XBeach Model Description and Manual

Unesco-IHE Institute for Water Education, Deltares and Delft University of Technology

Report

June 21, 2010 version 6

XBeach Model Description and Manual

Unesco-IHE Institute for Water Education, Deltares and Delft University of Technology

Dano Roelvink (P.I.), Ad Reniers, Ap van Dongeren, Jaap van Thiel de Vries, Jamie Lescinski, Robert McCall

Contents

Abstract

XBeach is a numerical model of nearshore processes intended as a tool to compute the natural coastal response during time-varying storm and hurricane conditions, including dune erosion, overwash and breaching.

The model consists of formulations for short wave envelope propagation, nonstationary shallow water equations, sediment transport and bed update. Innovations include a newly-developed time-dependent wave action balance solver, which solves the wave refraction and allows variation of wave action in x, y, time and over the directional space, and can be used to simulate the propagation and dissipation of wave groups. An added advantage to this set-up, compared to the existing surfbeat (infragravity wave) model, is that a separate wave model is not needed to predict the mean wave direction, and it allows different wave groups to travel in different directions. Wave-current interaction in the short wave propagation is included. Various wave breaking dissipation model are implemented for use in the nonstationary and stationary wave energy balance (in other words, when the wave energy varies on the wave group timescale).

The Generalised Lagrangean Mean (GLM) approach was implemented to represent the depth-averaged undertow and its effect on bed shear stresses and sediment transport, cf. Reniers et al. (2004). Quasi 3D formulations are included as well as ground water flow through a porous medium.

Soulsby – Van Rijn transport formulations have been included, which solves the 2DH advection-diffusion equation and produces total transport vectors, which can be used to update the bathymetry. The pickup function follows Reniers et al (2004) was implemented. An avalanching routine was implemented with separate criteria for critical slope at wet or dry points providing a smooth and robust solution for slumping of sand during dune erosion.

Since length scales are short in terms of wave lengths and supercritical flow frequently occurs, the numerical implementation is mainly first order upwind, which in combination with a staggered grid makes the model robust. The momentum-conserving form of Stelling and Duinmeijer (2004) is applied which improves long-wave runup and backwash on the beach. The model scheme utilizes explicit schemes with an automatic time step based on Courant criterion, with output at fixed or user defined time intervals, which keeps the code simple and makes coupling and parallellization easier, while increasing stability.

The model has been validated with a series of analytical, laboratory and field test cases. The model performs well in different situations including dune erosion, overwash and breaching and these cases are all modelled using a standard set of parameter settings.

1   Introduction

1.1   Motivation

The devastating effects of hurricanes on low-lying sandy coasts, especially during the 2004 and 2005 seasons have pointed at an urgent need to be able to assess the vulnerability of coastal areas and (re-)design coastal protection for future events, but also to evaluate the performance of existing coastal protection projects compared to 'do-nothing' scenarios. In view of this the Morphos-3D project was initiated by USACE-ERDC, bringing together models, modelers and data on hurricane winds, storm surges, wave generation and nearshore processes. Concurrently, the Dutch Dune Safety Assessment required that advanced models be developed which can be used to assess the protection provided by dunes against flooding of the hinterland for situations where empirical model assumptions do not hold anymore. In the Netherlands, most of the central Holland Coast is protected by dunes. For these reasons an open-source program, XBeach for eXtreme Beach behavior, has been developed to model the nearshore response to hurricane impacts and storms.

Existing tools to assess dune erosion under extreme storm conditions assume alongshore uniform conditions and have been applied successfully along relatively undisturbed coasts (Vellinga, 1986, Steetzel, 1993, Nishi and Kraus, 1996, Larson et al., 2004), but are inadequate to assess the more complex situation where the coast has significant alongshore variability. This variability may result from anthropogenic causes, such as the presence of artificial inlets, sea walls, and revetments, but also from natural causes, such as the variation in dune height along the coast or the presence of rip-channels and shoals on the shoreface (Thornton et al., 2007). A particularly complex situation is found when barrier islands protect storm impact on the main land coast. In that case the elevation, width and length of the barrier island, as well as the hydrodynamic conditions (surge level) of the back bay should be taken into account to assess the coastal response. Therefore, the assessment of storm impact in these more complex situations requires a two-dimensional process-based prediction tool, which contains the essential physics of dune erosion and overwash, avalanching, swash motions, infragravity waves and wave groups.

With regard to dune erosion, the development of a scarp and episodic slumping after undercutting is a dominant process (van Gent et al., 2008). This supplies sand to the swash and surf zone that is transported seaward by the backwash motion and by the undertow; without it the upper beach scours down and the dune erosion process slows down considerably. One-dimensional (cross-shore) models such as DUROSTA (Steetzel, 1993) focus on the underwater offshore transport and obtain the supply of sand by extrapolating these transports to the dry dune. Overton and Fisher (1988), Nishi and Kraus (1996) focus on the supply of sand by the dune based on the concept of wave impact. Both approaches rely on heuristic estimates of the runup and are well suited for 1D application but difficult to apply in a horizontally 2D setting. Hence, a more comprehensive modeling of the swash motions is called for.

Swash motions are up to a large degree a result from wave group forcing of infragravity waves (Tucker, 1954). Depending on the beach configuration and directional properties of the incident wave spectrum both leaky and trapped infragravity waves contribute to the swash spectrum (Huntley et al., 1981). Raubenheimer and Guza (1996) show that incident band swash is saturated, infragravity swash is not, therefore infragravity swash is dominant in storm conditions. Models range from empirical formulations (e.g. Stockdon et al, 2006) through analytical approaches (Schaeffer, 1994, Erikson et al, 2005) to numerical models in 1D (e.g. List, 1992, Roelvink, 1993b) and 2DH (e.g. van Dongeren et al, 2003, Reniers et al, 2004a, 2006). 2DH wave group resolving models are well capable of describing low-frequency motions. However, for such a model to be applied for swash, a robust drying/flooding formulation is required.

1.2   Objective

The main objective of the XBeach model is to provide a robust and flexible environment in which to test morphological modeling concepts for the case of dune erosion, overwashing and breaching. The top priority is to provide numerical stability and robustness, while still providing accurate results in a reasonable computational time.

1.3   Context

The XBeach model can be used as stand-alone model for small-scale (project-scale) coastal applications, but will also be used within larger (shelf sea) systems, where it will be driven by boundary conditions provided by the wind, wave and surge models and its main output to be transferred back will be the time-varying bathymetry and possibly discharges over breached barrier island sections.

1.4   Model approach

Our aim is to model processes in different regimes as described by Sallenger (2000). He defines an Impact Level to denote different regimes of impact on barrier islands by hurricanes, which are the 1) swash regime, 2) collision regime, 3) overwash regime and 4) inundation regime. The approach we follow to model the processes in these regimes is described below:

To resolve the swash dynamics the model employs a novel 2DH description of the wave groups and accompanying infragravity waves over an arbitrary bathymetry (thus including bound, free and refractively trapped infragravity waves). The wave group forcing is derived from the time-varying wave action balance e.g. Phillips (1977) with a dissipation model for use in combination with wave groups (Roelvink, 1993a). A roller model (Svendsen, 1984, Nairn et al., 1990, Stive and de Vriend, 1994) is used to represent momentum stored in surface rollers which leads to a shoreward shift in wave forcing.

The wave-group forcing drives infragravity motions and both longshore and cross-shore currents. Wave-current interaction within the wave boundary layer results in an increased wave-averaged bed shear stress acting on the infragravity waves and currents (e.g. Soulsby et al., 1993 and references therein). To account for the randomness of the incident waves the description by Feddersen et al. (2000) is applied which showed good skill for longshore current predictions using a constant drag coefficient (Ruessink et al., 2001).

During the swash and collision regime the mass flux carried by the waves and rollers returns offshore as return flow or rip-current. These offshore directed flows keep the erosion process going by removing sand from the slumping dune face. Various models have been proposed for the vertical profile of these currents (see Reniers et al. (2004b) for a review). However, the vertical variation is not very strong during extreme conditions and has been neglected for the moment.

Surf and swash zone sediment transport processes are very complex, with sediment stirring by a combination of short-wave and long-wave orbital motion, currents and breaker-induced turbulence. However, intra-wave sediment transports due to wave asymmetry and wave skewness are expected to be relatively minor compared to long-wave and mean current contributions (van Thiel de Vries et al., 2008). This allows for a relatively simple and transparent formulation according to Soulsby – Van Rijn (Soulsby, 1997) in a short-wave averaged but wave-group resolving model of surf zone processes. This formulation has been applied successfully in describing the generation of rip channels (Damgaard et al., 2002 Reniers et al., 2004a) and barrier breaching (Roelvink et al., 2003).

In the collision regime, the transport of sediment from the dry dune face to the wet swash, i.e. slumping or avalanching, is modeled with an avalanching model accounting for the fact that saturated sand moves more easily than dry sand, by introducing both a critical wet slope and dry slope. As a result slumping is predominantly triggered by a combination of infragravity swash run-up on the previously dry dune face and the (smaller) critical wet-slope.

During the overwash regime the flow is dominated by low-frequency motions on the time-scale of wave groups, carrying water over the dunes. This onshore flux of water is an important landward transport process where dune sand is being deposited on the island and within the shallow inshore bay as overwash fans (e.g. Leatherman, 1977, Wang and Horwitz, 2007). To account for this landward transport some heuristic approaches exist in 1D, e.g. in the SBeach overwash module (Larson et al, 2004) which cannot be readily applied in 2D. Here, the overwash morphodynamics are taken into account with the wave-group forcing of low frequency motions in combination with a robust momumtum-conserving drying/flooding formulation (Stelling and Duinmeijer, 2003) and concurrent sediment transport and bed-elevation changes.

Breaching of barrier islands occurs during the inundation regime, where a new channel is formed cutting through the island. Visser (1998) presents a semi-empirical approach for breach evolution based on a schematic uniform cross-section. Here a generic description is used where the evolution of the channel is calculated from the sediment transports induced by the dynamic channel flow in combination with avalanche-triggered bank erosion.

To this end, the code has the following functionalities:

Flow

- Depth-averaged shallow water equations including time-varying wave forcing terms; combination of sub- and supercritical flows;
- Numerical scheme in line with Stelling and Duinmeijer method, to improve long-wave runup and backwash on the beach. The momentum-conserving form is applied, while retaining the simple first-order approach.
- Generalised Lagrangean Mean (GLM) approach to represent the depth-averaged undertow and its effect on bed shear stresses and sediment transport, cf. Reniers et al. (2004)
- Smagorinsky viscosity formulation
- Drifter (passive particle) option
- White-Colebrook roughness
- Quasi 3D formulation
- Automatic time step based on Courant criterion, with output at fixed or user-defined time intervals.
- Ground water flow
- Discharge boundaries
- Non-hydrostatic formulation
- MPI (Message Passing Interface) implementation with automatic domain decomposition for parallel (multi-processor) computing

Waves
- Time-varying wave action balance including refraction, shoaling, current refraction and wave breaking;
- Roller model, including breaker delay
- Wave amplitude effects on wave celerity;
- Wave-current interaction
- Roelvink (1993) wave dissipation model for use in the nonstationary wave energy balance (in other words, when the wave energy varies on the wave group timescale)
- Baldock et al. (1998) wave dissipation formulation for stationary wave energy balance.

Sediment transport and bed updating
- Depth-averaged advection-diffusion equation to solve suspended transport;
- Bed updating algorithm including possibility of avalanching;
- Soulsby – Van Rijn transport formulations, cf Reniers et al (2004).
- Multiple sediment fractions and bed layer bookkeeping.
- Intra-wave sediment transport
- Avalanching mechanism, with separate criteria for critical slope at wet or dry points.
- Hard structures

2   Model formulations

The model solves coupled 2D horizontal equations for wave propagation, flow, sediment transport and bottom changes, for varying (spectral) wave and flow boundary conditions. Because the model takes into account the variation in wave height in time (long known to surfers) it resolves the long wave motions created by this variation. This so-called 'surf beat' is responsible for most of the swash waves that actually hit the dune front or overtop it. With this innovation the XBeach model is better able to model the development of the dune erosion profile, to predict when a dune or barrier island will start overwashing and breaching and to model the developments throughout these phases.

2.1   Coordinate system

XBeach uses a coordinate system where the computational x-axis is always oriented towards the coast, approximately perpendicular to the coastline, and the y-axis is alongshore, see Figure 2.1 . This coordinate system is defined relative to world coordinates (x_w, y_w) through the origin (x_{ori}, y_{ori}) and the orientation alfa, defined counter-clockwise w.r.t. the x_w-axis (East). The grid size in x- and y-direction may be variable but the grid must be rectilinear.
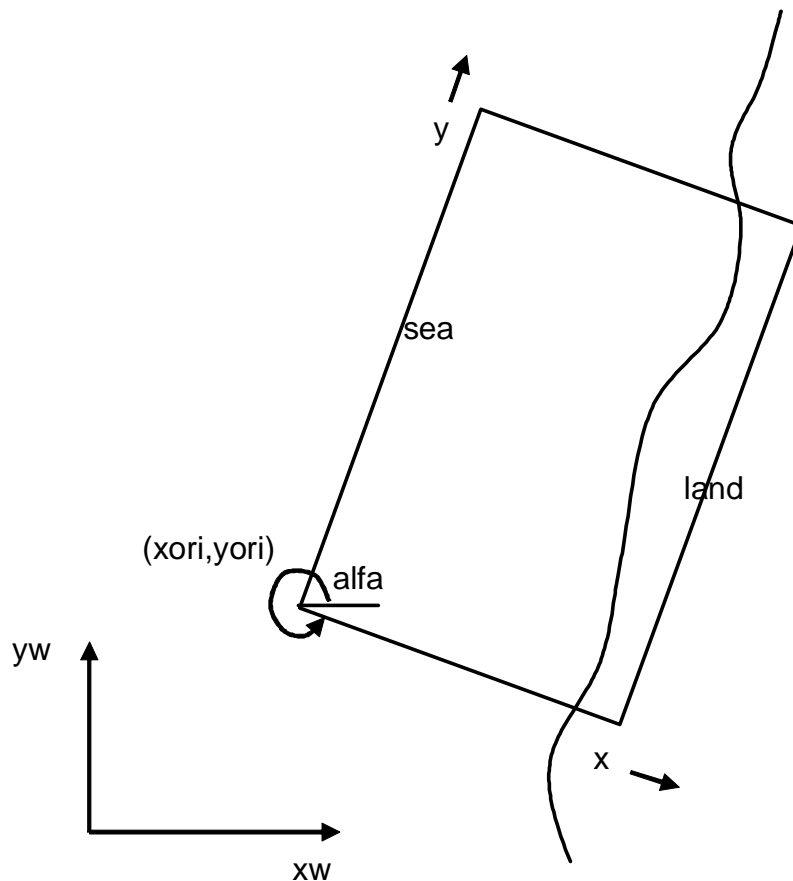
Figure 2.1   Coordinate system

2.2   Grid Setup

The grid applied is a staggered grid, where the bed levels, water levels, water depths and concentrations are defined in cell centers, and velocities and sediment transports are defined in u- and v-points, viz. at the cell interfaces. In the wave energy balance, the energy, roller energy and radiation stress are defined at the cell centers, whereas the radiation stress gradients are defined at u- and v-points, see Figure 2.2.

Velocities at the u- and v-points are denoted by uu and vv respectively; velocities u and v at the cell centers are obtained by interpolation and are for output purpose only. The water level, zs, and the bed level, zb, are both defined positive upward. uv and vu are the u-velocity at the v-grid point and the v-velocity at the u-grid point respectively. These are obtained by interpolation of the values of the velocities at the four surrounding grid points.

The model solves coupled 2D horizontal equations for wave propagation, flow, sediment transport and bottom changes, for varying (spectral) wave and flow boundary conditions. Because the model takes into account the variation in wave height in time (long known to surfers) it resolves the long wave motions created by this variation. This so-called 'surf beat' is responsible for most of the swash waves that actually hit the dune front or overtop it. With this innovation the XBeach model is better able to model the development of the dune erosion profile, to predict when a dune or barrier island will start overwashing and breaching and to model the developments throughout these phases.

Figure 2.2   Staggered grid

2.3   Wave action equation solver

The wave forcing in the shallow water momentum equation is obtained from a time dependent version of the wave action balance equation. Similar to Delft University's (stationary) HISWA model (Holthuijsen et al., 1989), the directional distribution of the action density is taken into account whereas the frequency spectrum is represented by a frequency, best represented by the spectral parameter $f_{m,-1,0}$.. The wave action balance is then given by:

$$\frac{\partial A}{\partial t} + \frac{\partial c_x A}{\partial x} + \frac{\partial c_y A}{\partial y} + \frac{\partial c_\theta A}{\partial \theta} = -\frac{D_w}{\sigma} \tag{2.1}$$

with the wave action:

$$A(x, y, t, \theta) = \frac{S_w(x, y, t, \theta)}{\sigma(x, y, t)} \tag{2.2}$$

where θ represents the angle of incidence with respect to the x-axis, S_w represents the wave energy density in each directional bin and σ the intrinsic wave frequency. The wave action propagation speeds in x- and y-direction are given by:

$$\begin{aligned} c_x(x, y, t, \theta) &= c_g \cos(\theta) + u^L \\ c_y(x, y, t, \theta) &= c_g \sin(\theta) + v^L \end{aligned} \tag{2.3}$$

With u^L and v^L the cross-shore and alongshore depth-averaged Lagrangian velocities respectively (defined below), and the group velocity c_g obtained from linear theory. If wave-current interaction is turned off (*wci=0*) then the last term in either equation is not taken into account.

The propagation speed in θ-space is obtained from:

$$c_\theta(x, y, t, \theta) = \frac{\sigma}{\sinh 2kh} \left(\frac{\partial h}{\partial x} \sin \theta - \frac{\partial h}{\partial y} \cos \theta \right) + \cos \theta \left(\sin \theta \frac{\partial u}{\partial x} - \cos \theta \frac{\partial u}{\partial y} \right) + \\ + \sin \theta \left(\sin \theta \frac{\partial v}{\partial x} - \cos \theta \frac{\partial v}{\partial y} \right) \tag{2.4}$$

taking into account bottom refraction (first term on the RHS) and current refraction (last two terms on the RHS) and h is the total water depth. If wave-current interaction is turned off (*wci=0*) then the last two terms are not taken into account.

The wave number k is obtained from the eikonal equations:

$$\frac{\partial k_x}{\partial t} + \frac{\partial \omega}{\partial x} = 0$$
$$\frac{\partial k_y}{\partial t} + \frac{\partial \omega}{\partial y} = 0$$

(2.5)

where the subscripts refer to the direction of the wave vector components and ω represents the absolute radial frequency. The wave number is then obtained from:

$$k = \sqrt{k_x^2 + k_y^2}$$

(2.6)

The absolute radial frequency is given by:

$$\omega = \sigma + k_x u^L + k_y v^L$$

(2.7)

and the intrinsic frequency is obtained from the linear dispersion relation. If wave-current interaction is turned off (*wci=0*) then the last two terms are not taken into account.

The total wave energy dissipation, i.e. directionally integrated, due to wave breaking is modelled according to Roelvink (1993a), which is coded as *break=1*:

$$\bar{D}_w = 2\frac{\alpha}{T_{rep}} Q_b E_w$$

$$Q_b = 1 - \exp\left(-\left(\frac{H_{rms}}{H_{max}}\right)^n\right), \quad H = \sqrt{\frac{8E_w}{\rho g}}, \quad H_{max} = \frac{\gamma \tanh kh}{k}$$

(2.8)

with $\alpha = O(1)$, ρ the water density, γ the breaker index (a free parameter) and the total wave energy is given by:

$$E_w(x, y, t) = \int_0^{2\pi} S_w(x, y, t, \theta) d\theta .$$

(2.9)

In a variation of the above, one could also state (*break = 3*)

$$\bar{D}_w = 2\frac{\alpha}{T_{rep}} Q_b E_w \frac{H_{rms}}{h}$$

(2.10)

Finally, Roelvink and Daly (*break = 4*) developed a breaking formulation based on (2.10) where the fraction of breaking waves is modelled as

$$\begin{cases} Q_b = 1 & if \quad H_{rms} > \gamma h \\ Q_b = 0 & if \quad H_{rms} < \gamma_2 h \end{cases} \tag{2.11}$$

which states that waves are fully breaking is the wave height exceeds a threshold and stop breaking if the wave height fall below another threshold. Details are in Daly and Roelvink (ICCE 2010).

In the stationary case, we apply Baldock et al. (1998) (*break = 2*) which states

$$\bar{D} = \frac{1}{4} \alpha Q_b \rho g f_{rep} \left(H_b^2 + H_{rms}^2 \right) \tag{2.12}$$

With $\alpha = O(1)$ and f_{rep} representing a representative intrinsic frequency. The fraction of breaking waves is given by:

$$Q_b = \exp\left[-\left(\frac{H_b^2}{H_{rms}^2} \right) \right] \tag{2.13}$$

Where the breaking wave height is:

$$H_b = \frac{0.88}{k} \tanh\left[\frac{\gamma k h}{0.88} \right] \tag{2.14}$$

And γ is a calibration parameter.

In either instationary or stationary case the total wave dissipation, $\bar{D}$, is distributed proportionally over the wave directions:

$$D_w(x, y, t, \theta) = \frac{S_w(x, y, t, \theta)}{E_w(x, y, t)} \bar{D}_w(x, y, t) \tag{2.15}$$

The bed friction dissipation is modelled as

$$D_f = \frac{2}{3} \rho \pi f_w \left(\frac{\pi H}{T_{rep} \sinh k h} \right)^3 \tag{2.16}$$

This closes the set of equations for the wave action balance. Given the spatial distribution of the wave action and therefore wave energy the radiation stresses can be evaluated (using linear wave theory):

$$S_{xx,w}(x,y,t) = \int\left(\frac{c_g}{c}\left(1+\cos^2\theta\right) - \frac{1}{2}\right)S_w d\theta$$

$$S_{xy,w}(x,y,t) = S_{yx,w} = \int \sin\theta\cos\theta\left(\frac{c_g}{c}S_w\right)d\theta \qquad (2.17)$$

$$S_{yy,w}(x,y,t) = \int\left(\frac{c_g}{c}\left(1+\sin^2\theta\right) - \frac{1}{2}\right)S_w d\theta$$

Instationary vs. stationary wave solver

For situations where infragravity motions play an important role the wave and roller equations must be solved in instationary, wave-group resolving mode (*instat=1, by default*). For some applications which do not focus on swash motions and where surfbeats are small, we have implemented an option (*instat=0*) to solve the stationary problem directly using a forward marching technique, where the equations are solved grid row by grid row in an iterative fashion. The wave module is then called every *wavint* seconds rather than each time step, which often means a large reduction in computation time.

2.3.1 Wave boundary conditions

Offshore wave boundary conditions

At present, a number of wave boundary conditions can be specified at the offshore boundary. These are numbered "instat=0" through "instat=41". The overview is as follows:

instat	abbreviated name	description
0	stat	stationary wave boundary condition (sea state)
1	bichrom	bichromatic (two wave component) waves
2	ts_1	first-order timeseries of waves (generated outside XBeach)
3	ts_2	second-order timeseries of waves (generated outside XBeach)
4	jons	wave groups generated using a parametric (Jonswap) spectrum
5	swan	wave groups generated using a SWAN 2D output file
6	vardens	wave groups generated using a formatted file
7	reuse	reuse of wave conditions
8	nonh	boundary conditions for nonhydrostatic option
9	off	no wave boundary condition
40	stat_table	a sequence of stationary conditions (sea states)
41	jons_table	a sequence of time-varying wave groups

In the params.txt file the boundary conditions options can be invoked using either the number ("instat=3" for instance) or by using the name "instat=ts_2". The name must be lower case and match exactly.

- Stationary wave boundary conditions (*instat* = 0).

