XBeach Model Description and Manual

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

Report

June 2009

Z4175

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

Report

June 200

Contents

Abstract

The XBeach program contains a number of Fortran 90/95 routines for short wave propagation, nonstationary shallow water equations, sediment transport and continuity equations that can be coupled in various ways and are designed to cope with extreme conditions such as encountered during hurricanes. 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 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 short wave propagation model contains 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 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. Full wave-current interaction in the short wave propagation is included. Roelvink (1993) wave dissipation model is implemented for use in the nonstationary 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). The numerical scheme was updated, 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. The resulting scheme has been verified with the well-known Carrier and Greenspan (1958) test.

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.

The model has been validated against a number of analytical and laboratory tests, both hydrodynamic and morphodynamic.

1 Introduction

This report has been prepared as part of the project 'Modeling of Hurricane Impacts', contract no. N62558-06-C-2006, which was granted by the US Army Corps of Engineers, Engineer Research and Development Center (ERDC), European Research Office and administered by FISC SIGONELLA, NAVAL REGIONAL CONTRACTING DET LONDON, SHORE/FLEET TEAM.

The project is being carried out by Prof. Dano Roelvink of UNESCO-IHE (Principal Investigator), Dr. Ad Reniers of Univ. of Miami, Jaap van Thiel de Vries and Robert McCall of Delft University of Technology and Dr. Ap van Dongeren and Jamie Lescinski of Delatares (formerly WL | Delft Hydraulics).

1.1 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; first order accuracy is accepted since there is a need for small space steps and time steps anyway, to represent the strong gradients in space and time in the nearshore and swash zone. Because of the many shock-like features in both hydrodynamics and morphodynamics we choose upwind schematizations, where this is acceptable without too much damping, as a means to avoid numerical oscillations which can be deadly in shallow areas.

1.2 Context

The XBeach model can be used as stand-alone model for small-scale (project-scale) coastal applications, but will also be used within the Morphos model system, 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.

1.3 Functionalities

The code has the following functionalities:

- Depth-averaged shallow water equations including time-varying wave forcing terms; combination of sub- and supercritical flows;
- Time-varying wave action balance including refraction, shoaling, current refraction and wave breaking;

- Roller model, including breaker delay
- Wave amplitude effects on wave celerity;
- Depth-averaged advection-diffusion equation to solve suspended transport;
- Bed updating algorithm including possibility of avalanching;
- 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. The resulting scheme has been tested against the well-known Carrier and Greenspan test.
- 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)
- 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)
- Soulsby – Van Rijn transport formulations, cf Reniers et al (2004).
- Multiple sediment fractions and bed layer bookkeeping.
- Automatic time step based on Courant criterion, with output at fixed or user-defined time intervals.
- Avalanching mechanism, with separate criteria for critical slope at wet or dry points.
- MPI (Message Passing Interface) implementation with automatic domain decomposition for parallel (multi-processor) computing

1.4 Outline of the report

In the following report we will detail the model development, activities and results. Chapter 2 provides a description of the XBeach structure, as well as an overview of siginificant attributes of the program. Chapter 3 contains an overview of all formulations used and their numerical schematization. Chapter 4 provides instructions on how to run the model, with a detailed input description. Chapter 5 offers a small discussion of how it is intended to distribute and maintain XBeach.

2 Descripton of program structure

2.1 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 2.1.

Table 2.1. 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
readtide(s,par)	Read tide levels
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
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
flow_timestep (s,par)	Carries out one flow timestep
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	

2.2 Implementation of parallel computing using MPI

To be done.

3 Model formulations

3.1 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 3.1 . This coordinate system is defined relative to world coordinates (xw,yw) through the origin (xori,yori) and the orientation alfa, defined counter-clockwise w.r.t. the xw-axis (East). The grid size in x- and y-direction may be variable but the grid must be rectilinear.

