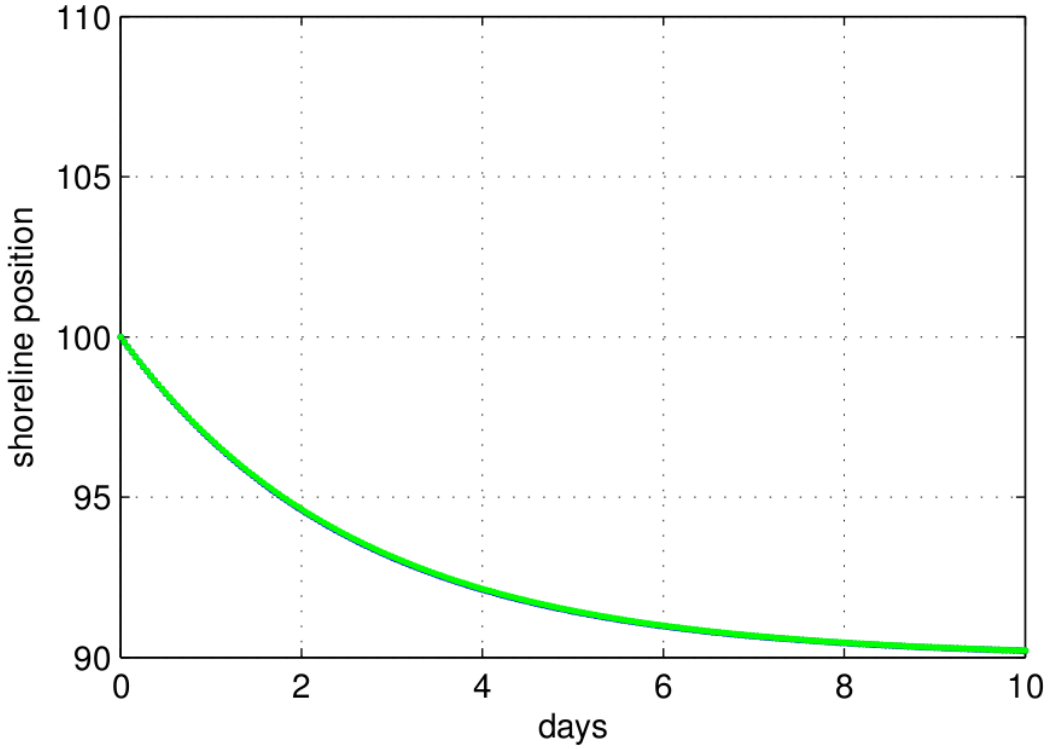

Figure 3: Erosion time scale example.

2.4 Longshore

- The CERC equation

$$Q_{sy} = \frac{K}{16(s-1)\sqrt{\gamma}} \sqrt{g} H_b^{2.5} \sin(2\alpha_b) \tag{16}$$