In this case a uniform, constant wave energy distribution is set, based on given values of H_{rms}, T_{m01}, direction and power of directional distribution function.

$$
e_0(\vartheta) = E_{mean} \frac{\cos^m(\vartheta - \vartheta_m)}{\sum\limits_{\vartheta_{min}}^{\vartheta_{max}} \cos^m(\vartheta - \vartheta_m) \Delta\vartheta} \quad , |\vartheta - \vartheta_m| < \pi/2
$$

$$
E_{mean} = \frac{1}{8}\rho g H_{rms}^2
$$

(2.18)

- Wave energy varying periodically in time (*instat* = 1):

In this case regular wave groups (i.e. bichromatic waves) are specified.

$$
e_0(\vartheta) = E_{mean} \frac{\cos^m(\vartheta - \vartheta_m)}{\sum\limits_{\vartheta_{min}}^{\vartheta_{max}} \cos^m(\vartheta - \vartheta_m) \Delta\vartheta} * \frac{1}{2}\left(1 + \cos\left(2\pi\left(\frac{t}{T_{long}} - \frac{y}{L_{long}}\right)\right)\right),
$$

$$
L_{long} = \frac{c_g T_{long}}{\sin(\vartheta_m)}
$$

(2.19)

- First-order longcrested, irregular wave groups (*instat* = 2)

In this case E is read in as a function of time; the timeseries is shifted along the y-axis to account for the oblique incidence.

$$
e_0(\vartheta, y) = E_{t-\tau(y)} \frac{\cos^m(\vartheta - \vartheta_m)}{\sum\limits_{\vartheta_{min}}^{\vartheta_{max}} \cos^m(\vartheta - \vartheta_m) \Delta\vartheta},
$$

$$
\tau(y) = \frac{y\sin(\vartheta_m)}{c_g}
$$

(2.20)

This time series needs to be made by a separate routine (not part of XBeach). An example of the required input format is given in the Appendix B

- Second-order longcrested, irregular wave groups (*instat* = 3)

In this case a bound wave is added to the wave groups using Longuet-Higgins and Stewart's (1964) theory. The format is prescribed in Appendix B

- Standard JONSWAP spectrum, based on user-input spectrum coefficients (*instat* = 4)

With this option alongshore varying timeseries of the wave energy E and bound long wave z_s are generated on the basis of a specified analytical 2D Jonswap-type spectrum. With this option realistic second-order bound, directionally-spread seas can be created.

The spectrum is determined by the peak period, wave height, spectral peakedness, mean angle and directional spreading (see section 5.7 for details on the input values). The routine follows the procedure as outlined by Van Dongeren et al. (2003), see next paragraph.

- Unmodified SWAN 2D spectrum output file (*instat* = 5)

This option uses a SWAN 2D output file (sp2 file) in unmodified form. The procedure to calculate the boundary conditions is analoguous to the instat=4 option.

- Formatted variance-density spectrum file (*instat* = 6)

This option uses a formatted variance-density spectrum file, which needs to adhere to certain criteria (see section 5.5). On the basis of this formatted spectrum, boundary conditions are calculated using the procedure as outlined under *instat* = 4.

- Reuse boundary condition files from an earlier XBeach simulation (*instat* = 7)

If the user does not wish to recalculate spectrum-based boundary condition files or specifically wants to reuse the spectrum-based boundary condition files of another XBeach simulation, it is possible to do so. In this case the user should select '*instat* = 7' in params.txt. No further wave boundary condition data need be given in params.txt. Obviously, the calculation grid should remain the same between runs, as the angles and number of grid points are embedded in the boundary condition files. In order to use instat 7, the user should copy *ebcflist.bcf* and *qbcflist.bcf* to the current directory. Additionally, the user should also copy all files listed in *ebcflist.bcf* and *qbcflist.bcf*. Generally, these files have E_ and q_ prefixes.

With instat=4, 5 and 6 time-varying (on the scale of wave groups) boundary conditions are computed on the basis of stationary input spectra. It is also possible to use a sequence of varying spectra to compute boundary conditions which not only vary on the time scale of the wave groups but also have a variation on the longer timescale. The procedure is similar to the one described above, only the implementation is through the specification of a list of spectrum files, see section 5.7.

- Boundary conditions for non-hydrostatic model (*instat* = 8)

XBeach can be ran as a non-hydrostatic model, which is essentially the nonlinear shallow water equations with dispersion terms and without a wave-action driver. The boundary conditions are described in a separate chapter and are activated using instat=8

- No boundary condition (*instat* = 9)

This is a simple "no wave action" boundary condition. It still allows for a tidal record to be specified, however through the zs0file parameter.

- Sequence of stationary sea states (*instat = 40*)

While instat=0 specifies one stationary sea state (stationary in the sense that there are no wave groups), it is also possible to specify a series of seastates, each with a duration. This is done through a file as

instat = 40
bcfile = jonswap1.txt

where the name of the bcfile is free, but the structure of the contents is not. It should contain lines with
Hm0, Tp, angle, gamma (3.3), spreading, duration (s), timestep (=0.05)

Ie
0.7 8 90 3.3 5 1000 0.05
0.8 7 110 3.3 5 1000 0.05

- <u>Sequence of sea states to make time-varying wave groups (*instat = 41*).</u>

This is an extension of instat=4. With instat = 41 it is possible to specify a sea state on the basis of which wave groups are imposed on the model for a certain duration, then specify another sea state and run wave groups again without having to stop the model.

This condition is specified as
instat = 41
bcfile = jonswap1.txt

where the name of the file is free and the structure of the contents is as in instat = 40.

Procedure for converting spectra to wave energy and bound long wave boundary conditions

Using instat=4,5,6,7 40 or 41 measured or parametric spectra are used as input to create time-varying wave amplitudes, the envelopes of wave groups. This is done using a procedure as outlined by Van Dongeren et al. (2003).

In order to create a time series of wave energy along the offshore boundary, the input spectrum is assumed to be composed of K single summation wave components [Miles and Funke, 1989; in van Dongeren *et al*., 2003] in the range around the spectral peak where the energy density is greater than a certain fraction of the peak energy density (proscribed by the keyword "sprdthr"). Each wave component has a specific frequency, phase, amplitude and direction. Summed together the wave components create a time series of the sea surface at the offshore boundary:

$$\eta\left(0,y,t\right)=\sum_{i=1}^{K}B_{i}\cos\left(k_{i}\sin\left(\theta_{i}\right)y-2\pi f_{i}t+\varphi_{i}\right) \tag{2.21}$$

where B_i represents the amplitude of each wave component.

In order to determine the specific properties of the wave components, the frequencies of all K components are distributed uniformly in the range around the spectral peak. This choice leads to a frequency resolution which is dependent on K. Each wave component is given a wave phase using the random phase model. The direction of each wave component is determined randomly using the Cumulative Distribution Function of the wave direction of the input spectrum, see Figure 2.7. At this stage the directional CDF is based on integration across all frequencies. In the case of strong frequency-directional correlation, it may be advisable to use frequency dependent CDFs instead.
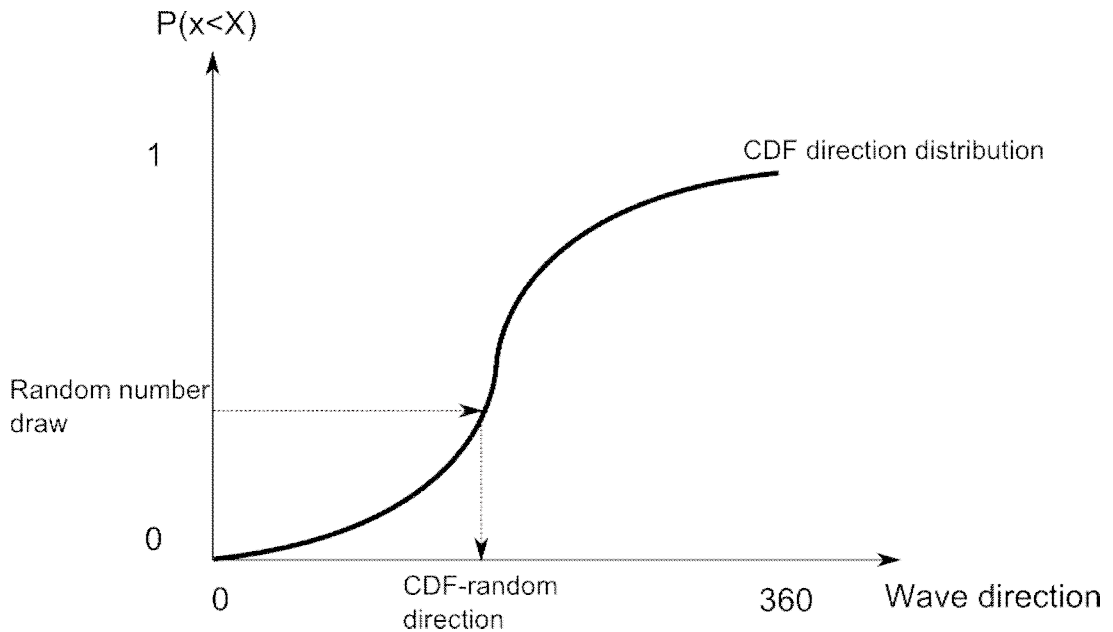
Figure 2.3 Random wave angles are chosen using a random number generator and the cumulative distribution function of the directional spreading of the input spectrum

Once the frequency and direction of each wave component has been selected, the amplitude Bi can be calculated by two dimensional interpolations across the 2D-input wave variance spectrum. A linear correction is made to the amplitudes of the wave components to ensure the integrated wave variance of the K components is the same as that of the input spectrum. The wave number k_i is determined using the dispersion relation, given the mean still water depth at the offshore boundary.

Equation (2.21) can now be used to generate a time series of the sea surface elevation at the offshore boundary. The envelope of the sea surface time series can be calculated using a Hilbert transform, the amplitude of which being a measure for the wave energy:

$$E_{\theta_{i}}\left(y,t\right)=\frac{1}{2}\rho g A_{\theta_{i}}\left(y,t\right) \tag{2.22}$$

The particle velocity due to bound infragravity waves at the offshore boundary is calculated according to the expressions developed by Herbers *et al*. [1994]. It is stated that bound infragravity waves are generated by the interaction of two wave components with different frequencies. The frequency of the bound infragravity wave is given as:

$$f_3 = f_2 - f_1 \qquad\qquad (2.23)$$

In the equation above, the subscripts on the right hand side refer to the indices of interacting short wave pairs. In order to ensure positive interaction frequencies, indices should be ordered according to increasing frequency of the short wave components. It should be noted that two interacting wave components contribute to only one infragravity wave frequency. However, one infragravity wave frequency may be forced by many different wave component interactions.

Similarly, other properties of the bound infragravity wave can be deduced from the associated properties of the short wave components. The bound wave number, wave group velocity and wave phase are given as:

$$k_3 \equiv \left| \vec{k_1} - \vec{k_2} \right| = \sqrt{k_1^2 + k_2^2 - 2k_1 k_2 \cos\left(\theta_3\right)} \qquad (2.24)$$

$$c_{g3} = \frac{2\pi f_3}{k_3} \qquad\qquad (2.25)$$

$$\varphi_3 = \varphi_2 - \varphi_1 + \pi \qquad\qquad (2.26)$$

Note that equation (2.26) is based on the assumption that the short wave groups and bound long waves are in equilibrium, and therefore are 180° out of phase. The angle of the bound long wave can be found using the following relation:

$$\theta_3 = \arctan\left(\frac{k_2 \sin\theta_2 - k_1 \sin\theta_1}{k_2 \cos\theta_2 - k_1 \cos\theta_1} \right) \qquad (2.27)$$

The energy related to a bound infragravity wave with a specific frequency is given as [van Dongeren et al., 2003]:

$$E_3\left(f_3\right) = 2 \int_{\Delta f}^{\infty} \int_0^{2\pi} \int_0^{2\pi} D^2\left(f + f_3, -f, \left|\theta_1 - \theta_2\right| + \pi\right) \\ \cdot E\left(f + f_3, \theta_1\right) E\left(f, \theta_2\right) d\theta_2 d\theta_1 df \qquad (2.28)$$

Where in equation (2.28) the first term behind the triple integral is the interaction coefficient as defined by Herbers et al.[1994]. This interaction coefficient is determined through a perturbation expansion of the Bernouilli equation, details are in original publication. In the wave boundary condition module a modification of the interaction coefficient is implemented to convert the output to surface level elevation instead of bed level pressure:

$$D_{surface} = D_{bed} \frac{\cosh\left(k_3 h\right)}{\cosh\left(k_1 h\right)\cosh\left(k_2 h\right)} \qquad\qquad (2.29)$$

The amplitude of each bound wave can be found from the bound wave energy [van Dongeren *et al*., 2003]:

$$A_3 = \sqrt{2E_3 df} \qquad (2.30)$$

Where *df* refers to the frequency resolution of the short waves, i.e. the frequency step used to generate all K frequency components around the peak of the short wave spectrum.

A time series for the cross shore water flux across the offshore boundary is generated by means of an Inverse Fourier Transform:

$$q_x(0,t) = IFFT\left[\sum_{i=1}^{K} \frac{A_{3,i}}{2} e^{-i\varphi_{3,i}} c_{g3,i} \cos\theta_{3,i} \right] \qquad (2.31)$$

This cross-shore flux is phase-shifted along the offshore boundary as:

$$q_x(y,t) = IFFT\left[\sum_{i=1}^{K} \frac{A_{3,i}}{2} e^{-i\varphi_{3,i}} c_{g3,i} \cos\theta_{3,i} e^{-ik_{y3,i}y} \right] \qquad (2.32)$$

The cross-shore flux (2.32) and the wave energy (2.22) are specified along the boundary.

Lateral wave boundary conditions

For the lateral boundary conditions we make the following reasonable assumptions for the incoming wave energy:

- In the stationary case, we assume that the alongshore gradient of the wave energy is zero; this means we copy the value of one row inside the domain to the boundary, for the directional bins where the direction is into the model domain;
- In the instationary case, we assume that the gradient along the crest of the wave group is zero. The direction of the crest is derived from the local mean wave direction and the values at the boundary are determined by interpolation between the two points on the row inside around a virtual point taken along the crest direction; in the figure, for example, the value at point (3,1) is interpolated from points (2,2) and (3,2).
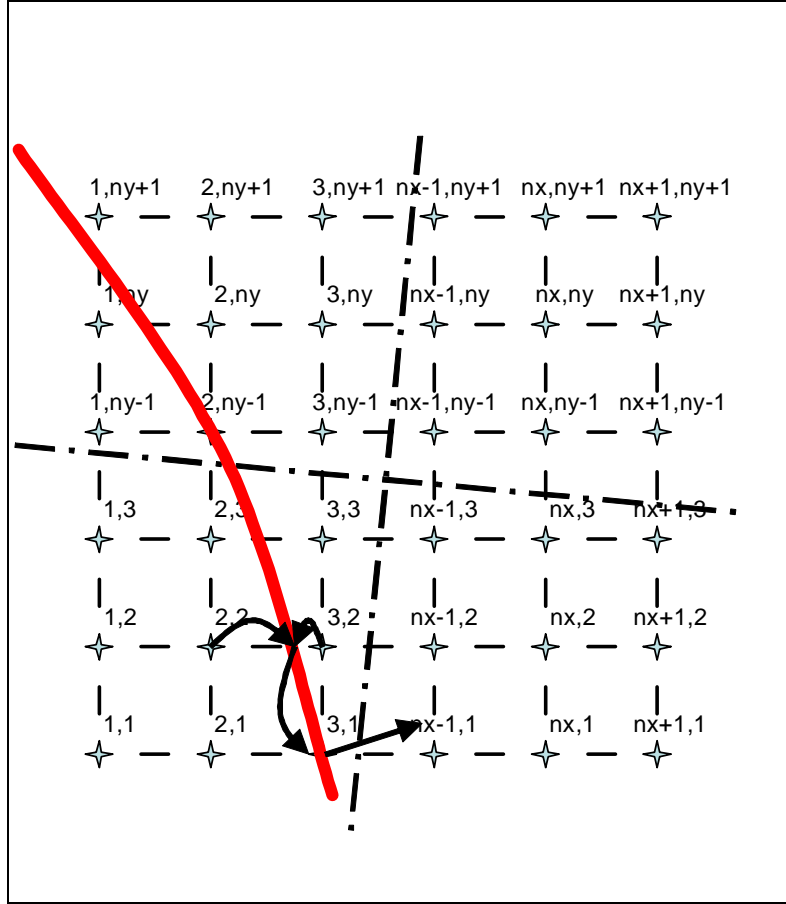
Figure 2.4   Example of interpolation at the lateral boundary.

2.4   Roller energy equation solver

The roller energy balance is coupled to the wave action/energy balance where dissipation of wave energy serves as a source term for the roller energy balance. Similar to the wave action the directional distribution of the roller energy is taken into account whereas the frequency spectrum is represented by a single mean frequency.

The roller energy balance is then given by:

$$\frac{\partial S_r}{\partial t} + \frac{\partial c_x S_r}{\partial x} + \frac{\partial c_y S_r}{\partial y} + \frac{\partial c_\theta S_r}{\partial \theta} = -D_r + D_w \tag{2.33}$$

with the roller energy in each directional bin represented by $S_r(x,y,t,\theta)$. The roller energy propagation speeds in x- and y-direction are given by:

$$\begin{aligned}
c_x(x,y,t,\theta) &= c\cos(\theta) + u^L \\
c_y(x,y,t,\theta) &= c\sin(\theta) + v^L
\end{aligned} \tag{2.34}$$

where θ represents the angle of incidence with respect to the x-axis. If wave-current interaction is turned off (*wci=0*) then the last terms in either equation are not taken into account.

The propagation speed in θ-space is identical to the expression used for the wave energy density propagation, (eq. (2.4), thus assuming that waves and rollers propagate in the same direction. The phase velocity is obtained from linear wave theory:

$$c = \frac{\sigma}{k} \tag{2.35}$$

The total roller energy dissipation is given by (Reniers et al., 2004a):

$$\bar{D}_r = \frac{2g\beta_r E_r}{c} \tag{2.36}$$

which combines concepts by Deigaard (1993) and Svendsen (1984)

Next the total roller dissipation, $\bar{D}_r$, is distributed proportionally over the wave directions:

$$D_r(x,y,t,\theta) = \frac{S_r(x,y,t,\theta)}{E_r(x,y,t)}\bar{D}_r(x,y,t) \tag{2.37}$$

This closes the set of equations for the roller energy balance.

The roller contribution to radiation stress is given by:

$$S_{xx,r}(x,y,t) = \int \cos^2\theta S_r d\theta$$
$$S_{xy,r}(x,y,t) = S_{yx,r}(x,y,t) = \int \sin\theta\cos\theta S_r d\theta \tag{2.38}$$
$$S_{yy,r}(x,y,t) = \int \sin^2\theta S_r d\theta$$

These roller radiation stress contributions are added to the wave-induced radiation stresses (eq. (2.17) to calculate the wave forcing utilizing the radiation stress tensor:

$$F_x(x,y,t) = -\left(\frac{\partial S_{xx,w} + S_{xx,r}}{\partial x} + \frac{\partial S_{xy,w} + S_{xy,r}}{\partial y}\right)$$
$$\tag{2.39}$$
$$F_y(x,y,t) = -\left(\frac{\partial S_{xy,w} + S_{xy,r}}{\partial x} + \frac{\partial S_{yy,w} + S_{yy,r}}{\partial y}\right)$$

2.5   Shallow water equations solver

2.5.1   Depth-averaged equations

For the low-frequency and mean flows we use the shallow water equations. To account for the wave induced mass-flux and the subsequent (return) flow these are cast into a depth-averaged Generalized Lagrangian Mean (GLM) formulation (Andrews and McIntyre, 1978, Walstra et al, 2000). In such a framework, the momentum and continuity equations are formulated in terms of the Lagrangian velocity, u^L, which is defined as the distance a water particle travels in one wave period, divided by that period. This velocity is related to the Eulerian velocity (the short-wave-averaged velocity observed at a fixed point) by:

$$u^L = u^E + u^S \quad and \quad v^L = v^E + v^S \tag{2.40}$$

Here u^S, v^S represents the Stokes drift in x- and y-direction respectively (Phillips, 1977):

$$u^S = \frac{E_w \cos\theta}{\rho hc} \quad and \quad v^S = \frac{E_w \sin\theta}{\rho hc} \tag{2.41}$$

where the wave-group varying short wave energy and direction are obtained from the wave-action balance (eq.(2.1). The resulting GLM-momentum equations are given by:

$$\frac{\partial u^L}{\partial t} + u^L \frac{\partial u^L}{\partial x} + v^L \frac{\partial u^L}{\partial y} - f\, v^L - v_h \left(\frac{\partial^2 u^L}{\partial x^2} + \frac{\partial^2 u^L}{\partial y^2} \right) = \frac{\tau_{sx}}{\rho h} - \frac{\tau_{bx}^E}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \tag{2.42}$$

$$\frac{\partial v^L}{\partial t} + u^L \frac{\partial v^L}{\partial x} + v^L \frac{\partial v^L}{\partial y} + f\, u^L - v_h \left(\frac{\partial^2 v^L}{\partial x^2} + \frac{\partial^2 v^L}{\partial y^2} \right) = +\frac{\tau_{sy}}{\rho h} - \frac{\tau_{by}^E}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \tag{2.43}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial hu^L}{\partial x} + \frac{\partial hv^L}{\partial y} = 0 \tag{2.44}$$

Here τ_{bx}, τ_{by} are the bed shear stresses, η is the water level, F_x, F_y are the wave-induced stresses, v_t is the horizontal viscosity and f is the Coriolis coefficient. The bottom shear stress terms are calculated with the Eulerian velocities as experienced by the bed:

$$u^E = u^L - u^S \quad and \quad v^E = v^L - v^S \tag{2.45}$$

and not with the GLM velocities. Also, the boundary condition for the flow computations are expressed in functions of (u^L, v^L) and not (u^E, v^E).

Smagorinsky viscosity

TEXTO BE INSERTED

White-Colebrook bottom roughness

TEXT TO BE INSERTED

2.5.2 Quasi-3D equations (*advanced option*)

NOT OPERATIONAL YET TO BE INSERTED

2.5.3 Flow boundary conditions

Offshore flow boundary conditions

Usually, the offshore boundary is an artificial boundary which has no physical meaning. On the offshore boundary wave and flow conditions are imposed. In the domain waves and currents will be generated which need to pass through the offshore boundary to the deep sea with minimal reflection. One way to do this is to impose a weakly reflective-type boundary condition.

The options are:

front	abbreviated name	description
0	abs1d	absorbing-generating (weakly-reflective) boundary in 1D
1	abs2d	same, in 2D (default setting)
2	wall	no flux wall
3	wlevel	water level specification (from file)
4	nonh_1d	boundary condition for nonhydrostatic option

1D absorbing-generating boundary condition

In XBeach, there are two options with regard to the offshore absorbing-generating boundary condition. With the parameter setting "front = 0" a simple one-dimensional radiating boundary condition is activitated.

It reads:

$$u = \left(1 + \frac{\sqrt{g\,h}}{c_g}\right) u_i + \overline{u} - \sqrt{\frac{g}{h}}(z_s - z_{s0}) \tag{2.46}$$

Where u_i is the incoming particle velocity and z_s is the surface elevation of the incoming bound long wave, and z_{s0} is the mean water level (averaged over many wave groups), $\overline{u}$ is the mean velocity (current). This boundary condition assumes all incoming and outgoing waves propagate normal to the boundary. It is therefore only useful for 1D (flume like) simulations.

2D absorbing-generating boundary condition

With option "front = 1" (default value) the formulation by Van Dongeren and Svendsen (1997) is activated which in turn is based on Verboom et al. (1981) and is based on the Method of Characteristics. This boundary condition allows for obliquely-incident and reflected waves, and is therefore suited for 1D and 2D computations.

The boundary conditions satisfy the following two necessary conditions:

1. the region outside the computation domain can influence the motion within the domain only through the incident (long) waves and through the currents along the boundaries; and
2. the (long) waves propagating out of the computational domain must be allowed to freely propagate through the open-ocean offshore boundary with minimal reflection.

By placing the open boundaries carefully, one can achieve weak local forcing near these boundaries. In practice this means that the offshore boundary is placed in sufficiently deep water, i.e. outside the shoaling zone. Then the dominant terms in the continuity and momentum equations near these boundaries are the nonlinear shallow water equation.s

For the general case of an arbitrary angle υ between the boundary at a point and the coordinate axes, one can follow the work of Abbott (1979) and Verboom et al. (1981) to derive the governing equations, which are valid for an arbitrary angle υ between the coordinate axes and the model boundary (Figure 2.5a).
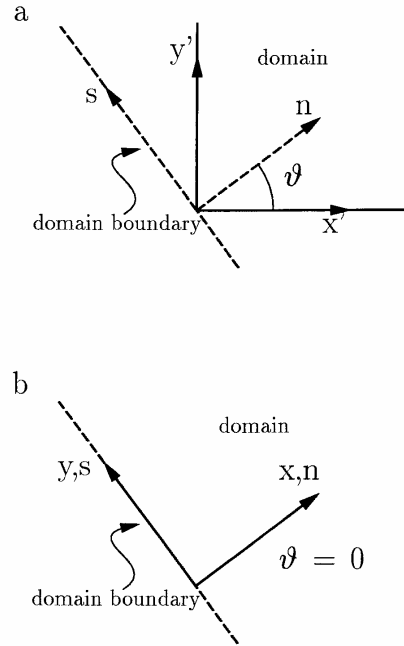
Figure 2.5   Coordinate system (a) for arbitrary angle υ between domain boundary and x-axis; (b) for υ=0

The derivation becomes simplified if the coordinate system is defined in a way the the x-axis is normally inward to the seaward boundary of the rectangular domain, which sets υ=0 (Figure 2.5b). The governing equations derived following Abbott (1979) and Verboom *et al.* (1981) then simplify to:

$$\frac{\partial \beta^-}{\partial t} = -(u-c)\frac{\partial \beta^-}{\partial x} - v\frac{\partial \beta^-}{\partial y} + c\frac{\partial v}{\partial y} + g\frac{\partial h_0}{\partial x} + F_{\beta^-} \tag{2.47}$$

$$\frac{\partial \beta^+}{\partial t} = -(u+c)\frac{\partial \beta^+}{\partial x} - v\frac{\partial \beta^+}{\partial y} - c\frac{\partial v}{\partial y} + g\frac{\partial h_0}{\partial x} + F_{\beta^+} \tag{2.48}$$

$$\frac{\partial \gamma}{\partial t} = -u\frac{\partial \gamma}{\partial x} - v\frac{\partial \gamma}{\partial y} - g\frac{\partial \eta}{\partial y} + F_\gamma \tag{2.49}$$

Where, F includes all local forcing and friction terms for the motion, c is the wave celerity, and h_0 is the still water depth. The Riemann variable β^- is defined as:

$$\beta^- = u - 2c = u - 2\sqrt{g(h_0 + \eta)} \tag{2.50}$$

Here u is the depth-averaged velocity. The Riemann variable β+ is similarly defined as $\beta^+ = u + 2c$. The γ-equation is the y-momentum equation, which has the Riemann variable:

$$\gamma = v \tag{2.51}$$

The definition sketch in Figure 2.6 shows that β- propagates in the negative *x*-direction, β+ propogates in the positive *x*-direction, and γ in the *y*-direction.