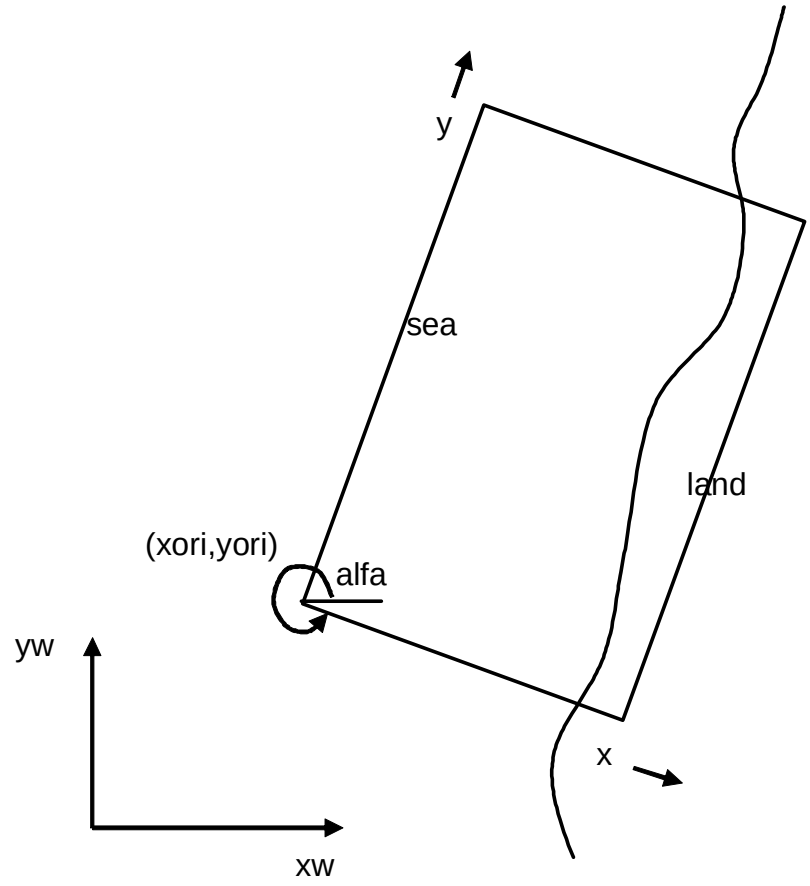

Figure 3.1 Coordinate system

3.2 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 3.2.

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.