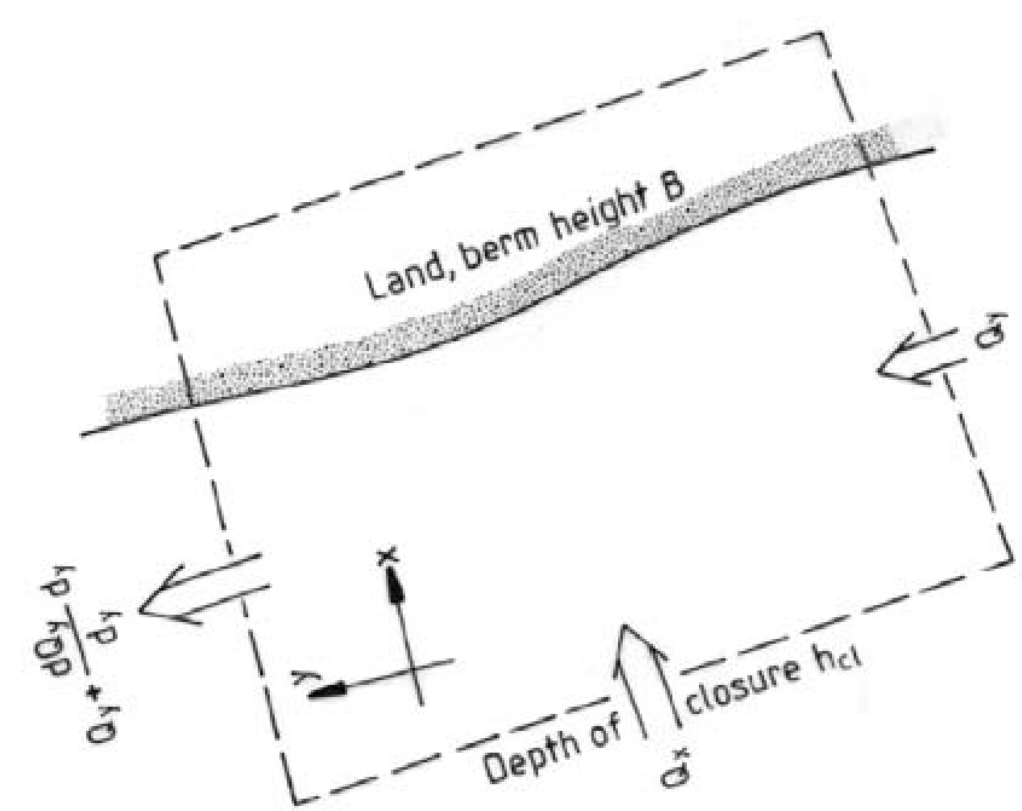

Figure 4: Longshore transport schematic.

2.5 Wave Shoaling and Refrection

The input wave condition time series (height, period and direction) is given at a single location only. The direct implication of this assumption is that the offshore waves along the simulated beach are approximately spatially homogeneous. Although somewhat site specific, the length of simulated beach in Evo should be kept below 100 km to satisfy this condition. Transformation of the wave height and direction from the offshore input location all the way to wave breaking near the shore is considered as two steps; 1) Offshore to intermediate depths and 2) Intermediate to breaking. Futher details are explained in the following sections.

2.5.1 Offshore to intermediate depths

The Evo model uses pre-computed wave lookup tables to account for wave transformation over the shelf and nearshore bathymety. Stationary SWAN model simulations are run to determine the relationship between the wave an-

gle and height offshore and multiple intermediate locations along the coast. From these simulations two data structures are calculated and written to NetCDF files. The wave lookup table holds critical results from the SWAN wave model and makes these readily available to Evo. Specifically two 4-dimensional data structures of coefficients $K\{direction, period, height, chainage\}$ and $D\{direction, period, height, chainage\}$ representing the change in height and direction respectively along the project length at approximately 10m depth. A formal review of this process is found in and an example case will be added as an appendix. Also to note, there is potential to bypass this SWAN modelling step if modelling beaches in NSW, Australia. Nearshore wave lookup tables have been computed for the entire coastline already.

2.5.2 Intermediate to breaking

The second porton of wave transformations is calculated within Evo. After the single point input wave parameters are passed through the lookup table, we now have wave parameters offshore of each transect in the Evo model. To calculate the change in wave height for wave shoaling K_s and refraction K_r between two points connected by straight and parallel contours, the equations are:

$$H_2 = H_1 K_s K_r \tag{17}$$

$$H_2 = H_1 \sqrt{\frac{C_{g1}}{C_{g2}}} \sqrt{\frac{\cos \alpha_1}{\cos \alpha_2}} \tag{18}$$

The change in wave direction α for waves shoaling onto straight and parallel contours the equations is given as

$$\frac{\sin \alpha_2}{C_2} = \frac{\sin \alpha_1}{C_1} \tag{19}$$

Note that the wave angle is applicable only to the longshore transport. The final calculation to determine the breaking wave height is the γ parameter relating the wave height to the water depth. Typically $\gamma \cong .78$ such that a wave that breaks at 2m depth will have a height of 1.56 m.