The forcing terms, F, in equations (2.47)-(2.49) originate from the right-hand side of the nonlinear shallow water equations, which imply that β- , β+, and γ are variables rather than constants.
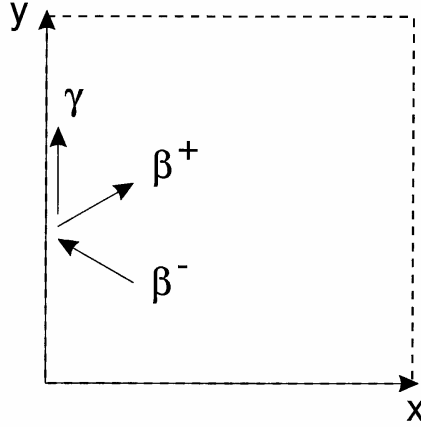
Figure 2.6   Definition sketch of the characteristics.

The offshore boundary conditions uses the outgoing β- variant which contains information about the waves leaving the domain and the γ variant which propagates along the boundary. The latter is extra information which we will use to estimate the direction of the outgoing wave which is the innovation in Van Dongeren and Svendsen (1997).

The procedure is as follows: during the computation, at time step n, we know the values of η^n and the total velocities (u^n, v^n) at all points in the domain. The incoming wave is specified along the open boundaries through the x and y components of the particle velocities of the incident wave (u_{in}, v_{in}). The numerical integration of nonlinear shallow water equations will provide the values of the total η , u, and v for the interior points in the domain at time step $n+1$, and then the equivalent total (incoming plus outgoing) values need to be determined along the boundaries at $n+1$. In other words, given the incoming wave, the outgoing wave needs to be determined.

In XBeach the lowest-order derived equations are implemented for the weakly reflective boundary conditions, with x=0 at the boundary. The outgoing wave angle (θ_r) and velocity in the x-direction (u_r) are solved iteratively. For specifics on this derivation we refer to Van Dongeren and Svendsen (1997), the shorter outline is given below

The β- is updated along the boundary only through (2.47) which discretized in XBeach (similarly as the x-momentum equation) reads

$$\frac{\beta_{i,j}^{n+1} - \beta_{i,j}^{n}}{\Delta t} = -(u-c)\frac{\partial \beta^n}{\partial x}\bigg|_{i,j} - v\frac{\partial \beta^n}{\partial y}\bigg|_{i,j} + c\frac{\partial v^n}{\partial y}\bigg|_{i,j} + g\frac{h_{i+1,j}^{n} - h_{i,j}^{n}}{x_{i+1,j} - x_{i,j}} + \frac{F_{x,i,j}}{\rho h_{u,i,j}} \qquad (2.52)$$

And is thus known at the time level $n+1$. We can then solve for the outgoing velocity u_r by expanding the Riemann variant (2.50) to lowest order as

$$\beta^- = u - 2\sqrt{g(h_0 + \eta)} = u - 2\sqrt{gh_0}\left(1 + \frac{1}{2}\frac{\eta}{h_0}\right) \tag{2.53}$$

We further have the identities

$$
\begin{aligned}
u &= u_i + u_r + \overline{u} \\
\eta &= \eta_i + \eta_r \\
u_i &= c_g \eta_i \cos\theta_i \\
u_r &= -\sqrt{gh_0}\,\eta_r \cos\theta_r
\end{aligned}
\tag{2.54}
$$

where the last two identities assume waves propagating with constant form. The incoming bound wave propagates at the group speed and the reflected wave at the free shallow water celerity, where θ_i and θ_r are the angles of the incoming (known) wave and the outgoing (yet unknown) wave, relative to the x=0 boundary. $\overline{u}$ is the mean current. Inserting these identities into (2.53) and re-arranging gives:

$$u_r = \left(\frac{\cos\theta_r}{\cos\theta_r + 1}\right)\left[\beta^- - \overline{u} + 2\sqrt{gh_0} - u_i\left(\frac{c_g \cos\theta_i - \sqrt{gh_0}}{c_g \cos\theta_i}\right)\right] \tag{2.55}$$

All the terms on the right hand side are known except for θr which can be solved from the γ = v variant as:

$$\theta_r = \arctan\left(\frac{u_r}{v_r}\right) = \arctan\left(\frac{u_r}{v - v_i - \overline{v}}\right) \tag{2.56}$$

Eqs. (2.55) and (2.56) are then solved iteratively to yield both u_r and θ_r. The final boundary condition is then the total velocity u $u = u_i + u_r + \overline{u}$ at the boundary at the time level *n+1*.

Wall boundary condition

This boundary condition type is activated using "front =2" and is a simple no flux boundary condition.

Water level boundary condition

This boundary sets the water level at a prescribed value. This can be constant or time-varying, see description of tide varying surge in Chapter 5. With this option the outgoing long waves are not absorbed.

Non-hydrostatic boundary condition

This option is to be used with the non-hydrostatic mode, see Appendix

Bay-side flow boundary condition

In many cases, the onshore boundary will be a land boundary. In some case the "onshore" will be a bay behind a barrier island. In that case it is possible to specify a number of boundary conditions.

The options are:

back	abbreviated name	description
0	wall	no flux wall
1	abs1d	absorbing-generating (weakly-reflective) boundary in 1D
2	abs2d	same, in 2D (default setting)
3	wlevel	water level specification (from file)

1D absorbing boundary condition

This boundary condition type is activated using "back = 0' and is analoguous to (2.46) without the possibility of specifying an incoming particle velocity.

Wall boundary condition

This boundary condition type is activated using "back =1" and is a simple no flux boundary condition

2D absorbing boundary condition (default)

This boundary condition type is activated using "back = 2" and is analoguous to the absorbing-generating boundary condition by Van Dongeren and Svendsen (1997), but without the incoming particle velocities (so 2D absorbing only).

Water level boundary condition

See above.

Lateral flow boundary conditions
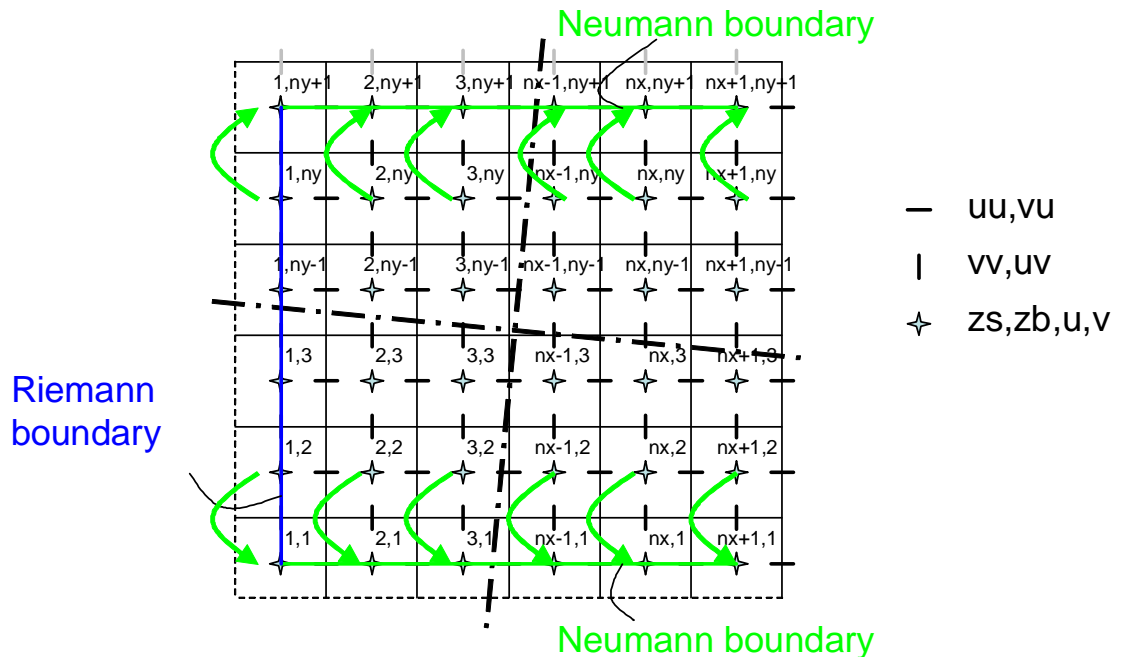
Figure 2.7   Stencil for Neumann-type boundary conditions.

Neumann boundaries (default)

Lateral boundaries are the boundaries perpendicular to the coastline. Usually these are artificial, because the model domain is limited but the physical coast will continue. At these boundaries we need to prescribe information about the area beyond the numerical model domain in such a way that the boundary condition does not influence the results in an adverse way. The best way to do this is to prescribe so-called "no-gradient" or Neumann boundaries, which state that there is locally no change in surface elevation and velocities.

These boundary conditions are activitate using "left = 0" or equivalently "left=neumann" and/or "right = 0" or equivalently "right=neumann"), where the longshore water level gradient is prescribed. The alongshore gradient is prescribed by the difference in specified water levels at the offshore corner points, divided by the alongshore length of the domain.

This type of Neumann boundary condition has been shown to work quite well with (quasi-) stationary situations, where the coast can be assumed to be uniform alongshore outside the model domain. So far we have found that also in case of obliquely incident wave groups this kind of boundary conditions appears to give reasonable results, though rigorous testing still has to be done. The implementation consists of copying water levels from row 2 to row 1 and from row *ny* to row *ny+1*, and doing the same for the cross-shore (along-boundary)

velocities. The alongshore velocities can now be computed from row 1 through row *ny*; no additional boundary conditions are required for the alongshore velocity.

Simple no-flux boundary conditions (walls) can be set using "left = 1" (or left = wall) and/or "right = 1" or "right = wall". Wall boundary conditions are preferred over Neumann boundary conditions in 1D (cross-shore) models.

2.6   Sediment transport

Advection-diffusion scheme

The sediment transport is modeled with a depth-averaged advection diffusion equation [Galappatti and Vreugdenhil, 1985]:

$$\frac{\partial hC}{\partial t} + \frac{\partial hCu^E}{\partial x} + \frac{\partial hCv^E}{\partial y} + \frac{\partial}{\partial x}\left[D_h h \frac{\partial C}{\partial x}\right] + \frac{\partial}{\partial y}\left[D_h h \frac{\partial C}{\partial y}\right] = \frac{hC_{eq} - hC}{T_s} \qquad (2.57)$$

where C represents the depth-averaged sediment concentration which varies on the wave-group time scale, and D_h is the sediment diffusion coefficient. The entrainment of the sediment is represented by an adaptation time T_s, given by a simple approximation based on the local water depth, h, and sediment fall velocity w_s:

$$T_s = \max\left(0.05\frac{h}{w_s}, 0.2\right) s \qquad (2.58)$$

where a small value of T_s corresponds to nearly instantaneous sediment response. The entrainment or deposition of sediment is determined by the mismatch between the actual sediment concentration, C, and the equilibrium concentration, C_{eq}, thus representing the source term in the sediment transport equation.

The bed-updating is discussed next. Based on the gradients in the sediment transport the bed level changes according to:

$$\frac{\partial z_b}{\partial t} + \frac{f_{mor}}{(1-p)}\left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y}\right) = 0 \qquad (2.59)$$

where p is the porosity, f_{mor} is a morphological acceleration factor of O(1-10) (e.g. Reniers et al., 2004a) and q_x and q_y represent the sediment transport rates in x- and y-direction respectively, given by:

$$q_x(x,y,t) = \left[\frac{\partial hCu^E}{\partial x}\right] + \left[\frac{\partial}{\partial x}\left[D_h h \frac{\partial C}{\partial x}\right]\right] \qquad (2.60)$$

and

$$q_y(x,y,t) = \left[\frac{\partial hCv^E}{\partial y}\right] + \left[\frac{\partial}{\partial y}\left[D_h h \frac{\partial C}{\partial y}\right]\right] \qquad (2.61)$$

To account for bed-slope effects on sediment transport a bed-slope correction factor f_{slope} is introduced.

Transport formulations

The equilibrium sediment concentration can be calculated with various sediment transport formulae. At the moment the sediment transport formulation of Soulsby-van Rijn (Soulsby, 1997) has been implemented. The C_{eq} is then given by:

$$C_{eq} = \frac{A_{sb} + A_{ss}}{h} \left(\left(|u^E|^2 + 0.018 \frac{u_{rms}^2}{C_d} \right)^{0.5} - u_{cr} \right)^{2.4} (1 - \alpha_b m) \tag{2.62}$$

where sediment is stirred by the Eulerian mean and infragravity velocity in combination with the near bed short wave orbital velocity, u_{rms}. In the default mode, the sediment is stirred due to mean and infragravity velocities. By setting $lws = 0$ the mean component can be excluded. The shortwave stirring can be turned off by setting $sws = 0$. By default $sws = 1$. The u_{rms} obtained from the wave-group varying wave energy using linear wave theory as

$$u_{rms} = \frac{\pi H_{rms}}{T_{rep} \sqrt{2} \sinh(kh + \delta H_{rms})} \tag{2.63}$$

The combined mean/infragravity and orbital velocity have to exceed a threshold value, u_{cr}, before sediment is set in motion. The drag coefficient, C_d, is due to flow velocity only (ignoring short wave effects). To account for bed-slope effects on the equilibrium sediment concentration a bed-slope correction factor is introduced, where the bed-slope is denoted by m and α_b represents a calibration factor. The bed load coefficients A_{sb} and the suspended load coefficient A_{ss} are functions of the sediment grain size, relative density of the sediment and the local water depth (see Soulsby (1997) for details). Note that the transport model does not contain transport contributions related to wave skewness.

The Soulsby-Van Rijn formulation is not strictly valid for sheet flow conditions. If applied in high velocity situations, the formulation as used in XBeach leads to unrealistically high sediment transport rates. In order to compensate this, steady flow velocities used to mobilize sediment are limited by an upper-bound Shields parameter for the start of sheet flow ($\theta_{sf} = 0.8 - 1.0$):

$$u_{flow,stirring}^2 = \min \left(\left(u^E \right)^2 + \left(v^E \right)^2, \theta_{sf} \frac{g D_{50} \Delta}{c_f} \right) \tag{2.64}$$

This approach assumes that in sheet flow conditions higher velocities lead to higher sediment transport rates, but not to higher equilibrium sediment concentrations, which is not necessarily correct. However, the assumption does cause sediment discharge under sheet flow conditions to become a linear function of flow discharge, which is in line with Kobayashi et al. (1996).

2.6.1 Wave asymmetry

The wave asymmetry enters the advection-diffusion equation, repeated here:

$$\frac{\partial hC}{\partial t} + \frac{\partial hCu_{AV}}{\partial x} + \frac{\partial hCv_{AV}}{\partial y} + \frac{\partial}{\partial x}\left[D_h h \frac{\partial C}{\partial x} \right] + \frac{\partial}{\partial y}\left[D_h h \frac{\partial C}{\partial y} \right] = \frac{hC_{eq} - hC}{T_s} \tag{2.65}$$

where

$$\begin{aligned} u_{AV} &= V_W \cos\theta_m + u^E \\ v_{AV} &= V_W \sin\theta_m + v^E \end{aligned} \tag{2.66}$$

with θm is the mean wave angle and

$$V_W = \gamma_{ua} u_{rms} \left(S_k - A_s \right) \tag{2.67}$$

is the velocity amplitude, with γua the free parameter, keyword facua. and

$$\begin{aligned} S_k &= \frac{0.79}{1 + \exp\dfrac{-0.61 - \log U_r}{-0.35}} \cos\left(-\frac{\pi}{2} + \frac{\pi}{2}\tanh\left(0.64 / U_r^{0.60}\right) \right) \\[2em] A_s &= \frac{0.79}{1 + \exp\dfrac{-0.61 - \log U_r}{-0.35}} \sin\left(-\frac{\pi}{2} + \frac{\pi}{2}\tanh\left(0.64 / U_r^{0.60}\right) \right) \end{aligned} \tag{2.68}$$

the skewness and asymmetry as parameterized as a function of the Ursell number by Ruessink and Van Rijn.

2.7 Morphological updating

2.7.1 Avalanching

To account for the slumping of sandy material during storm-induced dune erosion avalanching is introduced to update the bed-evolution. Avalanching is introduced when a user-defined critical bed-slope (keywords: wetslp and dryslp) is exceeded:

$$\left| \frac{\partial z_b}{\partial x} \right| > m_{cr} \tag{2.69}$$

Where the estimated bed slope is given by:

$$\frac{\partial z_b}{\partial x} = \frac{z_{b,i+1,j} - z_{b,i,j}}{\Delta x} \tag{2.70}$$

The bed-change within one time step is then given by:

$$\Delta z_b = \min\left(\left(\left|\frac{\partial z_b}{\partial x}\right| - m_{cr}\right)\Delta x, \quad 0.05\Delta t \right) \quad, \frac{\partial z_b}{\partial x} > 0$$

$$\Delta z_b = \max\left(-\left(\left|\frac{\partial z_b}{\partial x}\right| - m_{cr}\right)\Delta x, -0.05\Delta t \right) \quad, \frac{\partial z_b}{\partial x} < 0$$

(2.71)

Where a threshold of 0.05 m/s has been introduced to prevent the generation of large shockwaves. The corresponding bed update is given by:

$$z_{b,i,j}^{n+1} = z_{b,i,j}^{n} + \Delta z_{b,i,j}$$
$$z_{b,i+1,j}^{n+1} = z_{b,i+1,j}^{n} - \Delta z_{b,i,j}$$

(2.72)

To account for continuity, e.g. when sand is deposited within the wet part of the domain, the water level is also updated:

$$z_{s,i,j}^{n+1} = z_{s,i,j}^{n} + \Delta z_{b,i,j}$$
$$z_{s,i+1,j}^{n+1} = z_{s,i+1,j}^{n} - \Delta z_{b,i,j}$$

(2.73)

Similar expressions are used for the subsequent avalanching in the y-direction. Here we consider that inundated areas are much more prone to slumping and therefore we apply separate critical slopes for dry and wet points; default values are 1 and 0.3, respectively. The former value is consistent with the equilibrium profile according to Vellinga (1986); it is higher than the angle of natural repose and must be seen as an average slope observed after dune erosion, where some stretches may exhibit vertical slopes and other, drier parts may have slumped further. The underwater critical slope is much lower, and our estimate is based on the maximum underwater slopes we have observed in experiments, e.g. the Zwin test (see below) and tests carried out at Oregon State University with initially rather steep profiles.

When the critical slope between two adjacent grid cells is exceeded, sediment is exchanged between these cells to the amount needed to bring the slope back to the critical slope. This exchange rate is limited by a user-specified maximum avalanching transport rate, which for sandy environments is usually set so high as to have no influence on the outcome, while ensuring numerical stability.

In our model simulations, the avalanching mechanism is typically triggered when a high infragravity wave reaches the dune front and partly inundates it. The critical underwater slope is suddenly exceeded and the two grid cells at the dune foot are adjusted during the first timestep when this happens. In subsequent timesteps a chain reaction may take place both in points landward, where now the critical dry slope may be exceeded because of the lowering of the last wet point, and in points seaward, where now the critical wet slope may be exceeded. As a result, sediment is brought from the dry dune into the wet profile, where it is transported further seaward by undertow and infragravity backwash. An essential difference with similar procedures in other dune erosion models is the fact that avalanching is only applied between adjacent grid cells, rather than extrapolating profile behaviour well

beyond the wet domain. This is made possible by explicitly resolving the long-wave swash motions. Another big advantage with respect to existing procedures is that the simple avalanching algorithm is readily applied in two dimensions.

2.7.2 Morfac options

The morfac is the morphological acceleration factor that speeds up the morphological time scale relative to the hydrodynamic timescale. It means that if you have a simulation of 10 minutes with a morfac of 6 you effectively simulate the morphological evolution over one hour. There are now two ways in which you can input the time-varying parameters in combination with morfac; this is governed by the input option morfacopt.

morfacopt=1

All times are prescribed on input in morphological time. If you apply a morfac all input timeseries and other time parameters are divided internally by morfac. This way, you can specify all timeseries as real times, and vary the morfac without changing the rest of the input files. A typical application is that you run over a storm of some days and specify time-varying water level and wave conditions. If you now specify a morfac of 6, the model just runs for 10 (hydrodynamic) minutes each hour, during which the bottom changes per step are multiplied by a factor 6. This of course saves a factor of 6 in computation time.

An important thing to note is that the can only be done as long as the water level changes that are now accelerated by morfac do not modify the hydrodynamics too much. This is the case if the tide is perpendicular to the coast and the vertical variations do not lead to significant currents. If you have an alongshore tidal current, as is the case in shallow seas, you cannot apply this method because you would affect the inertia terms and thus modify the tidal currents.

morfacopt=0

In this new option the philosophy is different: you run the model over, say, a tidal cycle and apply the morfac without modifying the time parameters. This means you leave all the hydrodynamic parameters unchanged and just exaggerate what happens within a tidal cycle. As long as the evolution over a single tidal cycle is limited, the mean evolution over a tidal cycle using a morfac is very similar to running morfac tidal cycles without morfac. See Roelvink, 2006 for a more detailed description of this approach. This method is more appropriate for longer-term simulations with not too extreme events.

2.8   Multiple sediment fractions (*advanced option*)

To model the overwash deposits at barrier islands during extreme conditions XBEACH has been extended with a multiple sediment class formulation. This allows for the tracking of sediment but also for assigning different sediment characteristics such as grain size diameter, fall velocity, mobility, etc. For each sediment class, i, the equilibrium sediment

concentration, $c_{eq}^*(i)$, is calculated according to the Soulsby-van Rijn formulation, see equation (2.59).

The actual concentration then depends on the mismatch with the equilibrium concentration in combination with the available fraction at that location. It is assumed that a top-layer of 10 cm depth is readily available for sediment pick-up. So based on the fractions of the various sediment classes present in the top-layer the equilibrium concentration per sediment class can be expressed as:

$$c_{eq}(i) = frc(i,1)c_{eq}^*(i) \qquad (2.74)$$

Where the index *1* refers to the top layer and frc the fraction of a specific sediment class. Next the advection-diffusion equation (see eq. 2.54) is solved independently for the different sediment classes leading to class dependent sediment transport rates, S_i, from which the bottom changes per sediment class, Δz_i, can be derived:

$$\Delta z_i = \frac{\Delta t}{1-n_p} \left[\frac{\partial S_{i,x}}{\partial x} + \frac{\partial S_{i,y}}{\partial y} \right] \qquad (2.75)$$

Changes in fractional composition of the sediment classes in the top-layer due to sediment deposition are then calculated by:

$$frc^{n+1}(i,1) = \frac{\Delta z_i}{D_z} + \frac{D_z - \Delta z}{D_z} frc^n(i,1) \qquad (2.76)$$

Given dz <= Dz, else:

$$frc^{n+1}(i,1) = \frac{\Delta z_i}{\Delta z} \qquad (2.77)$$

And similarly for erosion:

$$frc^{n+1}(i,1) = \frac{\Delta z_i + D_z}{D_z} frc^n(i,1) - \frac{\Delta z}{D_z} frc^n(i,2) \qquad (2.78)$$

Where the number 2 refers to the layer immediately below the top layer. D_z is the constant layer thickness of 10 cm and Δz is the total change in bed elevation (all classes combined and positive upward) at time step *n+1* where n represent the time index. Next the underlying layers are updated according to:

$$frc^{n+1}(i,j) = \frac{D_z + \Delta z}{D_z} frc^n(i,j) - \frac{\Delta z}{D_z} frc^n(i,j+1) \qquad (2.79)$$

For erosion and:

$$frc^{n+1}(i,j) = \frac{D_z - \Delta z}{D_z} frc^n(i,j) + \frac{\Delta z}{D_z} frc^n(i,j-1) \qquad (2.80)$$

During sedimentation where the subscript j refers to the individual layers. In case of erosion, sediment is thus moving from the bottom layers towards the top layer and vice versa.
.