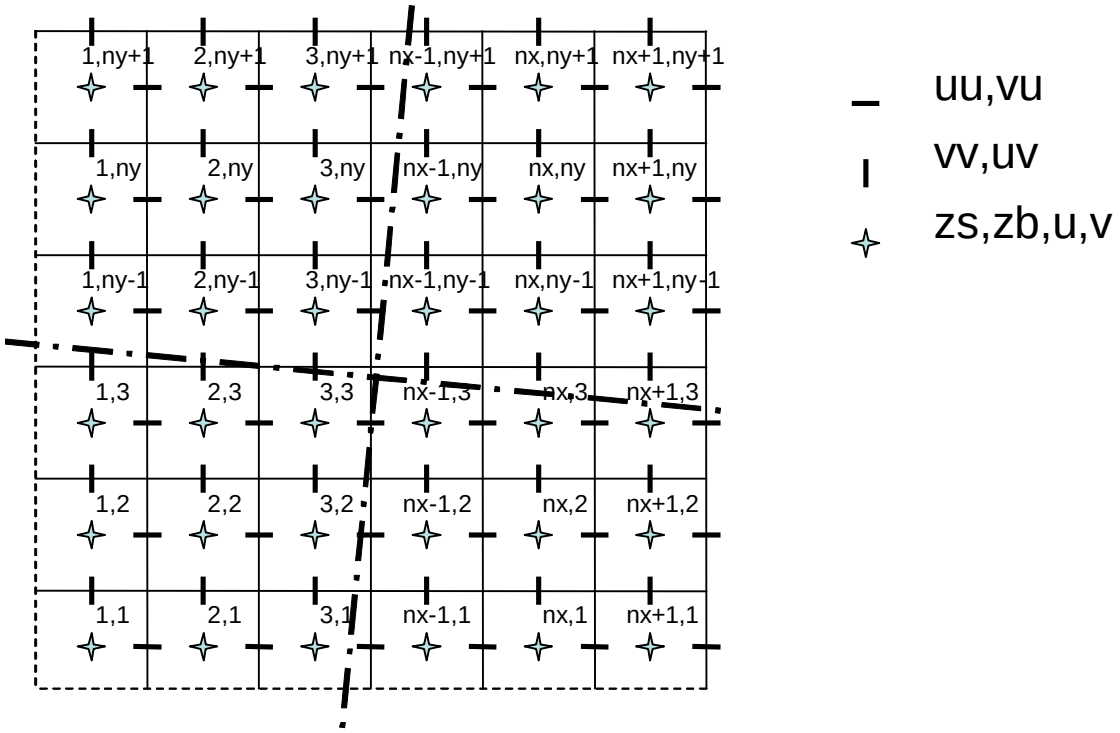

Figure 3.2 Staggered grid

3.3 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 HISWA model, the directional distribution of the action density is taken into account whereas the frequency spectrum is represented by a single mean frequency. The wave action balance is then given by:

$$\frac{\partial A}{\partial t}+\frac{\partial c_x A}{\partial x}+\frac{\partial c_y A}{\partial y}+\frac{\partial c_\theta A}{\partial \theta}=-\frac{D}{\sigma} \tag{3.1}$$

With D the wave dissipation due to wave breaking (wave dissipation due to bottom friction is relatively smal) and with the wave action:

$$A(x,y,\theta) = \frac{S_w(x,y,\theta)}{\sigma(x,y)} \tag{3.2}$$

Where S_w represents the wave energy in each directional bin and σ the intrinsic wave frequency. The wave action propagation speeds in x- and y-direction are given by:

$$\begin{aligned} c_x(x,y,\theta) &= c_g(x,y)\bullet\cos(\theta) + u(x,y) \\ c_y(x,y,\theta) &= c_g(x,y)\bullet\sin(\theta) + v(x,y) \end{aligned} \tag{3.3}$$

Where θ represents the angle of incidence with respect to the x-axis. The propagation speed in θ-space is obtained from:

$$\begin{aligned} c_\theta(x,y,\theta) = \frac{\sigma}{\sinh 2kh}\left(\frac{\partial h}{\partial x}\sin\theta - \frac{\partial h}{\partial y}\cos\theta\right) + \cos\theta\left(\sin\theta\frac{\partial u}{\partial x} - \cos\theta\frac{\partial u}{\partial y}\right) + \\ + \sin\theta\left(\sin\theta\frac{\partial v}{\partial x} - \cos\theta\frac{\partial v}{\partial y}\right) \end{aligned} \tag{3.4}$$

Taking into account bottom refraction (first term on the RHS) and current refraction (last two terms on the RHS). The wave number k is obtained from the eikonal equations:

$$\begin{aligned} \frac{\partial k_x}{\partial t} + \frac{\partial \omega}{\partial x} &= 0 \\ \frac{\partial k_y}{\partial t} + \frac{\partial \omega}{\partial y} &= 0 \end{aligned} \tag{3.5}$$

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

$$k = \sqrt{k_x^2 + k_y^2} \tag{3.6}$$

The absolute radial frequency is given by:

$$\omega = \sigma + \vec{k}\bullet\vec{u} \tag{3.7}$$

And the intrinsic frequency is obtained from the linear dispersion relation:

$$\sigma = \sqrt{gk\tanh kh} \tag{3.8}$$

The group velocity is obtained from linear wave theory:

$$c_g = nc = \left(\frac{1}{2} + \frac{kh}{\sinh 2kh} \right) \frac{\sigma}{k} \tag{3.9}$$

This concludes the advection of wave action. The wave energy dissipation due to wave breaking is modelled according to Roelvink (1993) for instationary runs or Baldock et al. [1998] for stationary runs.

The formulations for the total wave dissipation according to Roelvink (1993) are:

$$D = 2\alpha f_{rep} E_w Q_b \tag{3.10}$$

With $\alpha = O(1)$ and f_{rep} representing a representative intrinsic frequency.

Here

$$E_w = \frac{1}{8} \rho g H_{rms}^2 = \int S_w(x, y, \theta) d\theta \tag{3.11}$$

Where H_{rms} is the root mean squared wave height.