$$H_b = \gamma h_b \tag{20}$$

3 Model Organization

3.1 Example file structure of an Evo runs directory

For each simulation or run at a particular site, the suggested Evo file structure is to divide files into three main folders; *input*, *output* and *processed*. All files to run the model are kept in *input*, all direct output from the Evo simulation are directed into *output*, and the combined Matlab post-processing of input and outputs are saved in *processed*. Two exceptions being the *log* (.log) and *restart* (.rst) files that are written by Evo into the *input* folder.

```
runs/
└── example/
    ├── input/
    │   ├── run_script
    │   ├── example.evc
    │   ├── example_grid.csv
    │   ├── example_xpar.csv
    │   ├── example_icond.csv
    │   ├── example_lookup.nc
    │   ├── example_seawall.csv
    │   ├── example_groynes.csv
    │   ├── example_transport.csv
    │   ├── example_sourcesink.csv
    │   ├── example_wave.csv
    │   ├── example_wl.csv
    │   ├── example_xsect_ids.csv
    │   ├── example_xsect_ids_eq.csv
    │   ├── example.rst
    │   └── example.log
    ├── output/
    │   ├── example_xshore
    │   ├── example_volume
    │   ├── example_shoreline
    │   ├── example_transport
    │   ├── example_wave_off
    │   ├── example_wave_int
    │   ├── example_wave_brk
    │   ├── example_xsect
    │   └── example_eqxsect
    └── processed/
        ├── example_in.mat
        └── example_out.mat
```

Table 1: Evo runs directory structure.

3.2 Details of *input* files

The following is a description of the format and contents of Evo input files. All input files are simple ascii text files with the exception of the wave lookup table (netcdf). Further, most of the ascii files are of the comma delimited

type (.csv). As described here, the file names are based on the example case listed in the file structure above. Note that a graphical representation (plan and profile views) of the parameters defined in the following files are given in figures 1 and 2.

3.2.1 Control File

The .evc Evo control file defines a set of model instructions for each run. This file works by assigning values and names to various keywords described below. In the following the keyword is listed on the left-hand side and the values and names on the right-hand side and extra notes are in parentheses.

```
modeltype == simple, huxley ! (only two choices allowed)
doxshoreupdate == .true./.false. ! (only two choices allowed)
dolongshoretransport ==.true./.false. ! (only two choices
   allowed)
grid file == example_grid.csv
wave transformation file == example_lookup.nc
xsect parameter file == example_xpar.csv
initial condition file == example_icond.csv
seawall definition file == example_seawall.csv
groynes definition file == example_groynes.csv
time format == ISODATE
start time == 01/01/2000 00:00
end time == 01/01/2100 00:00
timestep == 3600.0 ! seconds
cerc coefficient == 0.160
Breaker index == 0.78
bc == wave, example_wave.csv
loop bc == .true./.false.
end bc
bc == water level, example_wl.csv
loop bc == .true./.false.
end bc
bc == transport , example_transport.csv
chainage == 0
loop bc == .true./.false.
end bc
bc == source/sink , example_sourcesink.csv
chainage == 1000
loop bc ==
end bc
output dir == ./../output
output == xshore
output interval == 10.0 ! days
end output
output == volume
output interval == 10.0 ! days
end output
output == shoreline
output interval == 10.0 ! days
end output
output == transport
output interval == 10.0 ! days
end output
```

```
output == offshore waves
output interval == 10.0 ! days
end output
output == intermediate waves
output interval == 10.0 ! days
end output
output == breaking waves
output interval == 10.0 ! days
end output
output == xsect
output interval == 10.0 ! days
output list == example_xsect_ids.csv
end output
output == equilibrium xsect
output interval == 10.0 ! days
output list == example_xsect_ids_eq.csv
end output
write restart dt == 18250
```

3.2.2 grid file

This file sets the computational domain of the Evo run. Longshore transport, groyne structures and wave heights and directions all use the Evo grid as a basis. The file contains the following header.

```
chainage,base_x,base_y,end_x,end_y
```

Where, the terms are defined as follows:

($chainage$) cumulative distance between consecutive grid base pairs (m).