2.9 Hard Layers (*advanced option*)

<mark>TEXT TO BE INSERTED</mark>

2.10 Groundwater flow (*advanced option*)

2.10.1 Physical and numerical principles

The groundwater module in XBeach utilizes the principle of Darcy flow and is therefore limited to laminar flow conditions. In situations in which the groundwater flow may become turbulent, the full momentum equations (e.g. van Gent, 1995) should be applied. The module includes a vertical interaction flow between the surface water and groundwater. This flow is assumed to be a magnitude smaller than horizontal flow and is not incorporated in the momentum balance.

2.10.2 Determining groundwater head

The driving force behind groundwater flow according to Darcy is the groundwater head gradient. In the XBeach module, the groundwater head p_{gw} has the unit [m]. Where there is no surface water, the groundwater head is equal to the groundwater surface level η_{gw}:

$$\left[p_{gw} \right]_{i,j}^{n} = \left[\eta_{gw} \right]_{i,j}^{n-1} \qquad\qquad \text{if } wetz_{i,j}^{n-1} = 0 \qquad\qquad (2.81)$$

Where there is surface water and the groundwater surface level is just below the surface of the bed z_b, the groundwater head is affected by the surface water head z_s. If the groundwater surface level is equal to the bed level, the groundwater head is equal to the surface water head. If the groundwater surface level is more than $d_{wetlayer}$ below the surface of the bed, the groundwater head is unaffected by the surface water head and is equal to the groundwater surface level. At intermediate depths a linear interpolation takes place, using the relative groundwater level fac:

$$\left[p_{gw}\right]_{i,j}^{n} = \left[\eta_{gw}\right]_{i,j}^{n-1} + \left(1 - fac_{i,j}^{n}\right)\left(\left[z_s\right]_{i,j}^{n-1} - \left[\eta_{gw}\right]_{i,j}^{n-1}\right) \qquad \text{if } wetz_{i,j}^{n-1} = 1$$

$$\tag{2.82}$$

$$fac_{i,j}^{n} = \frac{\left[z_b\right]_{i,j}^{n-1} - \left[\eta_{gw}\right]_{i,j}^{n-1}}{d_{wetlayer}} \qquad\qquad 0 \le fac \le 1$$

2.10.3 Momentum balance

Darcy flow is described by the following relationship between the groundwater head gradient, the permeability k, and the horizontal velocity:

$$u_{gw} = -k_x \frac{dp_{gw}}{dx}$$

$$v_{gw} = -k_y \frac{dp_{gw}}{dy}$$

$$\tag{2.83}$$

In the module, the head gradient is found numerically using:

$$\left[\frac{dp_{gw}}{dx}\right]_{i,j}^{n} = \frac{p_{i+1,j}^{n} - p_{i,j}^{n}}{x_{z,i+1,j} - x_{z,i,j}}$$

$$\left[\frac{dp_{gw}}{dy}\right]_{i,j}^{n} = \frac{p_{i,j+1}^{n} - p_{i,j}^{n}}{y_{z,i,j+1} - y_{z,i,j}}$$

$$\tag{2.84}$$

And horizontal flow is calculated by:

$$\left[u_{gw}\right]_{i,j}^{n} = -k_x \left[\frac{dp_{gw}}{dx}\right]_{i,j}^{n}$$

$$\left[v_{gw}\right]_{i,j}^{n} = -k_y \left[\frac{dp_{gw}}{dy}\right]_{i,j}^{n}$$

$$\tag{2.85}$$

2.10.4 Determining vertical flow

In order to simulate the interaction between the surface water and groundwater, a vertical flow between the surface water layer and groundwater layer (w) is introduced. This flow has the unit [ms^{-1}] and is defined positive from surface water to ground water and is given in terms of surface water for the continuity equation (i.e. 100% porosity).

Exfiltration

Exfiltration, or flow from the groundwater layer to the surface water layer, takes place if the groundwater surface level exceeds the bed level. The volume of groundwater (including porosity por) exceeding the bed level is joins the surface water within the same numerical time step. The vertical velocity can therefore be calculated by:

$$w_{i,j}^{n} = \left(\frac{\left[\eta_{gw} \right]_{i,j}^{n-1} - \left[z_{b} \right]_{i,j}^{n-1}}{\Delta t} \right) por \qquad \text{if} \quad \left[\eta_{gw} \right]_{i,j}^{n-1} \geq \left[z_{b} \right]_{i,j}^{n-1} \qquad (2.86)$$

Vertical infiltration model

Surface water running up and down a dry slope will infiltrate into the ground. In order to model this fully, a 3D model must be used. In the XBeach groundwater module, the option is made to model infiltration using a quasi-3D model.

In areas where there is surface water and the groundwater level is not greater than the bed level, infiltration can take place. To a certain degree of truth, infiltration can be calculated using Darcy flow:

$$w = -k_{z} \left(\frac{dp}{dz} + 1 \right) \qquad (2.87)$$

In an area that is covered by surface water, the head on the top of the bed can be said to be equal to the surface water head. In the absence of groundwater at the bed level, the head under the bed level is zero. As the distance between the top and bottom of the bed level is zero, the head gradient is infinite. The resulting vertical velocity becomes infinite and the method becomes numerically unstable. In order to circumvent this problem the vertical infiltration is divided into an instantaneous, but finite reaction in the upper ground layer and Darcy flow across a non-zero depth. The proportion of the instantaneous part to the Darcy flow part is governed by the relative groundwater level fac, as in section 2.10.2. The instantaneous part is handled in the same way as exfiltration. The head gradient for the Darcy flow is found by assuming the head at the bottom of the infiltration layer is zero, and the head on the top of the infiltration layer is equal to the height of water standing on the bed ($z_{s} - z_{b}$).

if $\left[wetz \right]_{i,j}^{n-1} = 1$ and $\left[\eta_{gw} \right]_{i,j}^{n-1} < \left[z_b \right]_{i,j}^{n-1}$:

$$w_{i,j}^n = \left((1 - fac) \frac{\left[\eta_{gw} \right]_{i,j}^{n-1} - \left[z_b \right]_{i,j}^{n-1}}{\Delta t} + fac_{i,j}^n \, k_z \left[\frac{z_s - z_b}{d_{infiltration}} \right]_{i,j}^{n-1} \right) por \qquad (2.88)$$

if $\left[wetz \right]_{i,j}^{n-1} = 0$ and $\left[\eta_{gw} \right]_{i,j}^{n-1} < \left[z_b \right]_{i,j}^{n-1}$:

$$w_{i,j}^n = 0$$

The infiltration velocity is limited by the amount of surface water available in the cell:

$$w_{i,j}^n \leq \frac{h_{i,j}^{n-1}}{\Delta t} \qquad (2.89)$$

The thickness of the infiltration layer ($d_{infiltration}$) is increased at the end of every time step by the infiltrating water. The infiltration speed in the next time step will therefore be less than that in the current time step. Infiltrating water is assumed to immediately become part of the groundwater for the purpose of groundwater level and groundwater head calculations. This approach is therefore not fully 3D and only uses a quasi-3D approximation to limit the infiltration speed.

$$\left[d_{infiltration} \right]_{i,j}^n = \left[d_{infiltration} \right]_{i,j}^{n-1} + \frac{w_{i,j}^n \Delta t}{por} \qquad (2.90)$$

For numerical stability, the infiltration layer thickness is restricted to a minimum of one third of $d_{wetlayer}$, corresponding with the centroid of the instantaneous infiltration part. The maximum thickness of the infiltration layer is equal to the depth of the groundwater level below the bed level. Once an area has no surface water, the thickness of the infiltration layer is reset to the minimum value, representing the fact that the infiltrated water has sunk out of the way of subsequent infiltrations.

$$\left[d_{infiltration} \right]_{i,j}^{n-1} = \frac{1}{3} d_{wetlayer} \qquad \qquad \text{if } wetz_{i,j}^{n-1} = 0$$

$$\frac{1}{3} d_{wetlayer} \leq \left[d_{infiltration} \right]_{i,j}^{n-1} \leq \left[z_b - \eta_{gw} \right]_{i,j}^{n-1} \qquad \text{if } wetz_{i,j}^{n-1} = 1 \qquad (2.91)$$

2.10.5 Mass balance

The continuity equation for the groundwater system can be written as:

$$\frac{d\eta_{gw}}{dt} + \frac{du_{gw}h_{ugw}}{dx} + \frac{dv_{gw}h_{vgw}}{dy} = \frac{w}{por} \tag{2.92}$$

The effective depths through which horizontal ground water flow takes place (h_{ugw}, h_{vgw}), are found by taking the mean difference between the groundwater level and bed of the aquifer ($z_{b,aquifer}$) in the two surrounding η -points:

$$\begin{aligned}
\left[h_{ugw} \right]_{i,j}^{n} &= \frac{\left[\eta_{gw} - z_{b,aquifer} \right]_{i,j}^{n} + \left[\eta_{gw} - z_{b,aquifer} \right]_{i+1,j}^{n}}{2} \\[2mm]
\left[h_{vgw} \right]_{i,j}^{n} &= \frac{\left[\eta_{gw} - z_{b,aquifer} \right]_{i,j}^{n} + \left[\eta_{gw} - z_{b,aquifer} \right]_{i,j+1}^{n}}{2}
\end{aligned} \tag{2.93}$$

This method is faster, but less momentum conservative than the method used in the surface water flow routine. Since large gradients in the groundwater level are not expected, the scheme is assumed sufficient.

Groundwater flux is limited in cells that are empty of groundwater. For such cells, groundwater may enter the cell, but no groundwater may leave until the amount of groundwater exceeds a minimum value (*eps*).

$$\left.\begin{aligned}
\left[u_{gw}h_{ugw} \right]_{i,j}^{n} &= \min\left(\left[u_{gw}h_{ugw} \right]_{i,j}^{n}, 0 \right) \\[1mm]
\left[u_{gw}h_{ugw} \right]_{i-1,j}^{n} &= \max\left(\left[u_{gw}h_{ugw} \right]_{i-1,j}^{n}, 0 \right) \\[1mm]
\left[v_{gw}h_{vgw} \right]_{i,j}^{n} &= \min\left(\left[v_{gw}h_{vgw} \right]_{i,j}^{n}, 0 \right) \\[1mm]
\left[v_{gw}h_{vgw} \right]_{i,j-1}^{n} &= \max\left(\left[v_{gw}h_{vgw} \right]_{i,j-1}^{n}, 0 \right)
\end{aligned}\right\} \quad \text{if} \quad \left[\eta_{gw} \right]_{i,j}^{n-1} \leq \left[z_{b,aquifer} \right]_{i,j}^{n-1} + eps \tag{2.94}$$

The continuity equation for the groundwater level is solved by the following:

$$\frac{\left[\eta_{gw} \right]_{i,j}^{n+1} - \left[\eta_{gw} \right]_{i,j}^{n}}{\Delta t} = -\frac{\left[u_{gw}h_{ugw} \right]_{i,j}^{n} - \left[u_{gw}h_{ugw} \right]_{i-1,j}^{n}}{x_{u,i,j} - x_{u,i-1,j}} - \frac{\left[v_{gw}h_{vgw} \right]_{i,j}^{n} - \left[v_{gw}h_{vgw} \right]_{i,j-1}^{n}}{y_{v,i,j} - y_{v,i,j-1}} + \frac{w_{i,j}^{n}}{por} \tag{2.95}$$

To account for infiltrating and exfiltrating groundwater, an additional term is added to the continuity equation of the surface water, but none to the momentum balance.

$$\frac{\eta_{i,j}^{n+1} - \eta_{i,j}^{n}}{\Delta t} = -\frac{u_{i,j}^{n+1}h_{i,j}^{n} - u_{i-1,j}^{n+1}h_{i-1,j}^{n}}{x_{u,i,j} - x_{u,i-1,j}} - \frac{v_{i,j}^{n+1}h_{i,j} - v_{i,j-1}^{n+1}h_{i,j-1}^{n}}{y_{v,i,j} - y_{v,i,j-1}} - w_{i,j}^{n} \tag{2.96}$$

2.10.6 Groundwater boundary conditions

Vertical boundary conditions

The groundwater level is bounded by the bottom of the aquifer. In the central domain the groundwater level is adjusted naturally by infiltration and exfiltration. The groundwater level has no bounding maximum in the vertical, except on the offshore, bay side and lateral boundaries. Here the groundwater level is bounded vertically by the bed level on the boundaries:

$$
\begin{aligned}
\left[\eta_{gw}\right]_{1,j}^{n} &= \min\left(\left[\eta_{gw}\right]_{1,j}^{n}, \left[z_{b}\right]_{1,j}^{n}\right) \\
\left[\eta_{gw}\right]_{nx+1,j}^{n} &= \min\left(\left[\eta_{gw}\right]_{nx+1,j}^{n}, \left[z_{b}\right]_{nx+1,j}^{n}\right) \\
\left[\eta_{gw}\right]_{i,1}^{n} &= \min\left(\left[\eta_{gw}\right]_{i,1}^{n}, \left[z_{b}\right]_{i,1}^{n}\right) \\
\left[\eta_{gw}\right]_{i,ny+1}^{n} &= \min\left(\left[\eta_{gw}\right]_{i,ny+1}^{n}, \left[z_{b}\right]_{i,ny+1}^{n}\right)
\end{aligned}
\tag{2.97}
$$

The bed of the aquifer is set equal to or less than the regular bed level:

$$
z_{b,aquifer} = \min\left(z_{b,aquifer}, z_{b} - eps\right)
\tag{2.98}
$$

Offshore boundary condition

At the offshore boundary, the groundwater head is set equal to the offshore surface water head:

$$
\left[p_{gw}\right]_{i,1}^{n} = \left[z_{s}\right]_{i,1}^{n}
\tag{2.99}
$$

Bay side boundary condition

For cases in which a bay side water level is given explicitly with a tidal level record (*tideloc = 4, tideloc = 2* and *paulrevere = 0*), the groundwater head on the bay side boundary is set equal to the bay side surface water head:

$$
\left[p_{gw}\right]_{i,ny+1}^{n} = \left[z_{s}\right]_{i,ny+1}^{n}
\tag{2.100}
$$

In all other cases, the bay side groundwater head is kept at the initial value:

$$\left[p_{gw} \right]_{i,ny+1}^{n} = \left[p_{gw} \right]_{i,ny+1}^{1} \tag{2.101}$$

Lateral boundary conditions

Neumann boundary conditions are applied to the groundwater head on the lateral boundaries:

$$\left[p_{gw} \right]_{i,1}^{n} = \left[p_{gw} \right]_{i,2}^{n}$$
$$\left[p_{gw} \right]_{i,ny+1}^{n} = \left[p_{gw} \right]_{i,ny}^{n} \tag{2.102}$$

Initial conditions

The bed of the aquifer and the initial groundwater head must be specified, see section 5.12 for a description. The initial groundwater level is calculated from the initial groundwater head.

2.11 Drifters (*advanced option*)

Drifters are objects that move with the lagrangean mean velocity. They are defined by their deployment location, deployment time and retrieval time. The positions of the drifters are evaluated every timestep; every tintp seconds the results are written to files drifternnn.dat, with nnn the drifter number padded with zeros. The files are in the usual binary format, with each record containing x,y and time of the drifter; invalid timepoints are characterized by -999 values.

The drifter module is activated by including the following statements in params.txt.

ndrifter=*number of drifters*
drifterfile=*name of drifterfile*

The drifter input file should contain for each drifter a record with 4 numbers indicating x and y (in world coordinates) of release location, release time and retrieval time (in seconds from start of simulation)

Example input in params.txt
ndrifter=5
drifterfile=drifters.txt

Content of drifters.txt
550. 100. 100. 1000.
550. 200. 100. 1000.
550. 250. 100. 1000.
550. 300. 100. 1000.
550. 400. 100. 1000.

Example function to read and plot drifter paths

```
function read_drifter()
ch={'r.';'b.';'k.';'g.';'m.';'c.'}
for id=1:6
    dep=mod((id-1),6)+1;
    if id<10
        fname=['drifter00' num2str(id) '.dat']
    else
        fname=['drifter0' num2str(id) '.dat']
    end
    try
    fidr=fopen(fname,'r')
    for i=1:10000
        xyt=fread(fidr,[3],'double');
        if isempty(xyt)
            break
        end
        xyt(xyt==-999)=nan;
        xd(i)=xyt(1);yd(i)=xyt(2);td(i)=xyt(3);
    end
    fclose(fidr)
    plot(xd,yd,ch{dep});axis equal
    clear xd yd td
    catch
    end
end
```
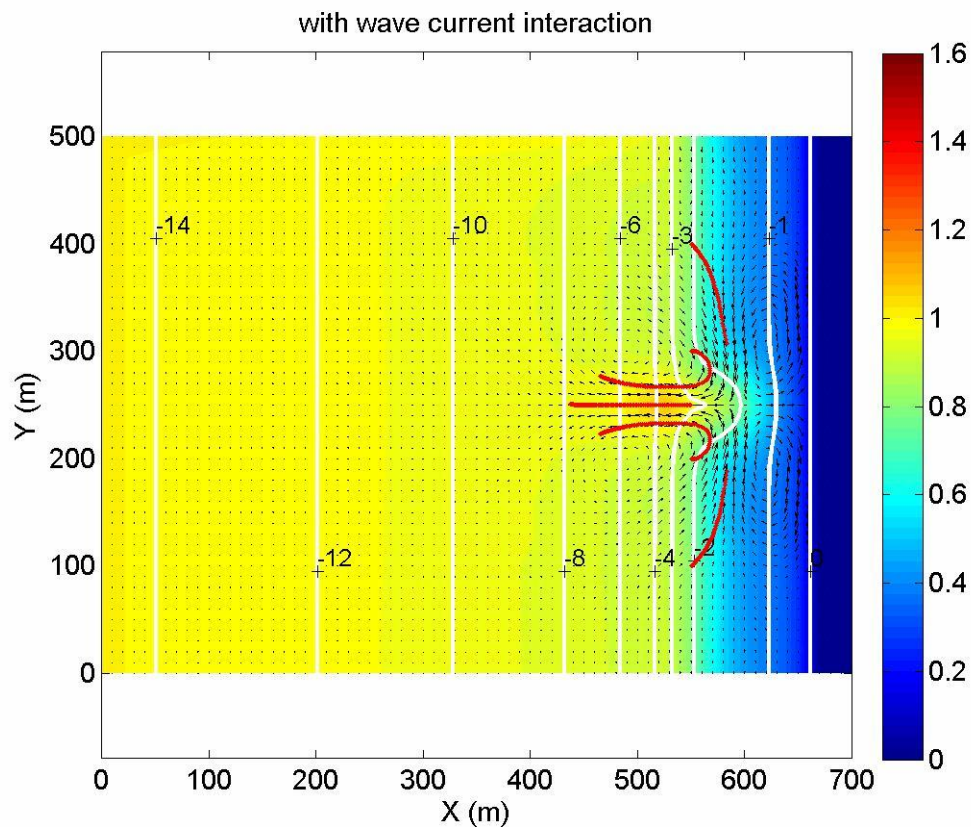
Figure 2.8   Example of drifter path output.

2.12 Discharge boundaries (river input) (*advanced option*)

It is possible to define river discharges to simulate the inflow from a river (i.e. uni-directional, no tides) and no sediment transport.

The user can define a number of sections along the boundaries of the grid through which discharges can be defined as time series. For each boundary section, the program checks how many grid cell centers lie within the section, given in world coordinates. The total length of the section is computed as the sum of the lengths of the grid cells along the boundary, and each cell length divided by this total length is used to divide the discharge over the cells within the boundary section.

For each boundary section a separate time series of total discharge is defined; actual discharges are determined by interpolation in time. Positive discharges mean discharges into the model.

The discharges may be defined on initially dry boundaries. They must, however be located on otherwise closed ("wall") boundaries, see parameter ("back"_

The discharge boundaries are activated with the following keywords in *params.txt*:

disch_loc_file=<*discharge locations file*>
disch_timeseries_file=<*discharge timeseries file*>

The contents of the discharge locations file are a number of records equal to the number of discharge boundary sections, with for each record four numbers:

X_begin Y_begin X_end Y_end

These world coordinates must be chosen such that they are close to the desired boundary and enclose the cell centers of the cells that must be part of the boundary section.

In the discharge time series file, the first column is time in seconds; the next columns give the total discharge time series per section.

Example: (River Outflow)

disch_loc_file=discharge_locations.txt
disch_timeseries_file=discharge_timeseries.txt

Contents of discharge_locations.txt:
840 820 840 890

Contents of discharge_timeseries.txt:
0 150
1000000 150
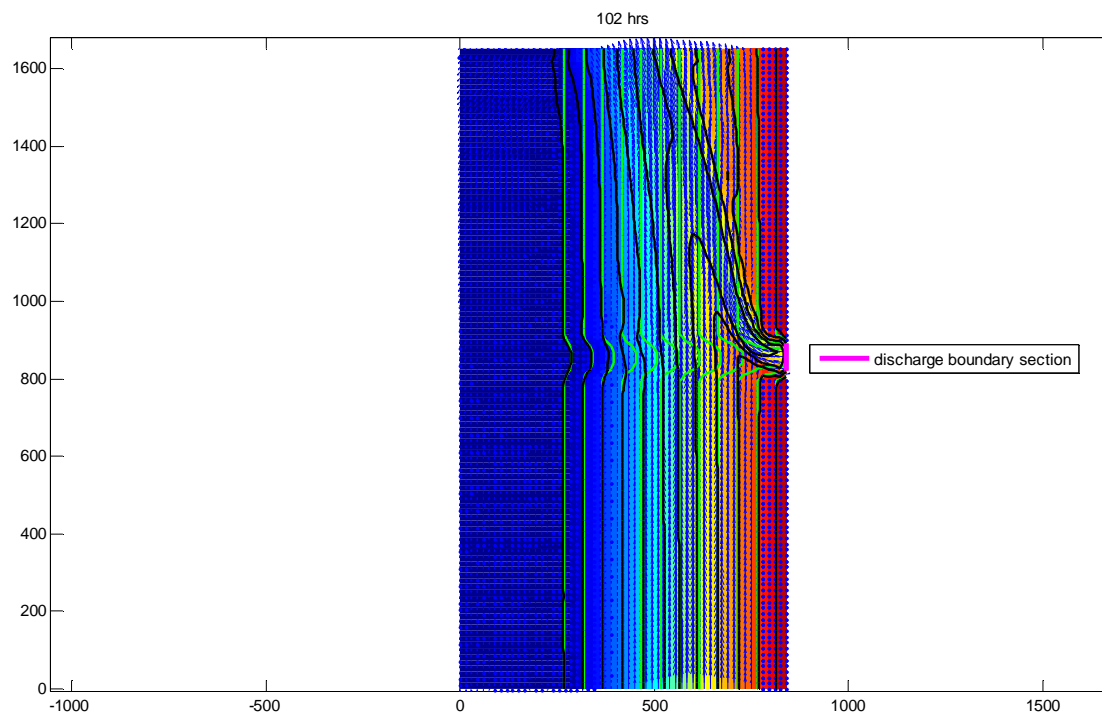
Figure 2.9   Example of coast with river inflow.

2.13 Dam break (*advanced option*)

Dam break problems require the same set of hydrodynamic equations which XBeach contains. The only difference between these types of problems and convential coastal model is the specification of the innitial condition file for the water level.

An non-uniform initial condition for the water level, for instance when modeling a dam break problem, can be specified by a file with the same format as the depth file. It is activated by the following keyword in *params.txt*:
zsinitfile = <initial condition file>

Example (Dambreak)

Zsinitfile=ini.dep
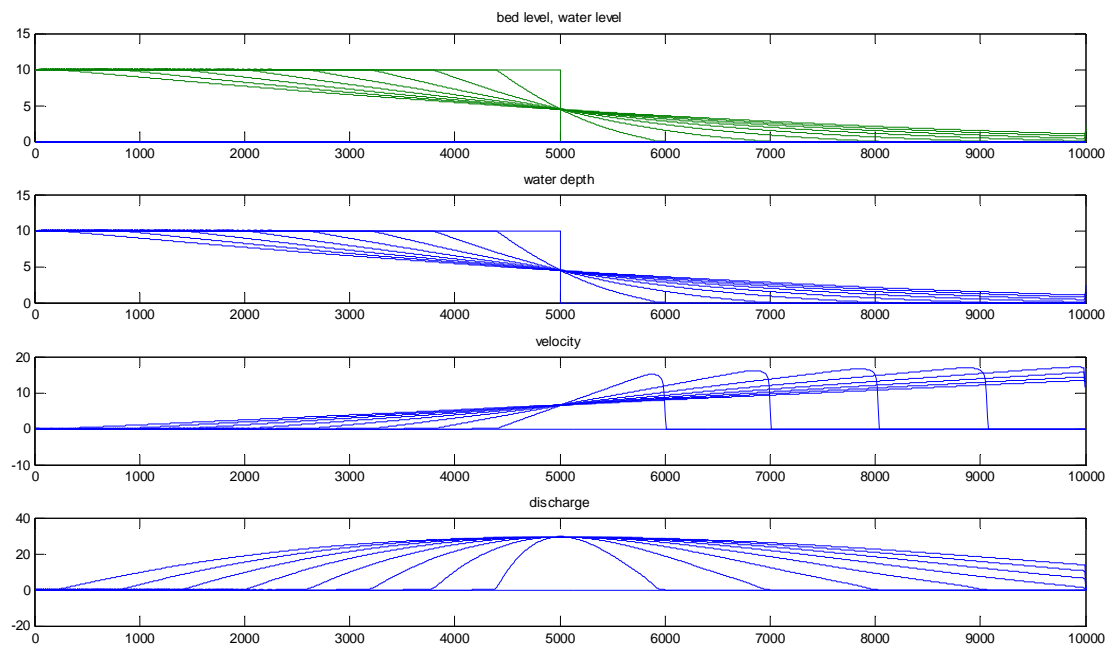
Figure 2.10   Dam break problem: snapshots of the water level, water depth, velocity and discharge.