The fraction of breaking waves is given by:

$$Q_b = \min \left(1 - e^{-\left(\frac{H_{ms}}{\gamma h} \right)^n}, \ 1 \right) \tag{3.12}$$

For Baldock et al. (1998) they are as follows:

$$\bar{D} = \frac{1}{4} \alpha Q_b \rho g f_{rep} \left(H_b^2 + H_{rms}^2 \right) \tag{3.13}$$

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

$$Q_b = \exp \left[-\left(\frac{H_b^2}{H_{rms}^2} \right) \right] \tag{3.14}$$

Where the breaking wave height is:

$$H_b = \frac{0.88}{k} \tanh \left[\frac{\gamma kh}{0.88} \right] \tag{3.15}$$

And γ is a calibration parameter.

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

$$D(x, y, \theta) = \frac{S_w(x, y, \theta)}{E_w(x, y)} \bar{D} \tag{3.16}$$

This closes the set of equations for the wave action balance. Given the spatial distribution of the wave action and therefore wave energy the wave forcing can be calculated utilizing the radiation stress tensor:

$$F_x = -\left(\frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{xy}}{\partial y} \right)$$
$$F_y = -\left(\frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} \right) \tag{3.17}$$

And:

$$S_{xx} = \int \left(\frac{c_g}{c}\left(1+\cos^2\theta\right) - \frac{1}{2} \right) S_w d\theta$$
$$S_{xy} = S_{yx} = \int \sin\theta\cos\theta \left(\frac{c_g}{c} S_w \right) d\theta \tag{3.18}$$
$$S_{yy} = \int \left(\frac{c_g}{c}\left(1+\sin^2\theta\right) - \frac{1}{2} \right) S_w d\theta$$

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:

$$\frac{\partial c_x^n A^n}{\partial x}(\mathrm{i,j,k}) = \frac{c^n_{x,i,j,k} A^n_{i,j,k} - c^n_{x,i-1,j,k} A^n_{i-1,j,k}}{x_{\mathrm{i,j}} - x_{\mathrm{i\text{-}1,j}}} \quad , c^n_{x,i,j,k} > 0$$
$$\frac{\partial c_x^n A^n}{\partial x}(\mathrm{i,j,k}) = \frac{c^n_{x,i+1,j,k} A^n_{i+1,j,k} - c^n_{x,i,j,k} A^n_{i,j,k}}{x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}} \quad , c^n_{x,i,j,k} < 0 \tag{3.19}$$

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

$$\frac{\partial c_x^n A^n}{\partial x}(\mathrm{i,j,k}) = \frac{c^n_{x,i+1,j,k} A^n_{i+1,j,k} - c^n_{x,i-1,j,k} A^n_{i-1,j,k}}{x_{\mathrm{i+1,j}} - x_{\mathrm{i\text{-}1,j}}} -$$
$$\frac{\Delta t}{2} \frac{\left[c^n_{x,i+1,j,k}\right]^2 A^n_{i+1,j,k} - 2\left[c^n_{x,i,j,k}\right]^2 A^n_{i,j,k} + \left[c^n_{x,i-1,j,k}\right]^2 A^n_{i-1,j,k}}{\left(x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}\right)\left(x_{\mathrm{i,j}} - x_{\mathrm{i\text{-}1,j}}\right)} \tag{3.20}$$

In y-direction, the upwind discretization is

$$\frac{\partial c_y^n A^n}{\partial y}(\mathrm{i,j,k}) = \frac{c^n_{y,i,j,k} A^n_{i,j,k} - c^n_{y,i,j-1,k} A^n_{i,j-1,k}}{y_{\mathrm{i,j}} - y_{\mathrm{i,j\text{-}1}}} \quad , c^n_{y,i,j,k} > 0$$
$$\frac{\partial c_y^n A^n}{\partial y}(\mathrm{i,j,k}) = \frac{c^n_{y,i,j+1,k} A^n_{i,j+1,k} - c^n_{y,i,j,k} A^n_{i,j,k}}{y_{\mathrm{i,j+1}} - y_{\mathrm{i,j}}} \quad , c^n_{y,i,j,k} < 0 \tag{3.21}$$