($base_x$) grid base easting (m).

($base_y$) grid base northing (m).

(end_x) grid end easting (m).

(end_y) grid end northing (m).

After the header, each subsequent row in the grid file contains 5 comma separated numbers defining a grid line in the model.

3.2.3 xpar file (cross-shore parameter)

This file sets parameters for each Evo profile (defined in between each grid) that remain constant throughout the Evo run. The file contains the following header.

```
chainage,dune_elevation,dune_slope,a,transition_slope,
   offshore_slpe,offshore_elevation,Ke,Ka,Hbmin,description
```

Where, the terms are defined as follows:

($chainage$) cumulative distance between consecutive grid base pairs (m).

(Z_d) vertical dune elevation (m).

(S_d) dune slope given as positive number although elevation decreases from land towards sea.

(A) profile slope parameter defines the active profile.

(S_t) transition slope.

(S_{off}) offshore slope.

(Z_{off}) offshore elevation (1/yrs)

(K_e) erosion rate coefficient (1/yrs).

(K_a) accretion rate coefficient.

(H_b) minimum breaking wave height (m).

($Description$) optional text comment.

After the header, each subsequent row in the xpar file contains 11 comma separated numbers defining a cross-shore parameters in the model.

3.2.4 icond file (initial condition)

This file sets the initial conditions for the active profile of each Evo profile (defined in between each grid). When combined with the cross-shore parameters, these initial conditions define the initial volume and shape of the each cross-shore profile. For details about reconstructing a profile see section 2.3.2 and figure 2. Note that the active profile will evolve based on the wave and water level forcing throughout the Evo run. The file contains the following header.

```
chainage,x0,z0,x1,z1,dx
```

Where, the terms are defined as follows:

($chainage$) cumulative distance between consecutive grid base pairs (m).

(X_0) cross-shore distance of active profile start (m).

(Z_0) elevation active profile start (m).

(X_1) cross-shore distance of active profile end (m).

(Z_1) elevation active profile end (m).

(dX) cross-shore distance increment (m).

After the header, each subsequent row in the icond file contains 5 comma separated numbers defining a crossshore parameters in the model.

3.2.5 Wave lookup table

```
netcdf
```

3.2.6 Wave file (forcing)

```
TIME,WVHT,WVPER,WVDIR
```

3.2.7 Wl file (forcing)

```
TIME,WL
```

3.2.8 Seawall file

```
CHAINAGE,WALL_X,WALL_Z
```

3.2.9 Groynes

```
CHAINAGE,GROYNE_X,START_TIME,END_TIME,COMMENT
```

3.2.10 Transport

```
TIME,TRANSPORT
```

3.3 Details of *output* files

3.3.1 xshore file (cross-shore parameter)

```
variable,time/chainage,chainage(1),chainage(2)...
```

Where, the terms are defined as follows:

(*variable*) sequential list of active profile parmaters (X_0, X_1, Z_0, Z_1).

(*time/chainge*) time and cross-shore distance of said parameter (DD/MM/YYY HH:MM:SSS, m).

(*chainge*(1)) alongshore position of cross-shore transect 1 (m).

(*chainge*(1)) alongshore position of cross-shore transect 2 (m).

After the header, each subsequent row in the xshore file contains N+2 comma separated numbers defining a cross-shore active profile parameters in the model. Where N is the number of cross-shore profiles, i.e. one less than the number of grids.

3.3.2 volume file

```
variable,time/chainage,chainage(1),chainage(2)...
```

Where, the terms are defined as follows:

(*variable*) sequential list of active profile parmaters. See below for list of variables.

(*time/chainge*) time and cross-shore distance of said parameter (DD/MM/YYY HH:MM:SSS, m).

(*chainge*(1)) alongshore position of cross-shore transect 1 (m).

(*chainge*(1)) alongshore position of cross-shore transect 2 (m).