3   Descripton of program structure

3.1   Single domain setup

The program XBeach consists of a main *Fortran 90* program, xbeach.f90, and a number of *subroutines* that operate on two *derived types (structures)*:

- par – this contains general input parameters
- s   – this contains all the arrays for a given computational domain

For a single-domain run, one structure s is passed between flow, wave, sediment and bed update solvers, which extract the arrays they need from the structure elements to local variables, do their thing and pass the results back to the relevant structure elements. This makes the overall program clear, prevents long parameter lists and makes it easy to add input variables or arrays where needed. The various subroutines and their purposes are outlined in Table 3.1.

Table 3.1.   Overview of Fortran 90 subroutine calls by xbeach.f90

Function call	Purpose
wave_input(par)	Creates elements of structure par containing wave input parameters
flow_input(par)	Adds elements of structure par containing flow input parameters
sed_input(par)	Adds elements of structure par containing sediment input parameters
grid_bathy(s)	Creates grid and bathymetry and stores them in structure s
distribute_par(par)	MPI
space_alloc_scalars(sglobal)	allocates space
grid_bathy(s,par)	sets up grid and bathymetry
xmpi_determine_processor_grid(s%nx	s%ny)
readtide(s,par)	Read tide levels
readwind (s, par)	Read wind field
init_output	Read output requests and initialize output files
wave_init (s,par)	Initialises arrays (elements of s) for wave computations
flow_init (s,par)	Initialises arrays (elements of s) for flow computations
gwinit(s,par)	Initialises arrays (elements of s) for groundwater module
sed_init (s,par)	Initialises arrays (elements of s) for sediment computations
init_output(sglobal,slocal,par,it)	
call readkey	

Start time loop	
timestep (s,par,it)	Calculate automatic timestep
wave_bc (s,par)	Wave boundary conditions update, each timestep
gwbc(s,par)	Groundwater boundary conditions update, each timestep
flow_bc (s,par)	Flow boundary conditions update, each timestep
wave_timestep(s,par)	Carries out one wave timestep, OR
wave_stationary(s,par)	Carries out stationary wave computation
gwflow(par,s)	carries out flow timestep of groundwater flow
flow_timestep (s,par)	Carries out one flow timestep
drifter (s,par)	
transus(s,par)	Carries out one suspended plus bedload transport timestep
bed_update(s,par)	Carries out one bed level update timestep
var_output(it,s,par)	Performs output
End time loop	

3.2   Implementation of parallel computing using MPI

We refer to the "Parallelization Report".

4   Compiling the model

4 – 1

The source code has been extensively tested in a version compiled under Compaq Visual Fortran version 6.6 on the Windows platform and under Linux under GFortran.

We expect it to compile with only minor problems under other compilers and/or under Linux, since only standard Fortran 90/95 is used. Once an executable has been created, it will be called xbeach.exe.

5   Running the model

5.1   Input file structure

On execution the *xbeach.exe* executable will read the file *params;txt* in the current directory and interpret the keyword=value combinations in it. The keywords refer to parameter values or filenames and can be listed in any order. Lines not containing an '=' sign are ignored and may be used for comments.

The 'params.txt' file contains grid and bathymetry info, wave input, flow input and morphological input. The tables below contain a description of the keywords, the default values and recommended minimum and maximum values.

5.2   Physical processes

Physical processes can be turned on or off by choice. The commonly used ones are turned on by default, so for common cases the user does not need to specify anything explicitly

keyword	description	default value	minimum value	maximum value	unit	remarks
swave	include/exclude shortwaves	1	0	1	-	turned on by default
lwave	include/exclude generation of long waves	1	0	1	-	turned on by default
flow	include or exclude NLSW equations	1	0	1	-	turned on by default
sedtrans	include/exclude sediment transport	1	0	1	-	turned on by default
morphology	include/exclude bed updating	1	0	1	-	turned on by default
nonh	turn on nonhydrostatic flow	0	0	1	-	turned **off** by default
gwflow	turn on groundwater flow	0	0	1	-	turned **off** by default
q3d	turn on quasi 3d	0	0	1	-	turned **off** by default

5.3   Specifying grid and depth

keyword	description	default value	minimum value	maximum value	unit	remarks
nx	number of grid points in x	50	2	10000	-	
ny	number of grid points in y	2	0	10000	-	
dx	grid size in x	-1	0	1e9	m	required value
dy	grid size in y	-1	0	1e9	m	required value
xori	x - origin in world coordinates	0	-1e9	1e9	m	
yori	y – origin in world coordinates	0	-1e9	1e9	m	
alfa	angle of grid	0	0	360	deg	
posdwn	depth defined positive down	1	-1	1		1 = positive down -1 = positive up
vardx	option of variable grid size	0	0	1	-	1 = varying grid size 0 = non-varying grid size
depfile	bathymetry file					no values; specify name
xfile	variable gridsize file x					no values; specify name;only with vardx=1
yfile	variable gridsize file y					no values; specify name only with vardx=1
thetamin	lower directional limit	-90	-180	180	deg	angle w.r.t computational x-axis
thetamax	upper directional limit	90	-180	180	deg	angle w.r.t computational x-axis
dtheta	directional resolution	10	0.1	20	deg	
thetanaut	option to enter thetamin thetamax in nautical convention	0	0	1	-	0 = cartesian, 1 = nautical

The *depfile* keyword contains the reference to a bathymetry file, which should contain, for each of ny+1 rows, nx+1 depth values, which may be defined positive downward or upward, depending on a keyword *posdwn* that may be 1 (depth positive downward) or -1 (positive upward).

If a non-equidistant grid is chosen, the keyword *vardx=1* should be selected and an xfile and yfile have to be specified.

The directional grid for short waves and rollers can either be specified as Cartesian (angle w.r.t. the computational x-axis) or as nautical directions (direction waves come from in deg. N, so from W is 270 deg. N); this depends on *thetanaut (*1 means nautical, default Cartesian is 0) .

5.4   Physical constants

keyword	description	default value	minimum value	maximum value	unit	remarks
rho	density of water	1025	1000	140	kg/m3	
g	gravitational acceleration	9.81	9.7	9.9	m/s2	

5.5   Time management

The hydrodynamic simulation starts at time 0 and starts outputting at time *tstart*; it stops at *tstop*. The time intervals are described in detail in section 1.1. The actual time step of the hydrodynamic simulation is determined based on a given maximum Courant number *CFL*. In future versions a nonhydrostatic pressure correction term will be implemented in the flow solver, allowing the modelling of individual short waves. This is not available in the present code yet.

keyword	description	default value	minimum value	maximum value	unit	remarks
tstart	start time of simulation	1	0	1000000	s	
tint	time interval output global values	1	0.01	100000	s	

tintg	time interval output global values	tint	0.01	100000		supercedes tint if specified
tintm	time interval output mean	ting	0.01	tstop		

keyword	description					
	global values					
tintp	time interval output point values	ting	0.01	100000		
tsglobal	file with list of output times for global values					no values; specify name
tsmean	file with list of output times for meanglobal values					no values; specify name
tspoints	file with list of output times for point values					no values; specify name
tscross	file with list of output times at cross sections					<mark>NOT IMPLEMENTED YET</mark>
tstop	stop time simulation	2000	1	1000000	s	
CFL	maximum courant number	0.7	0.1	0.9	-	actual CFL number varies during calculation

5.6   Wave input

5.6.1   Action balance

With the switch *wci* one can turn off or on the wave-current interaction, viz. the feedback of currents on the wave propagation, see section 2.3. With "scheme" the numerical scheme can be chosen, by default use the higher-order Lax Wendroff scheme to minimize numerical dissipation.

keyword	description	default value	minimum value	maximum value	unit	remarks
wavint	interval between stationary wave module calls	1	1	3600	s	*instat* = 0 only
scheme	Switch numerical schemes for wave action balance	2	1	2	-	1 = Upwind 2 = Lax Wendroff
wci	wave current interaction option	0	0	1	-	*instat* = 0 only
hwci	Minimum depth above which wave-current interaction is used	0.1	0.0001	1	m	

| cats | Current averaging time scale for wci, in terms of mean wave periods | 20 | 1 | 50 | Trep | |

5.6.2 Wave dissipation model

For instationary model runs, use the Roelvink (1993) model with either *break=1* or *break=3*. Note that the standard value *gamma=0.55* and *n=10* was calibrated for option *break=1*. for *break=3* the wave dissipation is proportional to H^3/h instead of H^2; this affects the calibration. For stationary runs the Baldock et al, 1998 the 'break=2 option', is suitable. The 'break=4' option is a model in which waves start and stop breaking

keyword	description	default value	minimum value	maximum value	unit	remarks
break	option breaker model	3	1	3	-	1 = 'roelvink1' 2 = 'baldock' 3 = 'roelvink2' 4 = 'roelvink_daly'
gamma	breaker parameter in Baldock or Roelvink formulation	0.55	0.4	0.9	-	
gamma2	end of breaking	0.3	0.0	0.5		break 4 only
gammax	maximum allowed waveheight over waterdepth	2	0.4	5		this cuts off the waveheight numerically.
alpha	wave dissipation coefficient	1	0.5	2	-	
n	power in roelvink dissipation model	10	5	20	-	
delta	Fraction of wave height to add to water depth	0	0	1	-	
fw	Bed friction factor	0	0	1	-	on wave action equation only! unrelated to bed friction in flow equation
breakerdelay		1	0	1	-	

5.6.3 Roller model

Using the roller model will give a shoreward shift in wave-induced setup, return flow and longshore current. This shift becomes greater for lower *beta* values.

keyword	description	default	minimum	maximum	unit	remarks

		value	value	value		
roller	option roller model	1	0	1	-	keyword not implemented yet. Roller model is turned on by default
beta	breaker slope coefficient in roller model	0.15	0.05	0.3	-	

5.7   Wave boundary conditions

5.7.1   User input

The user input options include various options for wave boundary conditions in *params.txt*. The meaning of each is summarised below:

instat	abbreviated name	description
0	stat	stationary wave boundary condition (sea state)
1	bichrom	bichromatic (two wave component) waves
2	ts_1	first-order timeseries of waves (generated outside XBeach)
3	ts_2	second-order timeseries of waves (generated outside XBeach)
4	jons	wave groups generated using a parametric (Jonswap) spectrum
5	swan	wave groups generated using a SWAN 2D output file
6	vardens	wave groups generated using a formatted file
7	reuse	reuse of wave conditions
8	nonh	boundary conditions for nonhydrostatic option
9	off	no wave boundary condition
40	stat_table	a sequence of stationary conditions (sea states)
41	jons_table	a sequence of time-varying wave groups

.

The steps that should be taken for an XBeach simulation using the new wave boundary conditions are explained in the following sections. The reader is advised to consult Figure 5.1 to determine which sections are relevant to the simulation.
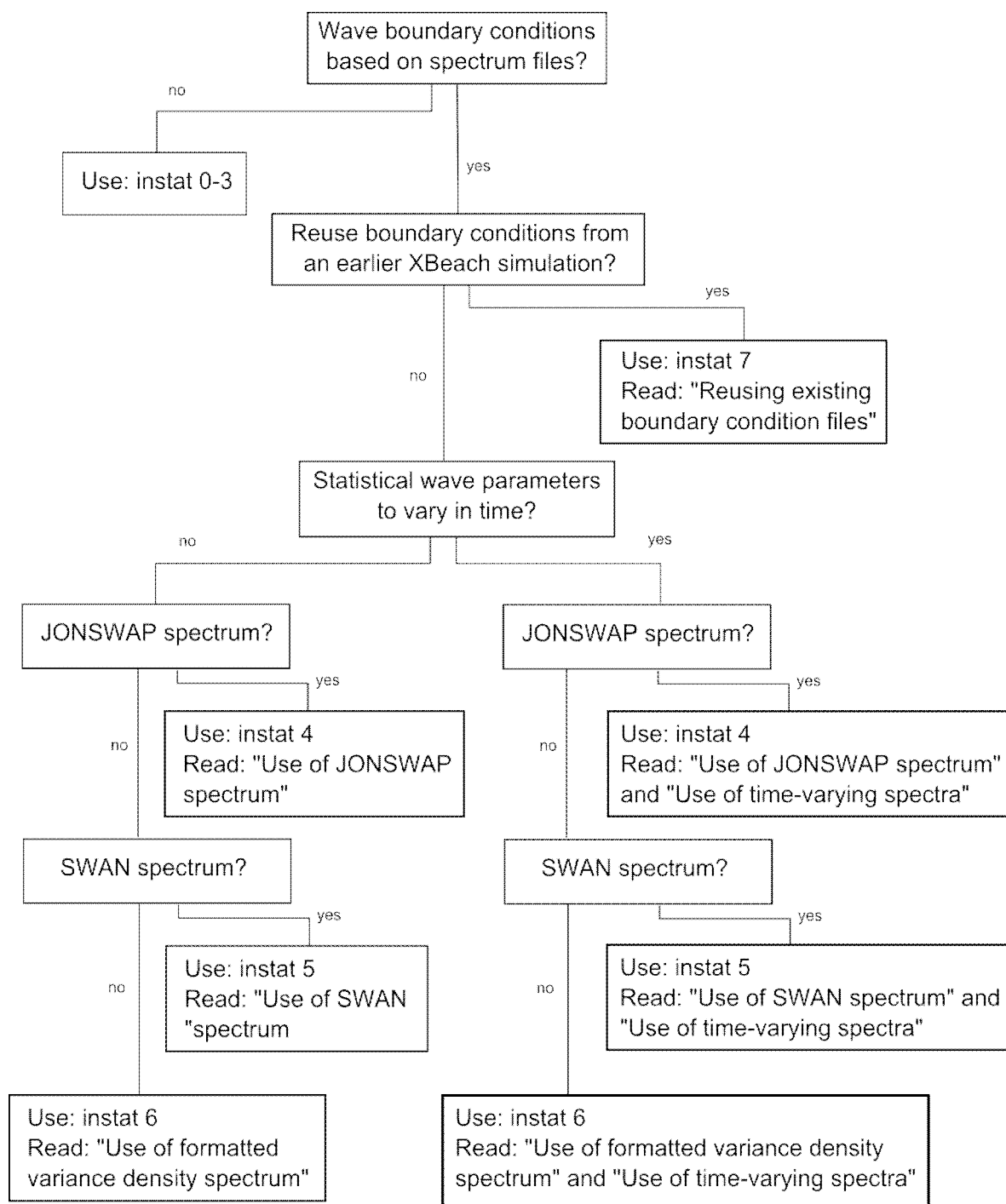
Figure 5.1   Reading guide

5.7.2 Instat = 0 to 3. wave boundary condition parameters for non-spectral input

keyword	description	default value	minimum value	maximum value	unit	remarks
dir0	mean wave direction	270	180	360	deg	instat=0-3

	(Nautical convention)					only
Hrms	rms wave height	1	0	10	m	instat=0-3 only
m	power in cos^m directional distribution	10	2	128	-	instat=0-3 only
Tm01	spectral period	10	1	20	s	instat=0-3 only
Trep	alternative keyword for representative period	10	1	20	s	instat=0-3 only; overrules value for Tm01
Tlong	long wave / wave group period	80	20	300	s	*instat = 1* only
taper	time to spin up wave boundary conditions	100	1e-6	1000	s	not for stationary waves

5.7.3 Instat = 4, spectral parameter input

The new XBeach module allows the user to provide JONSWAP parameters from which XBeach computes a spectrum. To make use of this option, the user must specify *'instat = 4'* in *params.txt*. XBeach will then attempt to read JONSWAP parameters from a separate file specified by *'bcfile ='* in *params.txt*. The user must also state in *params.txt* the required record length for the boundary condition file and the boundary condition file time step (keywords *'rt = '* and *'dtbc = '* respectively). If the record length (*rt*) is less than the total simulation time, XBeach will reuse the boundary condition file until the simulation is completed. The boundary condition file time step should be small enough to accurately represent the bound long wave, but need not be as small as the time step used in XBeach, see *Explanation of input/output*.

keyword	description	default value	minimum value	maximum value	unit	remarks
bcfile	input file for spectral computations	-	-	-		file, with contents specified below
rt	record length	3600	1200	7200	s	
dtbc	file time step	0.5	0.1	2	s	
fcutoff	low freq cutoff frequency for boundary conditions	0	0	40	Hz	for instat=4,5,6
sprdthr	threshold variance density above which spec dens are read in	0.08	0	1	-	this is to reduce the spectral width
nspr	set directional spreading long waves	0	0	1	-	1 = bin all incoming long wave directions (instat 4+) in the centres of the short wave directional grid cells
trepfac	Compute mean wave period over energy band:	0.8	0	1	-	par%trepfac*maxval(Sf) for instat 4,5,6; converges to Tm01 for trepfac = 0.0
random	random generator	1	0	1		Random seed on (1) or off (0)
oldwbc	old waveboundary condition	0	0	1	-	this value should stay at oldwbc=0.

The contents of the file specified by '*bcfile =*' in *params.txt* is a list of *keyword = value* combinations which determine the JONSWAP spectrum. These keywords are:

Keyword	Type	Description	Default	Minimum	Maximum
'Hm0 = '	real	H_{m0} of the wave spectrum, significant wave height [m]	0.0	0.0	5.0
'fp = '	real	Peak frequency of the wave spectrum [s^{-1}]	0.08	0.0625	0.4
'gammajsp = '	real	Peak enhancement factor in the JONSWAP expression [-]	3.3	1.0	5.0
's = '	real	Directional spreading coefficient, cosine law [-]	10.	1.0	1000.
'mainang = '	real	Main wave angle (in nautical terms) [°]	270.	180.	360.
'fnyq = '	real	Highest frequency used to create JONSWAP spectrum [s^{-1}]	0.3	0.2	1.0
'dfj = '	real	Step size frequency used to create JONSWAP spectrum [s^{-1}]	fnyq/200	fnyq/1000	fnyq/20

All variables are optional. If no value is given, the default value is used. It is advised not to specify *dfj* and allow XBeach to calculate the default value.

A typical input file contains the following:

```
Hm0       = 0.8
fp        = 0.125
mainang   = 285.
gammajsp  = 3.3
s         = 10.
fnyq      = 0.3
```

5.7.4 Instat = 5, SWAN spectrum input

XBeach has been programmed to read standard SWAN 2D variance density or energy density output files (.sp2 files), as specified in the SWAN v40.51 manual.

To make use of this option, the user must specify '*instat = 5*' in *params.txt*. XBeach will then read the SWAN output spectrum from a separate file specified by '*bcfile =*' in *params.txt*. The user must also state in *params.txt* the required record length for the boundary condition file and the boundary condition file time step (keywords '*rt =* ' and '*dtbc =* ' respectively). If the record length (*rt*) is less than the total simulation time, XBeach will reuse the boundary condition file until the simulation is completed. The boundary condition file time step should be small enough to accurately represent the bound long wave, but need not be as small as the time step used in XBeach, see *Explanation of input/output.* For definitions, see above.

XBeach assumes the output of the SWAN file is in nautical terms. If the file is in Cartesian angles, the user must specify the angle in degrees to rotate the x-axis in SWAN to the x-axis in XBeach (in Cartesian terms). This value is specified in *params.txt* after the keyword '*dthetaS_XB =*'.

keyword	description	default value	minimum value	maximum value	unit	remarks
dthetaS_XB	conversion of angles between SWAN and XBeach	0	-360	360	deg	If SWAN input is not in nautical degrees, dthetaS_XB is the angle from SWAN x-axis to XBeach x-axis in cathesian degrees

An example of a SWAN 2D output file is given below:

```
SWAN   1                              Swan standard spectral file, version
$ Data produced by SWAN version 40.51
$ Project:'projname'    ;   run number:'runnum'
LOCATIONS                             locations in x-y-space
   1                                  number of locations
   22222.22         0.00
RFREQ                                 relative frequencies in Hz
   23                                 number of frequencies
   0.0545
   0.0622
   0.0710
   0.0810
   0.0924
   0.1055
   0.1204
   0.1375
   0.1569
   0.1791
   0.2045
   0.2334
   0.2664
   0.3040
   0.3470
   0.3961
   0.4522
   0.5161
   0.5891
   0.6724
   0.7675
   0.8761
   1.0000
CDIR                                  spectral Cartesian directions in degr
   12                                 number of directions
   30.0000
   60.0000
   90.0000
  120.0000
  150.0000
  180.0000
  210.0000
  240.0000
  270.0000
  300.0000
  330.0000
  360.0000
QUANT
   1                                  number of quantities in table
VaDens                                variance densities in m2/Hz/degr
m2/Hz/degr                            unit
   -0.9900E+02                        exception value
FACTOR
   0.675611E-06
     51     242     574     956    1288    1482    1481    1286     957     579     244      51
    129     610    1443    2402    3238    3725    3724    3234    2406    1454     613     128
    273    1287    3054    5084    6846    7872    7869    6837    5091    3076    1295     271
    665    3152    7463   12402   16712   19229   19221   16690   12419    7518    3172     662
   1302    6159   14608   24275   32688   37618   37603   32644   24309   14716    6198    1296
   2328   10989   26020   43341   58358   67109   67080   58281   43401   26213   11058    2317
   3365   15922   37712   62733   84492   97150   97110   84380   62820   37991   16021    3349
   3426   16230   38440   63939   86109   99010   98969   85995   64027   38724   16331    3410
   2027    9612   22730   37790   50909   58529   58505   50841   37843   22898    9672    2018
    672    3178    7538   12535   16892   19440   19432   16870   12552    7594    3198     669
    101     479    1135    1890    2542    2924    2923    2539    1892    1144     482     101
      2      11      26      43      57      66      66      57      43      26      11       2
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0
```

5.7.5 Instat = 6, Use of formatted variance density spectrum

If the user has 2D spectrum information, but not in SWAN or JONSWAP form, the user can create a formatted spectrum file which can be read by XBeach.

To make use of this option, the user must specify '*instat = 6*' in *params.txt*. XBeach will then attempt to read the formatted output spectrum from a separate file specified by '*bcfile =*' in *params.txt*. The user must also state in *params.txt* the required record length for the boundary condition file and the boundary condition file time step (keywords '*rt = *' and '*dtbc = *' respectively). If the record length (*rt*) is less than the total simulation time, XBeach will reuse the boundary condition file until the simulation is completed. The boundary condition file time step should be small enough to accurately represent the bound long wave, but need not be as small as the time step used in XBeach, see *Explanation of input/output*.

The contents of the file specified by '*bcfile =*' in *params.txt* must follow a specified format. The information should be as follows:

- First line: number of frequencies (integer)
- Second line onwards: column with frequencies in Hz (each on a new line)
- Then: number of angles (integer)
- Then onwards: column with angles in degrees (each on a new line)
- Then per line: row of variance density in each direction, per frequency (i.e. first row corresponds to variance density per direction in the first frequency band, second row of the second frequency band, etc.).

Note that the angles in the input file must be in the calculation coordinate system of XBeach, i.e. 0° is in the direction of the x-axis, 90° is in the direction of the y-axis. Also, the angles must be increasing.