In the Lax-Wendroff discretization:

$$\frac{\partial c_y^n A^n}{\partial y}(\mathrm{i,j,k}) = \frac{c^n_{y,i,j+1,k} A^n_{i,j+1,k} - c^n_{y,i,j-1,k} A^n_{i,j-1,k}}{y_{\mathrm{i,j+1}} - y_{\mathrm{i,j\text{-}1}}} - \frac{\Delta t}{2} \frac{\left[c^n_{y,i,j+1,k}\right]^2 A^n_{i,j+1,k} - 2\left[c^n_{y,i,j,k}\right]^2 A^n_{i,j,k} + \left[c^n_{y,i,j-1,k}\right]^2 A^n_{i,j-1,k}}{\left(y_{\mathrm{i,j+1}} - y_{\mathrm{i,j}}\right)\left(y_{\mathrm{i,j}} - y_{\mathrm{i,j\text{-}1}}\right)} \tag{3.22}$$

In theta-space an upwind scheme is used only:

$$\begin{aligned}\frac{\partial c_\theta^n A^n}{\partial \theta}(\mathrm{i,j,k}) &= \frac{c_{\theta,i,j,k}^n A_{i,j,k}^n - c_{\theta,i,j,k-1}^n A_{i,j,k-1}^n}{\theta_{\mathrm{i,j,k}} - \theta_{\mathrm{i,j,k\text{-}1}}} \quad , c_{\theta,i,j,k}^n > 0 \\ \frac{\partial c_\theta^n A^n}{\partial \theta}(\mathrm{i,j,k}) &= \frac{c_{\theta,i,j,k+1}^n A_{i,j,k+1}^n - c_{\theta,i,j,k}^n A_{i,j,k}^n}{\theta_{\mathrm{i,j,k+1}} - \theta_{\mathrm{i,j,k}}} \quad , c_{\theta,i,j,k}^n < 0\end{aligned} \tag{3.23}$$

Similar for the wave action balance:

$$\frac{A_{i,j,k}^{n+1} - A_{i,j,k}^n}{\Delta t} = -\frac{\partial c_x^n A^n}{\partial x}_{i,j,k} - \frac{\partial c_y^n A^n}{\partial y}_{i,j,k} - \frac{\partial c_\theta^n A^n}{\partial \theta}_{i,j,k} - \frac{D}{\sigma}_{i,j,k} \tag{3.24}$$

Which yields the wave energy at the new time level.

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. 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.

3.4 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:

$$\frac{\partial S_r}{\partial t} + \frac{\partial c_x S_r}{\partial x} + \frac{\partial c_y S_r}{\partial y} + \frac{\partial c_\theta S_r}{\partial \theta} = -D_r + D_w \tag{3.25}$$

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

$$c_x(x,y,\theta) = c(x,y)\bullet\cos(\theta) + u(x,y)$$
$$c_y(x,y,\theta) = c(x,y)\bullet\sin(\theta) + v(x,y) \tag{3.26}$$

Where θ represents the angle of incidence with respect to the x-axis. The propagation speed in θ -space is obtained from:

$$c_\theta(x,y,\theta) = \frac{\sigma}{\sinh 2kh}\left(\frac{\partial h}{\partial x}\sin\theta - \frac{\partial h}{\partial y}\cos\theta\right) + \\ + \cos\theta\left(\sin\theta\frac{\partial u}{\partial x} - \cos\theta\frac{\partial u}{\partial y}\right) + \sin\theta\left(\sin\theta\frac{\partial v}{\partial x} - \cos\theta\frac{\partial v}{\partial y}\right) \tag{3.27}$$

taking into account bottom refraction (first term on the RHS) and current refraction (last two terms on the RHS). Hence, we are assuming that the waves and rollers propagate in the same direction. The phase velocity is obtained from linear wave theory:

$$c = \frac{\sigma}{k} \tag{3.28}$$

Which concludes the advection of roller energy. The roller energy dissipation is given by (Deigaard, 1993):

$$(\bar{D}_r = c\tau_r) \tag{3.29}$$

With τ_r representing the shear stress induced by the roller at the surface, which is expressed by (Svendsen, 1984):