After the header, each subsequent row in the xshore file contains N+2 comma separated numbers defining a cross-shore active profile parameters in the model. Where N is the number of cross-shore profiles, i.e. one less than the number of grids. The output variables are:

($volume - subaerial/m$)

($volume/m$)

($volume - total$)

($volume - error$)

3.3.3 shoreline file

It is not advised to use this output as it is the profile intersection with mean water level not the zero AHD contour.

3.3.4 transport file (cross-shore parameter)

```
variable,time,chainage(1),chainage(2)...
```

Where, the terms are defined as follows:

(*variable*) sequential list of transport parmaters. See below for list of variables.

(*time*) time of said parameter value (DD/MM/YYY HH:MM:SSS).

(*chainge*(1)) alongshore position of grid 1 (m).

(*chainge*(2)) alongshore position of grid 2 (m).

After the header, each subsequent row in the xshore file contains N+2 comma separated numbers defining the longshore transport. Where N is the number of grids.

(*thetab*) breaking wave angle (deg).

(*alphab*) relative angle between beach normal and breaking wave (deg).

($Qspot$) potential longshore transport($\frac{m^3}{yr}$)

(Qs) actual longshore transport ($\frac{m^3}{yr}$)

3.3.5 offshore/intermediate and breaking wave files

```
variable,time,chainage(1),chainage(2)...
```

Where, the terms are defined as follows:

(*variable*) sequential list of wave parmaters. See below for list of variables.

(*time*) time of said parameter value (DD/MM/YYY HH:MM:SSS).

(*chainge*(1)) alongshore position of grid 1 (m).

(*chainge*(2)) alongshore position of grid 2 (m).

After the header, each subsequent row in the wave output file contains N+2 comma separated numbers defining the longshore transport. Where N is the number of grids.

($WvHt$) breaking wave angle (m)

($WvPer$) relative angle between beach normal and breaking wave (s).

($WvDir$) potential longshore transport(deg)

3.3.6 xsect and eq_xsect files

```
xsect_id,variable,time/point,1,2...
```

Where, the terms are defined as follows:

($xsect - id$) index of cross-shore profile ($integer$).

($variable$) horizontal and vertical coordinates of profile.

($time$) time of said parameter value (DD/MM/YYY HH:MM:SSS).

($1:30$) profile defined as by 30 pairs of x,z coordinates.

After the header, each subsequent row in the wave output file contains 33 comma separated values defining the requested profile.

(XA, ZA) note that in addition to the X and Y pairs, a second set of values are present to describe the shape of the in the presence of seawalls.

3.4 Details of *processed* files

This directory is intended for the post processing of the ascii Evo output files. These are matlab data structures that contain all the input and output data from the specified model run. For details on generation these files refer to the call_read_inputs and call_read_outputs calls respectively.

3.4.1 case_in.mat

This file is useful for both verifying the specified input to the model is as intended and reconstructing the Evo profiles from the output files.

3.4.2 case_out.mat

This file is useful for analysis of the Evo results. The format is intended to be a straightforward data structure suitable for beach erosion analysis.

4 Running the Evo model

The following is a description of the file organisation structure and work flow used in the BNHCRC project. This method could certainly be modified to suit a particular need, however, this approach worked well for the present application. Specifically the project applied the Evo model in a probabilistic framewok with job arrays on a super computer.

4.1 Evo file hierarchy and running.

Recall from Table Matlab Evo file structure. the required organisation of the Evo input files. The run_script sets the computers environment variables and calls the evoexe executable along with the control file instructions. However, for preparing the Evo input files and analising the outputs, the following scripts are helpful.