An example of a formatted variance density file is given below:

```
    15
    0.0418
    0.0477
    0.0545
    0.0622
    0.0710
    0.0810
    0.0924
    0.1055
    0.1204
    0.1375
    0.1569
    0.1791
    0.2045
    0.2334
    0.2664
    13
 -180.0000
 -150.0000
 -120.0000
  -90.0000
  -60.0000
  -30.0000
    0.0000
   30.0000
   60.0000
   90.0000
  120.0000
  150.0000
  180.0000
     0      0      0      0      0      0      0      0      0      0      0      0
    51    242    574    956   1288   1482   1481   1286    957    579    244     51
   129    610   1443   2402   3238   3725   3724   3234   2406   1454    613    128
   273   1287   3054   5084   6846   7872   7869   6837   5091   3076   1295    271
   665   3152   7463  12402  16712  19229  19221  16690  12419   7518   3172    662
  1302   6159  14608  24275  32688  37618  37603  32644  24309  14716   6198   1296
  2328  10989  26020  43341  58358  67109  67080  58281  43401  26213  11058   2317
  3365  15922  37712  62733  84492  97150  97110  84380  62820  37991  16021   3349
  3426  16230  38440  63939  86109  99010  98969  85995  64027  38724  16331   3410
  2027   9612  22730  37790  50909  58529  58505  50841  37843  22898   9672   2018
   672   3178   7538  12535  16892  19440  19432  16870  12552   7594   3198    669
   101    479   1135   1890   2542   2924   2923   2539   1892   1144    482    101
     2     11     26     43     57     66     66     57     43     26     11      2
     0      0      0      1      1      1      1      1      1      0      0      0
     0      0      0      0      0      0      0      0      0      0      0      0
     0      0      0      0      0      0      0      0      0      0      0      0
```

5.7.6 Instat =7, reusing existing boundary condition files

If the user does not wish to recalculate boundary condition files or specifically wants to reuse the boundary condition files of another XBeach simulation, it is possible to do so. In this case the user should select '*instat = 7*' in *params.txt*. No further wave boundary condition data need be given in *params.txt*. Obviously, the calculation grid should remain the same between runs, as the angles and number of grid points are embedded in the boundary condition files.

In order to use instat 7, the user should copy *ebcflist.bcf* and *qbcflist.bcf* to the current directory. Additionally, the user should also copy all files listed in *ebcflist.bcf* and *qbcflist.bcf*. Generally, these files have *E_* and *q_* prefixes.

5.7.7 Instat = 8, boundary conditions for non-hydrostatic model

XBeach can be ran as a non-hydrostatic model, which is essentially the nonlinear shallow water equations with dispersion terms and without a wave-action driver. The boundary conditions are described in a separate chapter and are activated using instat=8

5.7.8 Instat = 9, no boundary condition

This is a simple "no wave action" boundary condition. It still allows for a tidal record to be specified, however through the zs0file parameter, see Chapter 5.9 below.

5.7.9 Instat = 40, sequence of stationary sea states

While instat=0 specifies one stationary sea state (stationary in the sense that there are no wave groups), it is also possible to specify a series of seastates, each with a duration. This is done through a file as

instat   = 40
bcfile   = jonswap1.txt

where the name of the bcfile is free, but the structure of the contents is not. It should contain lines with
Hm0, Tp, angle, gamma (3.3), spreading, duration (s), timestep (=0.05)

Ie
0.7 8 90 3.3 5 1000 0.05

0.8 7 110 3.3 5 1000 0.05

5.7.10 Instat = 41, sequence of sea states to make time-varying wave groups

This is an extension of instat=4. With instat = 41 it is possible to specify a sea state on the basis of which wave groups are imposed on the model for a certain duration, then specify another sea state and run wave groups again without having to stop the model.

This condition is specified as
instat   = 41
bcfile   = jonswap1.txt

where the name of the file is free and the structure of the contents is as in instat = 40.

5.7.11 Notes on generation of boundary conditions

At the start of the XBeach simulation, XBeach checks whether non-stationary varying wave boundary conditions are to be used. If this is the case, it next checks whether the *wave spectrum* of the wave boundary conditions is to change over time, or remain constant. If the wave spectrum is to remain constant, XBeach will only read from one input file to generate wave boundary conditions. If the wave spectrum is to vary in time, XBeach reads from multiple files.

Whether or not the wave spectrum of the boundary conditions changes over time, the XBeach module requires a record length during which the current wave spectral parameters are to apply. For the duration of the record length, boundary conditions are calculated at every boundary condition file time step. These time steps are not required to be the same as the time steps in the XBeach main program; XBeach will interpolate where necessary. The boundary condition time steps should therefore only be small enough to accurately describe the incoming bound long waves. The statistical data for the generation of the wave boundary conditions is read from user-specified files. At this stage the XBeach module can interpret SWAN 2D variance density output files and JONSWAP parameters.

The beginning and end of the boundary condition file is tapered by the XBeach module. This is done to ensure smooth transitions from one boundary condition file to the next.

The combination of a large record length and a small time step lead to large demands on the system memory. If the memory requirement is too large, the user must choose to either enlarge the boundary condition time step, or to reduce the record length. In case of the latter, several boundary condition files can be generated and read sequentially. It is unwise however to reduce the record length too much, as then the transitions between the boundary condition files start to play an important role.

Every time the XBeach wave boundary condition module is run, it outputs data to the local directory. Metadata about the wave boundary conditions are stored in list files: *ebcflist.bcf* and *qbcflist.bcf*. The main XBeach program uses the list files to know how and when to read and generate boundary condition files. The actual incoming short-wave energy and long-wave mass flux data is stored in other files. These files have E_- and q_- prefixes. The main XBeach program uses these files for the actual forcing along the offshore edge.

5.8   Flow input

The bed friction is influenced by the dimensionless friction coefficient *cf* or the dimensional *C* value. These values are uniform and stationary.

The horizontal viscosity is first composed by adding an overall background viscosity *nuh* and a viscosity depending on the roller dissipation, tuned by *nuhfac*. In the alongshore direction the viscosity may be multiplied by a factor *nuhv* to account for additional advective mixing.

keyword	description	default value	minimum value	maximum value	unit	remarks
cf	flow friction coefficient	0.003	0	0.1	-	
C	Chezy coefficient	sqrt(g/cf)	20	100	$m^{1/2}$/s	alternative for cf, overrules cf if set
nuh	horizontal background viscosity	0.15	0	1	m2/s	
nuhfac	viscosity coefficient for roller induced turbulent horizontal viscosity	0	0	1	-	
nuhv	additional shear dispersion factor	1	1	20	-	svendsen and putrevu (1994)
smag	Smagorinski	1	0	1		
lat	latitude	0	0	90	deg N	
omega	angular velocity of earth	1/24	0	1	1/hour	

5.9   Flow boundary conditions

Flow boundary conditions need to be specified on all sides of the domain. We will differentiate between the offshore, lateral and landward boundaries.

The *carspan* keyword is useful when you want to specify timeseries of free long waves incident on the sea boundary. In that case use the file bc\gen.ezs, specify the time in the first column, the long wave elevation in the second column and zero wave energy in the third column.

keyword	description	default value	minimum value	maximum value	unit	remarks
front	seaward boundary condition	1	0	4	-	see table below
ARC	active reflection compensation at seaward boundary	1	0	1	-	0 = off, 1 = on
order	order of wave steering at seaward boundary	2	1	2	-	1 = first order , 2 = second order
carspan	free long wave input	0	0	1	-	0 = use cg (default); 1 = use sqrt(gh) in *instat* = 3 for c&g tests
back	bayside boundary condition	2	0	3	-	see table below
left	left lateral boundary condition	0	0	1	-	0 = 'neumann', 1 = 'wall'
right	right lateral boundary condition	0	0	1	-	0 = 'neumann', 1 = 'wall'
epsi	weighting factor between steady flow and particle velocity	0	0	1	-	obsolete

front	abbreviated name	description
0	abs1d	absorbing-generating (weakly-reflective) boundary in 1D
1	abs2d	same, in 2D (default setting)
2	wall	no flux wall
3	wlevel	water level specification (from file)
4	nonh_1d	boundary condition for nonhydrostatic option

back	abbreviated name	description
0	wall	no flux wall
1	abs1d	absorbing-generating (weakly-reflective) boundary in 1D
2	abs2d	same, in 2D (default setting)
3	wlevel	water level specification (from file)

5.9.1 Time-varying tide/surge

XBeach can take in up to four time-vary tidal signals to be applied to the four boundaries. A time-varying water level signal is read into XBeach by *readtide.f90*, which is called from *XBeach.f90*. *readtide* opens the specified file in par%zs0file, and uses par%tideloc (number of water level signal locations to be read in) and par%tidelen (length of input signals) to format the file reading statements. The par%zs0file must follow the format with the first column being time and each subsequent column containing the water level signals. As a note, the input signal will be interpolated to the local time step of the simulation; therefore the signals only need to be long enough and temporally-fine enough to resolve the water level phenomenon of interest (i.e. tide variations, surge event). The above parameters and file name are specified in *params.txt*. If par%tideloc is equal to zero, *readtide* is not utilized and a uniform water level is applied according to par%zs0.

There are now four options for handling the tidal and/or surge contribution to the boundaries:
- Uniform water level
- One time-varying water level signal
- Two time-varying water level signals, which requires point of application indication.
- Four time-varying water level signals

The uniform water level is applied according to par%zs0, as stated above. For par%tideloc equal to 1, one water level signal is read and applied to the offshore boundary and par%zs0 is applied to the land boundary. If par%tideloc is equal to 2, two water level signals are read by *readtide*. par%paulrevere is utilized to indicate the locations of application for the two signals. For par%paulrevere equal to 0, one tidal record is applied to both sea corners and one tidal record to both land corners, whereas if par%paulrevere is equal to 1, the first tidal record is applied to the $(x=1,y=1)$ sea corner and the second tidal record (third column) to the $(x=1,y=N)$ sea corner. For par%tideloc equal to 4, four tidal records are read. These four time-varying signals need to be in ordered columns going clockwise around the domain, with first column signal being applied to $(x=1,y=1)$. Therefore, the columns must follow the order of: 1. $(x=1,y=1)$, 2. $(x=1,y=N)$, 3. $(x=N,y=N)$, 4. $(x=N,y=1)$.

The input signal(s) is(are) interpolated spatially along the boundaries for par%tideloc is equal to 2 (if par%paulrevere=2, and if par%paulrevere=1 and no land mass blocks the flow between the land and sea boundaries) and higher. The spatial interpolation occurs initially in *wave_init*, and subsequently at every time step in *flow_bc*. The input signal(s) is(are) also interpolated to the local time step in *flow_bc*.

keyword	description	default value	minimum value	maximum value	unit	remarks
zs01	initial water level	0	-5	5	m	
tideloc	number of input tidal time series	0	0	4	-	
paulrevere	option of sea/sea corner or sea/land corner specification	0	0	1	-	

5.10 Wind

keyword	description	default value	minimum value	maximum value	unit	remarks
rhoa	air density	1.25	1	2	kg/m3	
Cd	wind drag coefficient	0.002	0.0001	0.01	-	
windv	wind velocity	0	0	200	m/s	
windth	wind direction (nautical convention)	270	-360	360	deg	.

The implementation of the wind stress in the momentum balance in XBeach is based upon the work of Ruessink *et al.* (2001). The wind field is specified by the following parameters in *params.txt*:

- par%windth, the wind direction (Nautical coordinates)
- par%windv, the wind velocity specified in m/s
- par%rhoa, the air density (default 1.25 kg/m^3)
- par%Cd, the air drag coefficient (default 0.002, but should be looked into for simulations with wind velocities exceeding 50 m/s)

The input wind direction is converted to radians and Cartesian coordinates prior to use in the momentum balance. The wind contribution is computed as an additional term in the x and y momentum balances.

5.11   Limiters

Especially in very shallow water some processes need to be limited to avoid unrealistic behaviour. The parameters below can be adjusted by the user. Reducing *gammax* will reduce wave heights in very shallow water, probably 2 is a reasonable value. *hmin* prevents very strong return flows or high concentrations. *eps* determines whether points are dry or wet and can be taken quite small. *hwci* limits the computation of wave-current interaction in very shallow water where the procedure may not converge.

keyword	description	default value	minimum value	maximum value	unit	remarks
gammax	maximum ratio Hrms/hh	2	0.4	5	-	
hmin	threshold water depth for concentration and return flow	0.05	0.001	1	m	
eps	threshold depth for drying and flooding	0.005	0.001	0.1	m	
umin	threshold velocity upwind scheme	0.0	0.0	5	m/s	
hwci	depth below which wci is not applied	0.1	0.0001	1	m	

5.12   Sediment transport

keyword	description	default value	minimum value	maximum value	unit	remarks
waveform	option for waveshape model	1	1	1	-	1='ruessink_vanrijn' 2='vanthiel'
form	equilibrium sed. conc. formulation	1	1	2	-	1 = 'soulsby_vanrijn', 2 = 'vanthiel_vanrijn'
thetanum	option to switch between central and upwind scheme	1	0.5	1	[-]	1=upwind 0.5=central
z0	zero flow velocity level in Soulsby van Rijn (1997) sed.conc.	0.006	0.0001	0.05	m	

	expression					
facsl	bed slope factor	0	0	1.6		
BRfac	Calibration factor surface slope	1	0	1		
turb	Equilibrium sediment concentration computation option	2	-	-	-	0='none', 1='wave_averaged' 2='bore_averaged'
rhos	density of sediment (no pores)	2650	2400	2800	kg/ m3	
por	porosity	0.4	0.3	0.5	-	
smax	max shields value for overwash	-1	-1	3	[-]	
lws	longwave stirring	1	0	1		1 = on; 0 = off
sws	shortwave stirring	1	0	1		1 = on; 0 = off
tsfac	max value for fall velocity	0.1	0.01	1	[-]	
facua	asymmetry transport	0	0	1	-	
rfb		1	0	1	-	ONLY IF FORM=2 If rfb = 1 then maximum wave surface slope is fed back in roller energy balance; else rfb = par%Beta

5.13 **Multiple sediment fractions and hard layers**

Keywords related to hard layers and sediment classes:

struct	option for hard structures	0	0	1		0 = no revetment, 1 = multiple sediment classes
ne_layer						

ngd	number of sediment classes	1	1	20	-	
nd	number of sediment class layers	3	3	1000	-	
dzg1 dzg2 dzg3	thickness of sediment class layers	0.1	0.01	1	-	
D50	uniform D50 sediment diameter	0.0002	-	-	m	
D90	uniform D90 sediment diameter	0.0003	-	-	m	
sedcal	factor on sediment transport	1	-	-	-	advanced option, leave at sedcal=1
ucrcal	factor on critical velocity for erosion	1	-	-	-	advanced option, leave at ucrcal=1

==NEEDS MORE EXPLANATION FOR MULTIPLE SEDIMENT FRACTIONS==

5.14   Morphological updating and avalanching

keyword	description	default value	minimum value	maximum value	unit	remarks
morfac	morphological factor	0	0	1000	-	
morstart	start time of morphological updates	120	0	10000	s	
morfacopt	type of morfac updating	1	0	1		1 = on; 0 = off
sourcesink	morphological updating based on sediment transport gradients or source sink	0	0	1	-	0 = gradients 1= source sink
wetslp	critical avalanching slope under water	0.3	0.1	1	-	
dryslp	critical avalanching slope above water	1	0.1	2	-	
hswitch	water depth at interface from wetslp to dryslp	0.1	0.01	1		
dzmax		0.05	0.00	1		

5.15   Groundwater

In order to use the groundwater module in XBeach, a number of parameters have been added to *params.txt*. The keywords, their type, default value and description are given below.

keyword	description	default value	minimum value	maximum value	unit	remarks
gwflow	switch parameter	0	0	1	-	Turns the groundwater module on (1) or off (2)
kx	The Darcy permeability coefficient of the aquifer in x-direction	1E-4	1E-5	1E-2	ms^{-1}	
ky	The Darcy permeability coefficient of the aquifer in y-direction	kx	1E-5	1E-2	ms^{-1}	
kz	The Darcy permeability coefficient of the aquifer in z-direction	kx	1E-5	1E-2	ms^{-1}	
dwetlayer	Thickness of the interaction layer $d_{wetlayer}$	0.2d0	0.01	1	m	
aquiferbot	The bed level of the aquifer is set to the value of *aquiferbot* across the entire domain	minval(zb)-3.d0	-100	100	m	
aquiferbotfile	The initial bed level of the aquifer from file	-			-	This keyword overrides the *aquiferbot* keyword
gw0	The initial	0.d0	-5.0	5.0	m	uniform

	groundwater head					over domain
gw0file	The initial groundwater head from file	-			-	This keyword overrides the *gw0* keyword

Example

An example is given in which the groundwater module is used and the vertical permeability coefficient in the vertical is set differently to that in the horizontal. The initial bed level of the aquifer is read from a file and the initial groundwater head is set to a uniform value. Other parameters are kept at their default values.

```
----------------------------
Groundwater options

gwflow = 1
kx = 0.00015
kz = 0.0005
aquiferbotfile = zbaquifer.txt
gw0 = 0.5
```

5.16   Drifters

keyword	description	default value	minimum value	maximum value	unit	remarks
ndrifter	number of drifters	0	0	50	-	
drifterfile	name of drifter file	-	-	-	-	

5.17   Discharge boundaries (river input)

keyword	description	default value	minimum value	maximum value	unit	remarks
disch_loc_file	discharge locations	-	-	-	-	see description

	file				below
disch_timeseries_file	discharge time series fiel	-	-	-	see description below

The contents of the discharge locations file are a number of records equal to the number of discharge boundary sections, with for each record four numbers:

X_begin Y_begin X_end Y_end

These world coordinates must be chosen such that they are close to the desired boundary and enclose the cell centers of the cells that must be part of the boundary section.

In the discharge time series file, the first column is time in seconds; the next columns give the total discharge time series per section.

5.18 Nonhydrostatic

keyword	description	default value	minimum value	maximum value	unit	remarks
secorder	second order corrections to advection/non-linear terms based on mcCormack scheme	0	0	1	-	
solver_maxit	Maximum number of iterations in the linear SIP solver	30	1	1000		
solver_acc	accuracy with respect to the right-hand side used	0.005	0.00001	0.1		
solver_urelax	Underrelaxation parameter	0.92	0.5	0.99		
solver	Solver used to solve the linear system, 1=SIP, 2=TRIDIAG (only for 1d)	1	0	2		
kdmin	Minimum value of kd (pi/dx > minkd)	0.0	0.0	0.05		
dispc	Coefficient in front of the vertical pressure gradient	1	0.1	2		

Topt	Absolute period to optimize coefficient	10	1	20		

5.19 Data types

XBeach allows the user to choose between three types of output data: regular spatial output, time-averaged spatial output and point output. Of these three types, only the regular spatial output existed in previous versions of XBeach.

keyword	description	default value	minimum value	maximum value	unit	remarks
nglobalvar	number of global output variables	-1	-1	20	-	see below
nmeanvar	number of mean variables	0	0	15	-	
npoints	number of point output locations	0	0	50	-	
nrugauge	number of runup gauges	0	0	50	-	

If nglovalvar is set to '-1' then the standard output variables consist of 'H   ', 'zs   ', 'zs0 ', 'zb ', 'hh ', 'u   ', 'v   ', 'ue ', 've ', 'urms ', 'Fx ', 'Fy ', 'ccg ', 'ceqsg', 'ceqbg', 'Susg ', 'Svsg ', 'E   ', 'R   ', 'D   ', 'DR '

5.19.1 Regular spatial output

Regular spatial output describes the instantaneous state of variables across the entire model domain at various points in time. In previous versions of XBeach, the program automatically generated regular spatial output of 20 variables. In the current version of XBeach, the user may determine how many and which variables are output.

To make use of this option the user must specify the number of output variables required using the *'nglobalvar='* keyword in *params.txt*. The value of *nglobalvar* is always integer. This line is immediately followed by the keywords of the required variables, one per line. Refer to the *Variable keywords* section for a list of keywords. An example of the declaration of regular spatial output is given below.

```
----------------------------
Output options

nglobalvar = 5

zb
zs
u
v
```

Example of the declaration of regular spatial output variables

Output files generated using this option are given the name *keyword.dat*, for example *zs.dat*. The only exception is that files containing information about the wave height of the short waves are called *hrms.dat* instead of *H.dat* to maintain backward compatibility.

5.19.2 Time-averaged spatial output

This output describes the time-averaged state of variables across the entire model domain at various points in time. The period used to average the variables is selected by the user and is described in the section *Choosing output times*. The user may determine how many and which variables are output. To make use of this option the user must specify the number of output variables required using the *'nmeanvar='* keyword in *params.txt*. The value of *nmeanvar* is always integer. This line is immediately followed by the keywords of the required variables, one on each subsequent line. Refer to the *Variable keywords* section for a list of keywords. This statement block may be made before or after the regular spatial output block. An example of the declaration of time-averaged spatial output is given below.

```
----------------------------
Output options

nmeanvar = 2
u
v
```

5.20   Selecting output data types

XBeach v12 allows the user to choose between three types of output data: regular spatial output, time-averaged spatial output and point output. Of these three types, only the regular spatial output existed in previous versions of XBeach.

5.20.1 Regular spatial output

Regular spatial output describes the instantaneous state of variables across the entire model domain at various points in time. In previous versions of XBeach, the program automatically generated regular spatial output of 20 variables. In the current version of XBeach, the user may determine how many and which variables are output.

To make use of this option the user must specify the number of output variables required using the *'nglobalvar='* keyword in *params.txt*. The value of *nglobalvar* is always integer. This line is immediately followed by the keywords of the required variables, one per line. Refer to the *Variable keywords* section for a list of keywords. An example of the declaration of regular spatial output is given below.

```
----------------------------
Output options

nglobalvar = 5
H
zb
zs
u
v
```

Example of the declaration of regular spatial output variables

Output files generated using this option are given the name *keyword.dat*, for example *zs.dat*. The only exception is that files containing information about the wave height of the short waves are called *hrms.dat* instead of *H.dat* to maintain backward compatibility.

5.20.2 Time-averaged spatial output

This output describes the time-averaged state of variables across the entire model domain at various points in time. The period used to average the variables is selected by the user and is described in the section *Choosing output times*. The user may determine how many and which variables are output. To make use of this option the user must specify the number of output variables required using the *'nmeanvar='* keyword in *params.txt*. The value of *nmeanvar* is always integer. This line is immediately followed by the keywords of the required variables, one on each subsequent line. Refer to the *Variable keywords* section for a list of keywords. This statement block may be made before or after the regular spatial output block. An example of the declaration of time-averaged spatial output is given below.

```
----------------------------
Output options

nmeanvar = 2
u
v
```

Example of the declaration of time-averaged spatial output variables

Output files generated using this option are given the name *keyword_mean.dat*, for example *zs_mean.dat*.

5.20.3 Point output

Point output allows the user to select one or more locations for which a time series of data is provided. This output type is divided into two classes: fixed point output and runup gauge output.

Fixed point output

This output describes a time-series of a number of variables at one point in the model domain. To make use of this option, the user must specify the number of output locations using the *'npoints='* keyword in *params.txt*. The value of *npoints* is always integer. This statement is always immediately followed by one line per output location describing the location coordinates, the number of output variables required at that location and the output variable keywords. The location coordinates are given separately as x-coordinate and y-coordinate, in world coordinates. XBeach will link the output location to the nearest computational point. The number of required variables is integer. Variables are stated using their respective variable keywords, see the *Variable keywords* section. Each keyword, including the last keyword, must be followed by a number symbol (#).This statement block may be made before or after any other output option block.

An example is given below in which two output locations are defined. The first point is located on the offshore boundary (x-coordinate = 0.0) and somewhere in the middle of the model domain in y-direction (y-coordinate = 800.0). This location has two output variables, *u* and *v*. The second point is located on the lateral boundary (y-coordinate = 1600.0) and somewhere in the middle of the domain in x-direction (x-coordinate = 2000.0). This location has four output variables, *H*, *zs*, *zb* and *D*.

```
---------------------------
Output options

npoints = 2
0. 800. 2 u#v#
2000. 1600. 4 H#zs#zb#D#
```

Runup gauge output

This output describes a time-series of a number of variables at the waterline. In this case XBeach scans in a x-directional transect defined by the user to find the last wet point before shore (defined as the first dry point in a transect). Output information is recorded for this point. This is particularly useful to keep track of runup levels in cross shore transects.

The definition of runup gauges is similar to the definition of fixed point output. To specify runup gauges, the user should specify the number of runup gauges using the *'nrugauge='* keyword in *params.txt*. The value of *nrugauge* is always integer. This statement is always immediately followed by one line per output location describing the location coordinates, the number of output variables required at that location and the output variable keywords. The location coordinates are given separately as x-coordinate and y-coordinate, in world coordinates. XBeach will link the output location to the nearest computational grid row. The number of required variables is integer. Variables are stated using their respective variable keywords, see the *Variable keywords* section. Each keyword, including the last keyword, must be followed by a number symbol (#).This statement block may be made before or after any other output option block.

Point output file structure

All data corresponding to point locations will be stored in files called *pointxxx.dat* in the case of fixed point data and *rugauxxx.dat* in the case of runup gauge data. In the filenames *xxx* represents a number between 001 and 999, corresponding to the order in which the points are declared in *params.txt*. The data files contain one row per output time step. The first position on each row is the time at which the output is given. The subsequent positions in the row are the values of the variables at the given point at that given moment. The order of the variables is the same as the order in which they are defined for that point in *params.txt*.