4.2 script hierarchy

```
evo/
├── fortran_code/
│   ├── bin/
│   │   └── evoexe
├── mfiles/
│   ├── calls/
│   │   ├── call_write_evo_inputs.m
│   │   ├── call_read_input.m
│   │   ├── call_show_input.m
│   │   ├── call_read_output.m
│   │   └── call_show_output.m
│   ├── cases/
│   │   └── name/
│   │       ├── mfiles/
│   │       │   └── case_name.m
│   │       └── matfiles/
│   │           └── case_name.mat
│   ├── functions/
│   │   ├── write/
│   │   │   ├── write_evo_grid.m
│   │   │   ├── write_evo_groyne.m
│   │   │   ├── write_evo_icond.m
│   │   │   ├── write_evo_seawall.m
│   │   │   ├── write_evo_transport.m
│   │   │   ├── write_evo_wave.m
│   │   │   ├── write_evo_wl.m
│   │   │   ├── write_evo_xpar.m
│   │   │   ├── write_xsect_ids.m
│   │   │   ├── write_xsect_ids_eq.m
│   │   │   └── write_nc_evo.m
│   │   ├── evo_profile
│   │   ├── evo_rd_output.m
│   │   ├── read_input.m
│   │   └── read_output.m
├── runs/
```

Table 2: Matlab Evo file structure.

4.2.1 fortran_code

Further, details related to both code compiling the fortran_code itself can be found in Installing the Evo model and System Documentation sections respectively. Running the model requires only the compiled executable *evoexe* found in the bin directory.

4.2.2 mfiles

The mfiles folder contains tools in the form of matlab scripts for creating and visualizing Evo model input and output files. Each specific test is developed as a case. Once the case is designed the files can then be written in ascii format readable by Evo using the 'call_write_evo_input.m' script. However, this is a good time to check the written files using 'call_read_input.m' and 'call_show_input.m'.

4.2.3 functions

These are not intended to be modified in any way as the are used as subroutines and left separate from the remaining "calls" and "cases".

4.2.4 calls

Generic call for reading and writing the Evo inputs and outputs.

4.2.5 cases

Specific call for designing run cases. Example case is provided. Sequence would be example_case.m followed by call to call_write_evo_inputs which creates the run directory will all required inputs with two expeceptions; the control file and the runscript. Both of these need to be modified from an existing run.

5 Installing the Evo model

5.1 Overview and prerequisits.

5.2 Windows compilation

5.3 Linux compilation

The code is written in Fortran with some Intel specifics and also requires compiling with NetCDF libraries.

5.3.1 Building Zlib

```
#!/bin/bash

source /opt/intel/Compiler/11.1/069/bin/ifortvars.sh intel64
source /opt/intel/Compiler/11.1/069/bin/iccvars.sh intel64

cd zlib-1.2.7/
export CC=icc
export CFLAGS='-O3 -xHost -ip -fPIC'
./configure --prefix=/usr/local/zlib-1.2.7
make
make check
make install
```

5.3.2 Building hdf5

```
#!/bin/bash

source /opt/intel/Compiler/11.1/069/bin/ifortvars.sh intel64
source /opt/intel/Compiler/11.1/069/bin/iccvars.sh intel64

cd hdf5-1.8.8/
export CC=icc
export F9X=ifort
export CXX=icpc
./configure --prefix=/usr/local/hdf5-1.8.8 --enable-fortran
   --enable-cxx
--with-zlib=/usr/local/zlib-1.2.7
make
make check
make install
```

5.3.3 Building netcdf

```
source /opt/intel/Compiler/11.1/069/bin/ifortvars.sh intel64
source /opt/intel/Compiler/11.1/069/bin/iccvars.sh intel64

cd netcdf-4.1.3/
export CC=icc
export CXX=icpc
export CFLAGS='-O3 -xHost -ip -no-prec-div -static-intel'
export CXXFLAGS='-O3 -xHost -ip -no-prec-div -static-intel'
export F77=ifort
export FC=ifort
export F90=ifort
export FFLAGS='-O3 -xHost -ip -no-prec-div -static-intel'
export CPP='icc -E'
export CXXCPP='icpc -E'
export CPPFLAGS="-I/usr/local/zlib-1.2.7/include -I/usr/local
   /hdf5-1.8.8/include"
```

```
export LDFLAGS="-L/usr/local/zlib-1.2.7/lib -L/usr/local/hdf5
    -1.8.8/lib"
./configure --disable-dap-remote-tests
make
make check
su
make instal
```