5.20.4 Cross-sections

NOT YET IMPLEMENTED

5.20.5 Default values

If no output information is given with respect to the numbers and types of output, XBeach assumes the user requires only the original 20 regular spatial output types. The default number of time-averaged output variables and point output locations is zero.

5.21 Choosing output times

The user may determine the output times for regular spatial output variables, time averaged spatial variables and point location variables individually. Runup gauge output and fixed point output are given at the same points in time. For all three types of output the user may choose to either state a fixed interval time at which output is given, or supply an additional file containing times at which output should be given.

5.21.1 Output at fixed intervals

In this case the user should define a point in time (in seconds) after the start of the simulation at which the first output is generated. The user can do this by using the *'tstart='* keyword in *params.txt*. All output that is being generated at fixed intervals uses *tstart* as their base. The interval for regular global output is given by the *'tintg='* keyword in *params.txt*. The keywords for the interval of time-averaged global output and point output are *'tintm='* and *'tintp='* respectively. Note that *tintg*, *tintm* and *tintp* supersede the older *tint* parameter. An example of the definition of fixed intervals is given below.

```
---------------------------
Output options

tstart = 100.
tintg = 100.
tintp = 2.
tintm = 3600.
```

Example of the declaration of fixed interval output times

In the case of regular global output and point output, the first output is given at *tstart*. In the case of time-averaged global variables, the first output is given at *tstart+tintm*. This output represents the average condition over the interval between *tstart* and *tstart+tintm*.

5.21.2 Output times defined by external file

The user is given the option to have output at a set of points in time that are not separated by regular intervals. In this case the user must supply an additional file for each output type (regular, time-averaged or point). The user specifies the name of the output time series file for regular spatial output using the *'tsglobal='* keyword in *params.txt*. The keywords for time series files for time-averaged output and point output are *'tsmean='* and *'tspoint='* respectively. All time series files must contain on the first line the number of output times (integer) followed by every output time (in seconds after the start of the simulation) on a new line. An example of the definition of varying output time intervals is given below. An example of the contents of one of the time series definition files is given in the second example below.

```
---------------------------
Output options

tsglobal= timeseries1.txt
tspoints = timeseries2.txt
tsmean= timeseries3.txt
```

Example of the declaration of varying output time intervals

```
18
0.05
0.15
0.2
0.8
12.0
12.5
19.124
30.
60.
90.
120.
150.
160.
170.
177.
178.
179.
180.
```

Example of an output time series definition file (in the previous example this could be timeseries1.txt)

In the case of regular global output and point output, the first output is given at the first stated point in time. In the case of time-averaged global variables, the first output is given at the second stated point in time. This output represents the average condition over the interval between first and second stated point in time. Subsequent averaging is made over every interval.

5.21.3 Combinations of fixed interval and external files

The user is allowed to define certain types of output using fixed intervals and others using external files. The use of an external file supersedes the use of fixed intervals. Note that *tstart* will only apply to output of fixed interval type. An example of mixing fixed and varying output time intervals is given below.

```
---------------------------
Output options

tstart = 100.
tintg = 100.
tspoints = timeseries2.txt
tintm = 3600.
```

Example of the declaration of fixed and varying output time intervals

5.21.4 Default values

The default value of *tintg* is one second. If *tintp* or *tintm* is not stated, but output is declared (*npoints*, *nrugauge* or *nmeanvar* is stated larger than zero), XBeach assumes the same output interval as *tintg*. XBeach v12 has been designed to be backward compatible with older versions. Therefore if *tint* is defined, this replaces the default value of *tintg*.

5.22 NetCDF – CF output

Needs to be inserted.

5.23   Reading output files in Matlab

Various functions have been developed to help users read XBeach output in Matlab. The following script can be used to gather information about the XBeach output before reading the output.

```matlab
function XBdims=getdimensions
% XBdims=getdimensions
%
% Must start in XBeach output directory
% Output structure XBdims
% XBdims.nt       = number of regular spatial output timesteps
% XBdims.nx       = number of grid cells in x-direcion
% XBdims.ny       = number of grid cells in y-direcion
% XBdims.ngd      = number of sediment classes
% XBdims.nd       = number of sediment class layers
% XBdims.ntp      = number of point output timesteps
% XBdims.ntm      = number of time-average output timesteps
% XBdims.tsglobal = times at which regular output is given
% XBdims.tspoints = times at which point output is given
% XBdims.tsmean   = times at which time-averaged output is given
% XBdims.x    = x-coordinates grid
% XBdims.y    = y-coordinates grid
%
% Created 19-06-2008 : XBeach-group Delft

fid=fopen('dims.dat','r');
XBdims.nt=fread(fid,[1],'double');
XBdims.nx=fread(fid,[1],'double');
XBdims.ny=fread(fid,[1],'double');
XBdims.ngd=fread(fid,[1],'double');
XBdims.nd=fread(fid,[1],'double');
XBdims.ntp=fread(fid,[1],'double');
XBdims.ntm=fread(fid,[1],'double');
XBdims.tsglobal=fread(fid,[XBdims.nt],'double');
XBdims.tspoints=fread(fid,[XBdims.ntp],'double');
XBdims.tsmean=fread(fid,[XBdims.ntm],'double');
fclose(fid);

fidxy=fopen('xy.dat','r');
XBdims.x=fread(fidxy,[XBdims.nx+1,XBdims.ny+1],'double');
XBdims.y=fread(fidxy,[XBdims.nx+1,XBdims.ny+1],'double');
fclose(fidxy);
```

Matlab function to extract output metadata

Regular spatial output and time-averaged output can then be read using the following function.

```
function Vardata=readvar(fname,XBdims)
% Var=readvar(fname,XBdims)
%
% Output Var is [nx+1,ny+1,nt] array, where nt is XBdims.nt or XBdims.ntm
% Input - fname : name of data file to open, e.g. 'zb.dat' or 'u_mean.dat'
%       - XBdims: dimension data provided by getdimensions function
%
% Created 19-06-2008 : XBeach-group Delft

if (length(fname)>9 && strcmp(fname(end-8:end), '_mean.dat'))
    nt=XBdims.ntm;
else
    nt=XBdims.nt;
end
fid=fopen(fname,'r');
Vardata=zeros(XBdims.nx+1,XBdims.ny+1,nt);
for i=1:nt
    Vardata(:,:,i)=fread(fid,size(XBdims.x),'double');
end
fclose(fid)
```

Matlab function to read spatial output variables

To read point data the following function can be used.

```
function Pointdata=readpoint(fname,XBdims,nvar)
% Pointdata=readpoint(fname,XBdims,nvar)
%
% Output Point is [ntp,nvar+1] array, where ntp is XBdims.ntp
%               First column of Pointdata is time
%               Second and further columns of Pointdata are values of
%               variables
% Input - fname : name of data file to open, e.g. 'point001.dat' or 'rugau001.dat'
%       - XBdims: dimension data provided by getdimensions function
%       - nvar  : number of variables output at this point location
%
% Created 19-06-2008 : XBeach-group Delft

Pointdata=zeros(XBdims.ntp,nvar+1);
fid=fopen(fname,'r');
for i=1:XBdims.ntp
    Pointdata(i,:)=fread(fid,nvar+1,'double');
end
fclose(fid);
```

Matlab function to read point output variables

5.24   Output variable keywords

All variables in spaceparams.tmpl can be defined as output.

real*8 2 x s%nx+1 s%ny+1 d [m]   x-coord. comp. grid (positive shoreward, perp. to coastline)

real*8 2 y s%nx+1 s%ny+1 d [m]   y-coord. comp. grid

real*8 1 xz s%nx+1 d [m]   x-coord. comp. grid (positive shoreward, perp. to coastline)

real*8 1 yz s%ny+1 d [m]   y-coord. comp. grid

real*8 1 xu s%nx+1 d [m]   x-coord. in u points

real*8 1 yv s%ny+1 d [m]   y-coord. in v points

real*8 2 xw s%nx+1 s%ny+1 d [m]   world x-coordinates

real*8 2 yw s%nx+1 s%ny+1 d [m]   world y-coordinates

real*8 0 dx b [m]   grid size x-direction

real*8 0 dy b [m]   grid size y-direction

real*8 0 xori b [m]   x-origin of grid in world coordinates

real*8 0 yori b [m]   y-origin of grid in world coordinates

real*8 0 alfa b [rad]   (deg on input) angle of grid w.r.t. East

real*8 0 posdwn b [-]   depths defined positive downwards (1) or upwards(-1)

real*8 0 thetamin b [rad]   minimum angle of computational wave grid (carth. in rad)

real*8 0 thetamax b [rad]   minimum angle of computational wave grid (carth. in rad)

integer 0 nx   b [-]   local number of grid cells x-direction

integer 0 ny   b [-]   local number of grid cells y-direction

real*8 2 zb   s%nx+1 s%ny+1 d [m]   bed level

real*8 2 zb0 s%nx+1 s%ny+1 d [m]   initial bed level

real*8 1 theta s%ntheta b [rad]   wave angles directional distribution w.r.t. comp. x-axis

integer 0 ntheta b [-]   number of wave direction bins

real*8 0 dtheta b [rad]   wave direction bin size

real*8 0 theta0 b [rad]   mean incident wave angle

real*8 1 cxsth s%ntheta b [-]   cos(theta)

real*8 1 sxnth s%ntheta b [-]   sin(theta)

real*8 2 thetamean s%nx+1 s%ny+1 d [rad]   mean wave angle

real*8 2 Fx   s%nx+1 s%ny+1 d [N/m2]   wave force x-direction

real*8 2 Fy   s%nx+1 s%ny+1 d [N/m2]   wave force y-direction

real*8 2 Sxy   s%nx+1 s%ny+1 d [N/m]   radiation stress

real*8 2 Syy   s%nx+1 s%ny+1 d [N/m]   radiation stress

real*8 2 Sxx   s%nx+1 s%ny+1 d [N/m]   radiation stress

real*8 2 n   s%nx+1 s%ny+1 d [-]   ratio group velocity/wave celerity

real*8 2 H   s%nx+1 s%ny+1 d [m]   wave height

real*8 3 cgx   s%nx+1 s%ny+1 s%ntheta d [m/s]   group velocity x-direction

real*8 3 cgy   s%nx+1 s%ny+1 s%ntheta d [m/s]   group velocity y-direction

real*8 3 cx   s%nx+1 s%ny+1 s%ntheta d [m/s]   wave celerity x-direction

real*8 3 cy   s%nx+1 s%ny+1 s%ntheta d [m/s]   wave celerity y-direction

real*8 3 ctheta s%nx+1 s%ny+1 s%ntheta d [rad/s]   wave celerity theta-direction (refraction)

real*8 3 ee s%nx+1 s%ny+1 s%ntheta d [J/m2/rad] directionally distributed wave energy

real*8 3 thet s%nx+1 s%ny+1 s%ntheta d [rad] wave angles

real*8 3 costhet s%nx+1 s%ny+1 s%ntheta d [-]   cos of wave angles

real*8 3 sinthet s%nx+1 s%ny+1 s%ntheta d [-]   sin of wave angles

real*8 3 sigt s%nx+1 s%ny+1 s%ntheta d [rad/s] relative frequency

real*8 3 rr s%nx+1 s%ny+1 s%ntheta d [J/m2/rad] directionally distributed roller energy

real*8 2 k s%nx+1 s%ny+1 d [rad/m] wave number

real*8 2 c s%nx+1 s%ny+1 d [m/s] wave celerity

real*8 2 cg s%nx+1 s%ny+1 d [m/s] group velocity

real*8 2 sigm s%nx+1 s%ny+1 d [rad/s] mean frequency

real*8 2 wm s%nx+1 s%ny+1 d [rad/s] mean abs frequency

real*8 2 hh s%nx+1 s%ny+1 d [m] water depth

real*8 2 zs s%nx+1 s%ny+1 d [m] water level

real*8 2 zs0 s%nx+1 s%ny+1 d [m] water level due to tide alone

real*8 1 tideinpt par%tidelen b [s] input time of input tidal signal

real*8 2 tideinpz par%tidelen par%tideloc b [m] input tidal signal

real*8 1 windinpt par%windlen b [s] input time of input wind signal

real*8 1 windvel par%windlen b [m/s] input wind velocity

real*8 1 winddir par%windlen b [deg nautical] input wind direction

real*8 0 windvnow b [m/s] uniform wind velocity current time

real*8 0 winddirnow b [m/s] uniform wind direction current time

real*8 2 dzsdt s%nx+1 s%ny+1 d [m/s] rate of change water level

real*8 2 dzsdx s%nx+1 s%ny+1 d [m/s] water surface gradient in x-direction

real*8 2 dzsdy s%nx+1 s%ny+1 d [m/s] water surface gradient in y-direction

real*8 2 dzbdx s%nx+1 s%ny+1 d [-] bed level gradient in x-direction

real*8 2 dzbdy s%nx+1 s%ny+1 d [-] bed level gradient in y-direction

real*8 2 dzbdt s%nx+1 s%ny+1 d [m/s] rate of change bed level

real*8 2 uu s%nx+1 s%ny+1 d [m/s] (GLM) x-velocity in u-points

real*8 2 vv s%nx+1 s%ny+1 d [m/s] (GLM) y-velocity in v-points

real*8 2 qx s%nx+1 s%ny+1 d [m2/s] x-discharge in u-points

real*8 2 qy s%nx+1 s%ny+1 d [m2/s] y-discharge in u-points

real*8 2 sedero s%nx+1 s%ny+1 d [m] cum. sedimentation/erosion

real*8 2 dcbdx s%nx+1 s%ny+1 d [kg/m3/m] bed concentration gradient x-dir.

real*8 2 dcbdy s%nx+1 s%ny+1 d [kg/m3/m] bed concentration gradient y-dir.

real*8 2 dcsdx s%nx+1 s%ny+1 d [kg/m3/m] suspended concentration gradient x-dir.

real*8 2 dcsdy s%nx+1 s%ny+1 d [kg/m3/m] suspended concentration gradient y-dir.

real*8 2 ui s%nx+1 s%ny+1 d [m/s] incident bound wave velocity

real*8 2 E s%nx+1 s%ny+1 d [Nm/m2] wave energy

real*8 2 R s%nx+1 s%ny+1 d [Nm/m2] roller energy

real*8 2 urms s%nx+1 s%ny+1 d [m/s] orbital velocity

real*8 2 D s%nx+1 s%ny+1 d [W/m2] dissipation

real*8 2 Qb s%nx+1 s%ny+1 d [-] fraction breaking waves

real*8 2 ust s%nx+1 s%ny+1 d [m/s] Stokes drift

real*8 2 tm s%nx+1 s%ny+1 d [rad] mean wave direction

real*8 2 ueu s%nx+1 s%ny+1 d [m/s] Eulerian mean velocity x-dir.

real*8 2 vev   s%nx+1 s%ny+1 d [m/s]   Eulerian mean velocity y-dir.

real*8 2 vmagu s%nx+1 s%ny+1 d [m/s]   (GLM) velocity magnitude u-points

real*8 2 vmageu s%nx+1 s%ny+1 d [m/s]   (GLM) velocity magnitude u-points

real*8 2 vmagv s%nx+1 s%ny+1 d [m/s]   (GLM) velocity magnitude v-points

real*8 2 vmagev s%nx+1 s%ny+1 d [m/s]   (GLM) velocity magnitude v-points

real*8 2 u   s%nx+1 s%ny+1 d [m/s]   (GLM) x-velocity cell centre (for output)

real*8 2 v   s%nx+1 s%ny+1 d [m/s]   (GLM) y-velocity cell centre (for output)

real*8 2 ue   s%nx+1 s%ny+1 d [m/s]   Eulerian mean x-velocity cell centre (for output)

real*8 2 ve   s%nx+1 s%ny+1 d [m/s]   Eulerian mean y-velocity cell centre (for output)

real*8 2 hold   s%nx+1 s%ny+1 d [m]   water depth previous time step

integer 2 wetu s%nx+1 s%ny+1 d [-]   mask wet/dry u-points

integer 2 wetv s%nx+1 s%ny+1 d [-]   mask wet/dry v-points

integer 2 wetz s%nx+1 s%ny+1 d [-]   mask wet/dry eta-points

real*8 2 hu   s%nx+1 s%ny+1 d [m]   water depth in u-points

real*8 2 hv   s%nx+1 s%ny+1 d [m]   water depth in v-points

real*8 2 hum   s%nx+1 s%ny+1 d [m]   water depth in u-points

real*8 2 hvm   s%nx+1 s%ny+1 d [m]   water depth in v-points

real*8 2 vmag   s%nx+1 s%ny+1 d [m/s]   velocity magnitude in cell centre

real*8 3 ccg   s%nx+1 s%ny+1 par%ngd d [m3/m3]   depth-averaged suspended concentration for each sediment fraction

real*8 3 ccbg   s%nx+1 s%ny+1 par%ngd d [m3/m3]   depth-averaged bed concentration for each sediment fraction

real*8 2 uwf   s%nx+1 s%ny+1 d [m/s]   x-comp. Stokes drift

real*8 2 vwf   s%nx+1 s%ny+1 d [m/s]   y-comp. Stokes drift

real*8 2 ustr   s%nx+1 s%ny+1 d [m/s]   return flow due to roller

real*8 2 usd   s%nx+1 s%ny+1 d [m/s]   return flow due to roller after breaker delay

real*8 1 bi   s%ny+1 d [m]   incoming bound long wave

real*8 2 DR   s%nx+1 s%ny+1 d [W/m2]   roller energy dissipation

real*8 2 umean 2 s%ny+1 2 [m/s]   longterm mean velocity at bnds in x-direction

real*8 2 vmean 2 s%ny+1 2 [m/s]   longterm mean velocity at bnds in y-direction

integer 0 vardx b [-]   0 = uniform grid size, 1 = variable grid size

real*8 2 vu   s%nx+1 s%ny+1 d [m/s]   y velocity in u points

real*8 2 uv   s%nx+1 s%ny+1 d [m/s]   x velocity in v points

real*8 1 D50   par%ngd b [m]   D50 grain diameters for all sediment classses

real*8 1 D90   par%ngd b [m]   D90 grain diameters for all sediment classses

real*8 1 sedcal par%ngd b [-]   equilibrium sediment concentartion factor for each sediment class

real*8 1 ucrcal par%ngd b [-]   calibration factor for u critical for each sediment class

real*8 3 Tsg   s%nx+1 s%ny+1 par%ngd d [s]   sediment response time for each sediment class

real*8 3 Susg   s%nx+1 s%ny+1 par%ngd d [m2/s]   suspended sediment transport x-dir. for each sediment class (excluding pores)

real*8 3 Svsg   s%nx+1 s%ny+1 par%ngd d [m2/s]   suspended sediment transport y-dir. for each sediment class (excluding pores)

real*8 3 Subg   s%nx+1 s%ny+1 par%ngd d [m2/s]   bed sediment transport x-dir. for each sediment class (excluding pores)

real*8 3 Svbg   s%nx+1 s%ny+1 par%ngd d [m2/s]   bed sediment transport y-dir. for each sediment class (excluding pores)

real*8 3 ceqbg s%nx+1 s%ny+1 par%ngd d [m3/m3]   depth-averaged bed equilibrium concentration for each sediment class

real*8 3 ceqsg s%nx+1 s%ny+1 par%ngd d [m3/m3] depth-averaged suspended equilibrium concentration for each sediment class

real*8 2 ua   s%nx+1 s%ny+1 d [m/s]   time averaged flow velocity due to wave assymetry

real*8 2 BR   s%nx+1 s%ny+1 d [-]   maximum wave surface slope used in roller dissipation formulation

real*8 2 kb s%nx+1 s%ny+1 d [m^2/s^2] near bed turbulence intensity due to depth induces breaking

real*8 2 Tbore s%nx+1 s%ny+1 d [s]   wave period interval associated with breaking induced turbulence

real*8 2 dzav s%nx+1 s%ny+1 d [m]   total bed level change due to avalanching

real*8 2 maxzs s%nx+1 s%ny+1 d [m]   maximum elevation in simulation

real*8 2 minzs s%nx+1 s%ny+1 d [m]   minimum elevation in simulation

real*8 2 L1   s%nx+1 s%ny+1 d for subroutine dispersion

real*8 2 Sk s%nx+1 s%ny+1 d [-] skewness of short waves

real*8 2 As s%nx+1 s%ny+1 d [-] asymmetry of short waves

!   real*8 2 Fimpact s%nx+1 s%ny+1 d ! [N/m^2]   Force on dune face

real*8 2 gwhead   s%nx+1 s%ny+1 d [m] groundwater head (differs from gwlevel)

real*8 2 gwlevel s%nx+1 s%ny+1 d [m] groundwater table (min(zb,gwhead))

real*8 2 gwheight s%nx+1 s%ny+1 d [m] vertical size of aquifer through which groundwater can flow

real*8 2 gwbottom s%nx+1 s%ny+1 d [m] level of the bottom of the aquifer

real*8 2 gwu   s%nx+1 s%ny+1 d [m/s] groundwater flow in x-direction

real*8 2 gwv   s%nx+1 s%ny+1 d [m/s] groundwater flow in y-direction

real*8 2 gww   s%nx+1 s%ny+1 d [m/s] groundwater flow in z-direction (interaction between surface and ground water)

real*8 2 dinfil   s%nx+1 s%ny+1 d [m] Infiltration layer depth used in quasi-vertical flow model for groundwater

real*8 2 gw0back 2   s%ny+1 2 [m] boundary condition back boundary for groundwater head

real*8 2 kturb   s%nx+1 s%ny+1 d [m^2/s^2] depth averaged turbulence intensity due to long wave breaking

real*8 3 ero   s%nx+1 s%ny+1 par%ngd d [m/s] bed erosion rate per fraction

real*8 3 depo_im   s%nx+1 s%ny+1 par%ngd d [m/s] implicit bed deposition rate per fraction

real*8 3 depo_ex   s%nx+1 s%ny+1 par%ngd d [m/s] explicit bed deposition rate per fraction

integer 2 nd   s%nx+1 s%ny+1 d [-] number of bed layers (can be different for each computational cell)

real*8 4 pbbed   s%nx+1 s%ny+1 max(par%nd,2) par%ngd d [-] NO DESCRIPTION

real*8 3 dzbed   s%nx+1 s%ny+1 max(par%nd,2) d [-] NO DESCRIPTION

real*8 2 z0bed   s%nx+1 s%ny+1 d [-] NO DESCRIPTION

real*8 2 ureps   s%nx+1 s%ny+1 d [m/s] representative flow velocitie for sediment advection and diffusion in x-direction

real*8 2 vreps   s%nx+1 s%ny+1 d [m/s] representative flow velocitie for sediment advection and diffusion in y-direction

real*8 2 urepb   s%nx+1 s%ny+1 d [m/s] representative flow velocitie for sediment advection and diffusion in x-direction

real*8 2 vrepb   s%nx+1 s%ny+1 d [m/s] representative flow velocitie for sediment advection and diffusion in y-direction

real*8 2 umwci   s%nx+1 s%ny+1 d [m/s]   u-velocity (time-averaged) for wci

real*8 2 rolthick s%nx+1 s%ny+1 d [m] long wave roller thickness

real*8 2 vmwci   s%nx+1 s%ny+1 d [m/s]   v-velocity (time-averaged) for wci

real*8 2 zswci   s%nx+1 s%ny+1 d [m]   waterlevel (time-averaged) for wci

real*8 2 pres   s%nx+1 s%ny+1 d [m^2/s^2] normalized dynamic pressure

real*8 2 wb   s%nx+1 s%ny+1 d [m/s]   vertical velocity at the bottom

real*8 2 ws   s%nx+1 s%ny+1 d [m/s]   vertical velocity at the free surface

real*8 2 taubx   s%nx+1 s%ny+1 d [N/m^2]   x-component of bed shear stress

real*8 2 tauby   s%nx+1 s%ny+1 d [N/m^2]   y-component of bed shear stress

real*8 2 Df   s%nx+1 s%ny+1 d [W/m^2]   dissipation rate due to bed friction

real*8 2 Dp   s%nx+1 s%ny+1 d [W/m^2]   dissipation rate in the swash due to transformation of kinetic wave energy to potential wave energy

real*8 2 Sutot   s%nx+1 s%ny+1 d [m2/s]   Sediment transport integrated over bed load and suspended and for all sediment grains

real*8 2 Svtot   s%nx+1 s%ny+1 d [m2/s]   Sediment transport integrated over bed load and suspended and for all sediment grains

real*8 2 cctot   s%nx+1 s%ny+1 d [m3/m3]   Sediment concentration integrated over bed load and suspended and for all sediment grains

real*8 2 wi   2   s%ny+1 2 [m/s]   Vertical velocity at boundary due to (short) waves2

real*8 2 zi   2   s%ny+1 2 [m]   Surface elevation at boundary due to (short) waves

real*8 2 nuh   s%nx+1 s%ny+1 d [m2/s] horizontal viscosity coefficient

real*8 2 cf   s%nx+1 s%ny+1 d [-] Friction coefficient flow

real*8 2 D50top   s%nx+1 s%ny+1 d [-] Friction coefficient flow

real*8 2 D90top   s%nx+1 s%ny+1 d [-] Friction coefficient flow

real*8 2 structdepth   s%nx+1 s%ny+1 d [m] Depth of structure in relation to instantaneous bed level

5.25   Example params.txt file

The following example shows a partial *params.txt* file in which all three types of output are specified.

```
Output options

tstart = 100.
tintg = 60.
tintm = 720.
tspoints = timespoints.txt

nglobalvar = 4
zs
zb
H
ue

nmeanvar = 2
ue
H

npoints = 3
124 12 2 H#zs#
174 12 2 H#zs#
199 12 4 H#zs#ue#ve#

nrugauge = 1
12 3 zs#ue#ve#

```

Example of a shortened params.txt file with various output options

6   References

Abbott, M.B., 1979. Computational Hydraulics. Elements of the theory of free surface flow. Pitman Publishing, London, 324 pp.

Arcilla, A.S., Roelvink, J.A., O'Connor, B.A. Reniers, A., and Jimenez. J.A. (1994). The Delta Flume '93 Experiment. Coastal Dynamics '94. Arcilla, Stive and Kraus (eds), ASCE, New York, pp. 488-502.

Carrier, G.F. and H.P. Greenspan (1958). Water waves of finite amplitude on a sloping beach. J. Fluid Mech., Vol. 4, pp. 97-109.

Galapatti, R., 1983. A depth integrated model for suspended transport. Report 83-7, Communications on Hydraulics, Department of Civil Engineering, Delft University of Technology.

Reniers, A.J.H.M., J.A. Roelvink and E.B. Thornton. (2004). Morphodynamic modelling of an embayed beach under wave group forcing. J. of Geophysical Res. , VOL. 109, C01030, doi:10.1029/2002JC001586, 2004

Reniers, A.J.H.M., E.B. Thornton, T. Stanton and J.A. Roelvink. (2004b) Vertical flow structure during Sandy Duck: Observations and Modeling. Coastal Engineering, Volume 51, Issue 3, May 2004, Pages 237-260

Roelvink, J.A. (1993a) Dissipation in random wave groups incident on a beach. Coastal Eng., 19 (1993) pp. 127-150.

Roelvink, J.A. (1993b) Surf beat and its effect on cross-shore profiles. Ph.D. thesis, Delft University of Technology.

Ruessink, B.G., J.R. Miles, F. Feddersen, R.T. Guza and S. Elgar, 2001. Modeling the alongshore current on barred beaches. *Journal of Geophysical Research*, 106, 22451-22464.

Soulsby, R. (1997). Dynamics of marine sands. Thomas Telford Publications, London, ISBN 0 7277 2584 X.

Stelling GS, Duinmeijer SPA. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. International Journal for Numerical Methods in Fluids. 2003; 43:1329-1354.

Van Dongeren, A.R. and I.A. Svendsen (1997). An Absorbing-Generating Boundary condition for Shallow Water Models. J. of Waterways, Ports, Coastal and Ocean Engineering, vol. 123, no. 6, pp. 303-313.

Van Dongeren, A.R., A.J.H.M. Reniers, J.A. Battjes and I.A. Svendsen (2003). "Numerical modeling of infragravity wave response during Delilah." J. Geoph. Res, 108, (C9) , 3288, doi:10.1029/2002JC001332

van Gent, M.R.A., 1995. Wave Interaction with Permeable Coastal Structures, Delft University of Technology, Delft.

Verboom, G.K., G.S. Stelling and M.J. Officier (1981). Boundary conditions for the shallow water equations. In: Abbott, M.B. and J.A. Cung, eds. Engineering Applications of Computational Hydraulics, vol. 1., pp. 230-262.

A   Description of gen.ezs file (instat+2,3)

This file should be in subdirectory bc and has a fixed name of *gen.ezs*.
The contents are as follows:
- An arbitrary header line
- Number of time points, number of columns
- Values for time, long wave elevation and wave energy (3 columns)

Example:

```
BL01
10 3
0    0.10   0
2    0.25 200
4    0.20 300
6    0.15  250
```

B   Numerical implementation

B.1   Wave action equation

We use an up-wind schematisation to solve the wave action balance (keyword "scheme = 1' and activated by default). The wave action is given at the same points at the water level. The advection of wave action is then discretized as follows:

$$\frac{\partial c_x^n A^n}{\partial x}(i,j,k) = \frac{c_{x,i,j,k}^n A_{i,j,k}^n - c_{x,i-1,j,k}^n A_{i-1,j,k}^n}{x_{i,j} - x_{i-1,j}} \quad , c_{x,i,j,k}^n > 0$$

$$\frac{\partial c_x^n A^n}{\partial x}(i,j,k) = \frac{c_{x,i+1,j,k}^n A_{i+1,j,k}^n - c_{x,i,j,k}^n A_{i,j,k}^n}{x_{i+1,j} - x_{i,j}} \quad , c_{x,i,j,k}^n < 0$$

(B.1)

The discretization using a Lax-Wendroff scheme ("scheme = 2") is:

$$\frac{\partial c_x^n A^n}{\partial x}(i,j,k) = \frac{c_{x,i+1,j,k}^n A_{i+1,j,k}^n - c_{x,i-1,j,k}^n A_{i-1,j,k}^n}{x_{i+1,j} - x_{i-1,j}} -$$

$$\frac{\Delta t}{2} \frac{\left[c_{x,i+1,j,k}^n \right]^2 A_{i+1,j,k}^n - 2\left[c_{x,i,j,k}^n \right]^2 A_{i,j,k}^n + \left[c_{x,i-1,j,k}^n \right]^2 A_{i-1,j,k}^n}{\left(x_{i+1,j} - x_{i,j} \right)\left(x_{i,j} - x_{i-1,j} \right)}$$

(B.2)

In y-direction, the upwind discretization is

$$\frac{\partial c_y^n A^n}{\partial y}(i,j,k) = \frac{c_{y,i,j,k}^n A_{i,j,k}^n - c_{y,i,j-1,k}^n A_{i,j-1,k}^n}{y_{i,j} - y_{i,j-1}} \quad , c_{y,i,j,k}^n > 0$$

$$\frac{\partial c_y^n A^n}{\partial y}(i,j,k) = \frac{c_{y,i,j+1,k}^n A_{i,j+1,k}^n - c_{y,i,j,k}^n A_{i,j,k}^n}{y_{i,j+1} - y_{i,j}} \quad , c_{y,i,j,k}^n < 0$$

(B.3)

In the Lax-Wendroff discretization:

$$\frac{\partial c_y^n A^n}{\partial y}(i,j,k) = \frac{c_{y,i,j+1,k}^n A_{i,j+1,k}^n - c_{y,i,j-1,k}^n A_{i,j-1,k}^n}{y_{i,j+1} - y_{i,j-1}} - \frac{\Delta t}{2} \frac{\left[c_{y,i,j+1,k}^n \right]^2 A_{i,j+1,k}^n - 2\left[c_{y,i,j,k}^n \right]^2 A_{i,j,k}^n + \left[c_{y,i,j-1,k}^n \right]^2 A_{i,j-1,k}^n}{\left(y_{i,j+1} - y_{i,j} \right)\left(y_{i,j} - y_{i,j-1} \right)}$$

(B.4)

In theta-space an upwind scheme is used only:

$$\frac{\partial c_\theta^n A^n}{\partial \theta}(i,j,k) = \frac{c_{\theta,i,j,k}^n A_{i,j,k}^n - c_{\theta,i,j,k-1}^n A_{i,j,k-1}^n}{\theta_{i,j,k} - \theta_{i,j,k-1}} \quad , c_{\theta,i,j,k}^n > 0$$

$$\frac{\partial c_\theta^n A^n}{\partial \theta}(i,j,k) = \frac{c_{\theta,i,j,k+1}^n A_{i,j,k+1}^n - c_{\theta,i,j,k}^n A_{i,j,k}^n}{\theta_{i,j,k+1} - \theta_{i,j,k}} \quad , c_{\theta,i,j,k}^n < 0$$

(B.5)

Similar for the wave action balance:

$$\frac{A_{i,j,k}^{n+1} - A_{i,j,k}^{n}}{\Delta t} = -\frac{\partial c_x^n A^n}{\partial x}\bigg|_{i,j,k} - \frac{\partial c_y^n A^n}{\partial y}\bigg|_{i,j,k} - \frac{\partial c_\theta^n A^n}{\partial \theta}\bigg|_{i,j,k} - \frac{D}{\sigma}\bigg|_{i,j,k} \tag{B.6}$$

We apply an upwind schematisation, since the horizontal scale of the problem is limited and such a scheme deals with shocks in a natural way.

B.2   Shallow water equations

We apply a staggered grid, where bed levels and water levels are defined in the centre of cells, and velocity components at the cell interfaces.

If *nx,ny* are the number of cells in both directions, the water level points are numbered from 1 to *nx+1* and from 1 to *ny+1*.

The water level gradients are computed at the cell interfaces and are given by:

$$\frac{\partial \eta}{\partial x}(i,j) = \frac{\eta_{i+1,j} - \eta_{i,j}}{x_{i+1,j} - x_{i,j}} \tag{B.7}$$

$$\frac{\partial \eta}{\partial y}(i,j) = \frac{\eta_{i,j+1} - \eta_{i,j}}{x_{i,j+1} - x_{i,j}} \tag{B.8}$$

For computing the shear stresses at the cell interfaces we need the velocity magnitudes at these interfaces. These are composed by combining the normal velocity component at the interface and the average of the 4 adjacent tangential components:

$$v_{u,i,j} = \frac{1}{4}(v_{i,j-1} + v_{i,j} + v_{i+1,j-1} + v_{i+1,j})$$

$$u_{v,i,j} = \frac{1}{4}(u_{i-1,j} + u_{i,j} + u_{i-1,j+1} + u_{i,j+1}) \tag{B.9}$$

The water depth in each cell is computed as:

$$h_{i,j} = \eta_{i,j} - z_{b,i,j} \tag{B.10}$$

For the depth at cell interfaces, following Stelling and Duinmeijer (2003) we distinguish between the depth used in the continuity equation and that used in the momentum equation. The depth at the interfaces *for the continuity* equation is taken as the upwind depth in case the velocity is greater than a minimum velocity, or the maximum water level minus the maximum bed level in case the velocity is less than this minimum velocity:

$$
\begin{aligned}
h_{u,i,j} &= h_{i,j} & , u_{i,j} &> u_{min} \\
h_{u,i,j} &= h_{i+1,j} & , u_{i,j} &< -u_{min} \\
h_{u,i,j} &= \max(z_{s,i,j}, z_{s,i+1,j}) - \max(z_{b,i,j}, z_{b,i+1,j}) & , |u_{i,j}| &< u_{min}
\end{aligned}
\tag{B.11}
$$

$$h_{v,i,j} = h_{i,j} \qquad\qquad\qquad\qquad\qquad\qquad\qquad , v_{i,j} > v_{\min}$$

$$h_{v,i,j} = h_{i,j+1} \qquad\qquad\qquad\qquad\qquad\qquad , v_{i,j} < -v_{\min} \qquad\qquad \text{(B.12)}$$

$$h_{v,i,j} = \max(z_{s,i,j}, z_{s,i,j+1}) \; - \; \max(z_{b,i,j}, z_{b,i,j+1}) \quad , \left| v_{i,j} \right| < v_{\min}$$

For the depth *in the momentum balance* we take the average depth between the cell centers:

$$h_{mu,i,j} = \frac{1}{2}(h_{i,j} + h_{i+1,j}) \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(2.13)}$$

$$h_{mv,i,j} = \frac{1}{2}(h_{i,j} + h_{i,j+1}), \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(2.14)}$$

The advection terms in x-direction are approximated as follows:

$$u\frac{\partial u}{\partial x}^{n}_{i,j} = \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i-1,j}u_{i-1,j}}{h_{mu,i,j}}\frac{u^n_{i,j} - u^n_{i-1,j}}{x^n_{i,j} - x^n_{i-1,j}} \qquad , u^n_{i,j} > 0$$

$$u\frac{\partial u}{\partial x}^{n}_{i,j} = \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i+1,j}u_{i+1,j}}{h_{mu,i,j}}\frac{u^n_{i+1,j} - u^n_{i,j}}{x^n_{i+1,j} - x^n_{i,j}} \qquad , u^n_{i,j} < 0$$

(B.15)

$$v\frac{\partial u}{\partial y}^{n}_{i,j} = \frac{1}{2}\frac{h_{u,i,j}v^{n}_{u,i,j} + h_{u,i,j-1}v^{n}_{u,i,j-1}}{h_{mu,i,j}}\frac{u^n_{i,j} - u^n_{i,j-1}}{y^n_{i,j} - y^n_{i,j-1}} \quad , v^{n}_{u,i,j} > 0$$

$$v\frac{\partial u}{\partial y}^{n}_{i,j} = \frac{1}{2}\frac{h_{u,i,j}v^{n}_{u,i,j} + h_{u,i,j+1}v^{n}_{u,i,j+1}}{h_{mu,i,j}}\frac{u^n_{i,j+1} - u^n_{i,j}}{y^n_{i,j+1} - y^n_{i,j}} \quad , v^{n}_{u,i,j} > 0$$

(B.16)

The advection terms in y-direction are approximated as follows:

$$v\frac{\partial v}{\partial y}^{n}_{i,j} = \frac{1}{2}\frac{h^n_{v,i,j}v^n_{i,j} + h^n_{v,i,j-1}v^n_{i,j-1}}{h^n_{mv,i,j}}\frac{v^n_{i,j} - v^n_{i,j-1}}{y^n_{i,j} - y^n_{i,j-1}} \qquad , v^n_{i,j} > 0$$

$$v\frac{\partial v}{\partial y}^{n}_{i,j} = \frac{1}{2}\frac{h^n_{v,i,j}v^n_{i,j} + h^n_{v,i,j+1}v^n_{i,j+1}}{h^n_{mv,i,j}}\frac{v^n_{i,j+1} - v^n_{i,j}}{y^n_{i,j+1} - y^n_{i,j}} \qquad , v^n_{i,j} < 0$$

(B.17)

$$u\frac{\partial v}{\partial x}^{n}_{i,j} = \frac{1}{2}\frac{h_{v,i,j}u^{n}_{v,i,j} + h_{v,i-1,j}u^{n}_{v,i-1,j}}{h_{mv,i,j}}\frac{v^n_{i,j} - v^n_{i-1,j}}{x^n_{i,j} - x^n_{i-1,j}}, \qquad u^{n}_{v,i,j} > 0$$

$$u\frac{\partial v}{\partial x}^{n}_{i,j} = \frac{1}{2}\frac{h_{v,i,j}u^{n}_{v,i,j} + h_{v,i+1,j}u^{n}_{v,i+1,j}}{h_{mv,i,j}}\frac{v^n_{i+1,j} - v^n_{i,j}}{x^n_{i+1,j} - x^n_{i,j}}, \qquad u^{n}_{v,i,j} > 0$$

(B.18)

The momentum equation is discretized as follows:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = -u\frac{\partial u}{\partial x}^{n}_{i,j} - v\frac{\partial u}{\partial y}^{n}_{i,j} - c_f\frac{u_{i,j}^{n}}{h_{u,i,j}^{n}}\sqrt{1.16u_{rms}^2 + u_{i,j}^{n\,2} + v_{u,i,j}^{n\,2}} - g\frac{\eta_{i+1,j}^{n} - \eta_{i,j}^{n}}{x_{i+1,j} - x_{i,j}} + \frac{F_{x,i,j}}{\rho h_{u,i,j}}$$

$$(B.19)$$

$$\frac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta t} = -v\frac{\partial v}{\partial y}^{n}_{i,j} - u\frac{\partial v}{\partial x}^{n}_{i,j} - c_f\frac{v_{i,j}^{n}}{h_{v,i,j}^{n}}\sqrt{1.16u_{rms}^2 + u_{v,i,j}^{n\,2} + v_{i,j}^{n\,2}} - g\frac{\eta_{i,j+1}^{n} - \eta_{i,j}^{n}}{y_{i,j+1} - y_{i,j}} + \frac{F_{y,i,j}}{\rho h_{v,i,j}}$$

$$(B.20)$$

From this, the velocities at the new time step level are computed. The water level is then updated by:

$$\frac{\eta_{i,j}^{n+1} - \eta_{i,j}^{n}}{\Delta t} = -\frac{u_{i,j}^{n+1}h_{i,j}^{n} - u_{i-1,j}^{n+1}h_{i-1,j}^{n}}{x_{u,i,j} - x_{u,i-1,j}} - \frac{v_{i,j}^{n+1}h_{i,j} - v_{i,j-1}^{n+1}h_{i,j-1}^{n}}{y_{v,i,j} - y_{v,i,j-1}} \qquad (B.21)$$

which yields the wave energy at the new time level.

B.3   Advection-diffusion equations

The differential equations for the advection diffusion of sediment is solved with finite differences using the first order up-wind scheme discussed earlier with the water depths at the old time level and the corresponding velocities at the new time level. The horizontal x-advection is then given by:

$$
\left(\frac{\partial hCu^E}{\partial x}\right)_{i,j} = \frac{\left(h^n C^n u^{E,n+1}\right)_{i,j} - \left(h^n C^n u^{E,n+1}\right)_{i-1,j}}{x_{i,j} - x_{i-1,j}} \qquad , u_{i,j}^{E,n+1} > 0
$$

$$
\left(\frac{\partial hCu^E}{\partial x}\right)_{i,j} = \frac{\left(h^n C^n u^{E,n+1}\right)_{i+1,j} - \left(h^n C^n u^{E,n+1}\right)_{i,j}}{x_{i+1,j} - x_{i,j}} \qquad , u_{i,j}^{E,n+1} < 0 \qquad (2.22)
$$

A similar expression for the horizontal advection in the y-direction:

$$
\left(\frac{\partial hCv^E}{\partial y}\right)_{i,j} = \frac{\left(h^n C^n v^{E,n+1}\right)_{i,j} - \left(h^n C^n v^{E,n+1}\right)_{i-1,j}}{y_{i,j} - y_{i,j-1}} \qquad , v_{i,j}^{E,n+1} > 0
$$

$$
\left(\frac{\partial hCv^E}{\partial y}\right)_{i,j} = \frac{\left(h^n C^n v^{E,n+1}\right)_{i,j+1} - \left(h^n C^n v^{E,n+1}\right)_{i,j}}{y_{i,j+1} - y_{i,j}} \qquad , v_{i,j}^{E,n+1} < 0 \qquad (2.23)
$$

The horizontal diffusion is evaluated at the old time level n and approximated by:

$$\left(\frac{\partial}{\partial x}\left(D_H h \frac{\partial C}{\partial x} \right) \right)_{i,j} = \frac{\left(D_H h C_{\partial x} \right)_{i+1,j} - \left(D_H h C_{\partial x} \right)_{i,j}}{x_{i+1,j} - x_{i,j}} \tag{2.24}$$

Where the cross-shore gradient in the sediment concentration is given by:

$$C_{\partial x} = \left(\frac{\partial C}{\partial x} \right)_{i,j} = \frac{C_{i+1,j} - C_{i,j}}{x_{i+1,j} - x_{i,j}} \tag{2.25}$$

And similarly for the y-direction:

$$\left(\frac{\partial}{\partial y}\left(D_H h \frac{\partial C}{\partial y} \right) \right)_{i,j} = \frac{\left(D_H h C_{\partial y} \right)_{i,j+1} - \left(D_H h C_{\partial y} \right)_{i,j}}{y_{i,j+1} - y_{i,j}} \qquad , v_{i,j}^E < 0 \tag{2.26}$$

Where the along-shore gradient in the sediment concentration, C_y, is given by:

$$\left(\frac{\partial C}{\partial y} \right)_{i,j} = \frac{C_{i,j+1} - C_{i,j}}{y_{i,j+1} - y_{i,j}} \tag{2.27}$$

The time up-date of the sediment concentration is then given by:

$$\frac{h_{i,j}^{n+1} C_{i,j}^{n+1} - h_{i,j}^n C_{i,j}^n}{\Delta t} + \left[\frac{\partial h C u^E}{\partial x} \right]_{i,j}^n + \left[\frac{\partial h C v^E}{\partial y} \right]_{i,j}^n +$$
$$+ \left[\frac{\partial}{\partial x}\left[D_h h \frac{\partial C}{\partial x} \right] \right]_{i,j}^n + \left[\frac{\partial}{\partial y}\left[D_h h \frac{\partial C}{\partial y} \right] \right]_{i,j}^n = \left[\frac{h C_{eq} - h C}{T_s} \right]_{i,j}^n \tag{2.28}$$

B.4 Bed update

The bed-update is then approximated by:

$$\frac{z_{b,i,j}^{n+1} - z_{b,i,j}^n}{\Delta t} + \frac{f_{mor}}{(1-p)}\left[\frac{S_{x,i,j}^n - S_{x,i-1,j}^n}{\Delta x} + \frac{S_{y,i,j}^n - S_{y,i,j-1}^n}{\Delta y} \right] = 0 \tag{2.29}$$

Where f_{mor} represents a morphological factor to speed up the bed evolution (see e.g. Roelvink, 2006).

C   Multiple sediment fractions – proof of concept

To test the implementation of the sediment class formulation a comparison is made with observations of pre-and post hurricane Ivan cross-barrier island profiles (see Figure 6.1) at Beasly Park, Florida, USA (Wang and Horwitz, 2007).

The hurricane Ivan impact is simulated with a constant surge level of *1.8* m present for *10* hours at which time the offshore incident significant wave height is kept at *10* m with a mean wave period of *12* s. The sediment class distribution used in the calculations discriminates between sand located within the frontal dune (class 1) and sand located on and behind the barrier island (class 2) (see upper panel in). The sand on the barrier island is mostly vegetated which mitigates the erosion. Hence this sand has been given a mobility restriction that makes it more difficult to pick-up by means of a reduction factor of *0.25* on the equilibrium concentration. Grain sizes for both sand composites are the same with a D_{50} of 0.0035 m and a D_{90} of 0.005 mm. The initial sediment class distribution is presented in the top panel of Figure 1, where an intensity of 1 corresponds to sediment class one only and -1 to the presence of sediment class 2 only.

The bed-elevation and sediment class distribution after 10 hours are shown in the lower panel of . The calculated bed-level is similar to the observations although differences are apparent. These differences can be related to the fact that the hurricane impact is simply modeled (i.e. constant conditions) and the fact that the post-survey was performed approximately 10 months after the hurricane had past. Still the overall evolution is consistent with the observations. The calculated changes in the sediment classes are also consistent with the observations of Wang and Horwitz (2007) based on a number of cores showing that the intersection of the new washover with the pre-hurricane sediment occurs approximately at the original bed level.
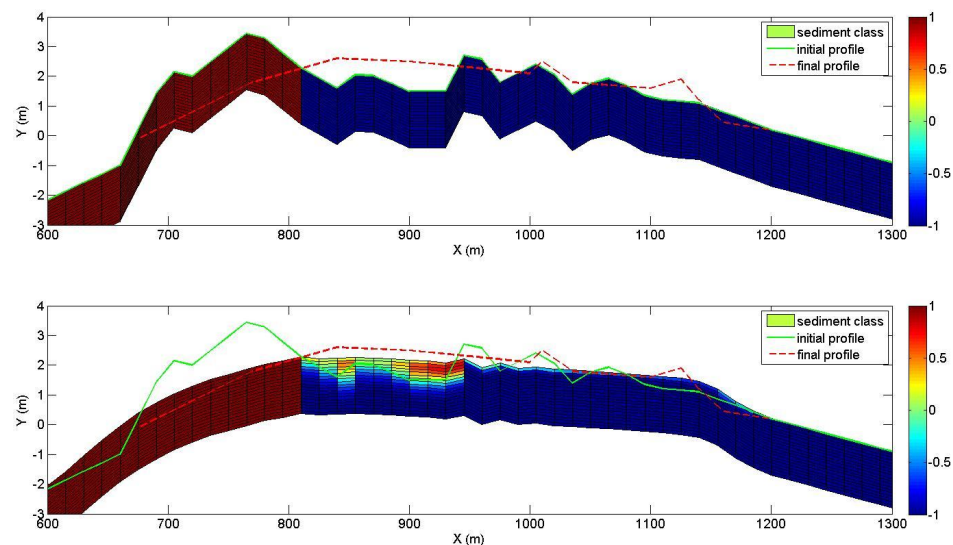

Figure 6.1 Top panel: Initial pre-hurricane bed elevation (green line) and sediment class distribution. A value of 1 corresponds to 100% of sediment class 1, a value of 0 to 50% of

class 1 and 50 % of class 2, and a value of -1 corresponds to 100 % of sediment class two. Post-hurricane bed elevation (dashed red line) given as a reference. Bottom panel: Calculated bed-evolution (corresponding to the position of the top layer) and corresponding sediment class distribution showing the thickness of the wash-over layer located behind the initial dune. Pre- (green line) and post-hurricane (red dashed line) bed elevation given as a reference