5.3.4 Building Evo

```
#!/bin/bash

source /opt/intel/Compiler/11.1/069/bin/ifortvars.sh intel64
source /opt/intel/Compiler/11.1/069/bin/iccvars.sh intel64

export PRECISION=1
export NETCDFINCL=/usr/local/include
export NETCDFLIB=/usr/local/lib
export NETCDFLIBNAME="-lnetcdff -lnetcdf"
export HDF5LIB=/usr/local/hdf5-1.8.8/lib
export HDF5LIBNAME="-lhdf5hl -lhdf5"
#export COMPILATIONMODE=production
export COMPILATIONMODE=debug
export FORTRANCOMPILER=ifort
cd /Desktop/evo/fortran_code/platform/linuxifort
env
make clean
make
```

6 System Documentation

BCmod.f90
EVOctrl.f90
EVO.f90
EVOmod.f90
EVOutils.f90
ERROR.F90
GEN_FILE.F90
GEN_GEO.F90
GEN_STRING.F90
GEN_UTIL.F90
LOG.F90
PRECISION.F90
globals.f90

```
LSmod.f90
OUTmod.f90
WVmod.f90
XSmod.f90
```

References

C Huxley. Shoreline response modelling assessing the impacts of climate change. In *18th NSW Coastal Conference, Ballina*. Citeseer, 2009.

C Huxley et al. Shoreline response to climate change: A new assessment approach. *Climate Change 2010: Practical Responses to Climate Change*, page 86, 2010.

Christopher Huxley. Quantification of the physical impacts of climate change on beach shoreline response, 2011.

Ian Teakle, Chris Huxley, Dean Patterson, Jesper Nielsen, Hamid Mirfenderesk, et al. Gold coast shoreline process modelling. In *Coasts and Ports 2013: 21st Australasian Coastal and Ocean Engineering Conference and the 14th Australasian Port and Harbour Conference*, page 751. Engineers Australia, 2013.

Uriah Gravois, David Callaghan, Tom Baldock, Katrina Smith, and Bronte Martin. Review of beach profile and shoreline models applicable to the statistical modelling of beach erosion and the impacts of storm clustering. 2016.

Per Bruun. *Coast erosion and the development of beach profiles*, volume 44. US Beach Erosion Board, 1954.

RG Dean. Equilibrium beach profiles: Us atlantic and gulf coasts. university of delaware, newark, delaware, usa, 1977. *Ocean Engineering Report*, (12), 1977.

Per Bruun. Sea-level rise as a cause of shore erosion. *Journal of the Waterways and Harbors division*, 88(1):117–132, 1962.

RG Dean. Equilibrium Beach Profiles - Characteristics and Applications. *JOURNAL OF COASTAL RESEARCH*, 7(1):53–84, WIN 1991. ISSN 0749-0208.

Robert G. Dean and Robert A. Dalrymple. *Coastal Processes with Engineering Applications*. Cambridge University Press, 2001. doi: 10.1017/CBO9780511754500.

DL Kriebel and RG Dean. NUMERICAL-SIMULATION OF TIME-DEPENDENT BEACH AND DUNE EROSION. *COASTAL ENGINEERING*, 9(3):221–245, 1985. ISSN 0378-3839. doi: {10.1016/0378-3839(85)90009-2}.

DL Kriebel and RG Dean. CONVOLUTION METHOD FOR TIME-DEPENDENT BEACH-PROFILE RESPONSE. *JOURNAL OF WATERWAY PORT COASTAL AND OCEAN ENGINEERING-ASCE*, 119 (2):204–226, MAR-APR 1993. ISSN 0733-950X. doi: {10.1061/(ASCE)0733-950X(1993)119:2(204)}.

JK Miller and RG Dean. A simple new shoreline change model. *COASTAL ENGINEERING*, 51(7):531–556, SEP 2004. ISSN 0378-3839. doi: {10.1016/j.coastaleng.2004.05.006}.