$$\tau_r = \frac{\rho g R}{L}\beta_r \tag{3.30}$$

Where R represents the roller area, L the roller length and β_r is the slope of the breaking wave (typically between 0.05 and 0.10). The roller area is related to the roller energy trough:

$$E_r = \frac{1}{2}\frac{\rho R c^2}{L} \tag{3.31}$$

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

$$D_r(x,y,\theta) = \frac{S_r(x,y,\theta)}{E_r(x,y)}\bar{D}_r \tag{3.32}$$

Similarly, the source term is obtained from the wave action/energy balance:

$$D_w(x,y,\theta) = \frac{S_w(x,y,\theta)}{E_w(x,y)}\bar{D} \tag{3.33}$$

This closes the set of equations for the roller energy balance. The roller also affects the wave forcing and has therefore to be included in the radiation stress terms:

$$\begin{aligned} S_{xx,r} &= \int \cos^2\theta S_r d\theta \\ S_{xy,r} &= S_{yx,r} = \int \sin\theta\cos\theta S_r d\theta \\ S_{yy,r} &= \int \sin^2\theta S_w d\theta \end{aligned} \tag{3.34}$$

These roller radiation stress contributions are added to the wave-induced radiation stresses. Similar to the solution of the wave action equations we use an up-wind schematisation to solve the roller energy balance.

3.5 Shallow water equations solver

The depth-averaged and shortwave-averaged shallow water equations read :

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - f\,v - \nu_h\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \frac{\tau_{sx}}{\rho h} - \frac{\tau_{bx}}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \tag{3.35}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + f\,u - \nu_h\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = +\frac{\tau_{sy}}{\rho h} - \frac{\tau_{by}}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \tag{3.36}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} + w = 0 \tag{3.37}$$

Here, h is the water depth, u, v are velocities in x and and y direction, τ_{bx}, τ_{by} are the bed shear stresses, τ_{sx}, τ_{sy} are the wind stresses g is the acceleration of gravity, η is the water level, F_x, F_y are the wave-induced stresses and w is the groundwater infiltration and exfiltration velocity (see section 3.9).

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.

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:

$$\frac{\partial \eta}{\partial x}(\mathrm{i,j}) = \frac{\eta_{\mathrm{i+1,j}} - \eta_{\mathrm{i,j}}}{x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}} \tag{3.38}$$

$$\frac{\partial \eta}{\partial y}(\mathrm{i,j}) = \frac{\eta_{\mathrm{i,j+1}} - \eta_{\mathrm{i,j}}}{x_{\mathrm{i,j+1}} - x_{\mathrm{i,j}}} \tag{3.39}$$

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:

$$\begin{aligned} v_{u,i,j} &= \frac{1}{4}(v_{i,j-1} + v_{i,j} + v_{i+1,j-1} + v_{i+1,j}) \\ u_{v,i,j} &= \frac{1}{4}(u_{i-1,j} + u_{i,j} + u_{i-1,j+1} + u_{i,j+1}) \end{aligned} \tag{3.40}$$

The water depth in each cell is computed as:

$$h_{i,j} = \eta_{i,j} - z_{b,i,j} \tag{3.41}$$

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:

$$\begin{aligned} h_{u,i,j} &= h_{i,j} && , u_{i,j} > u_{\min} \\ h_{u,i,j} &= h_{i+1,j} && , u_{i,j} < -u_{\min} \\ h_{u,i,j} &= \max(z_{s,i,j}, z_{s,i+1,j}) - \max(z_{b,i,j}, z_{b,i+1,j}) && , \left| u_{i,j} \right| < u_{\min} \end{aligned} \tag{3.42}$$

$$\begin{aligned} h_{v,i,j} &= h_{i,j} && , v_{i,j} > v_{\min} \\ h_{v,i,j} &= h_{i,j+1} && , v_{i,j} < -v_{\min} \\ h_{v,i,j} &= \max(z_{s,i,j}, z_{s,i,j+1}) - \max(z_{b,i,j}, z_{b,i,j+1}) && , \left| v_{i,j} \right| < v_{\min} \end{aligned} \tag{3.43}$$

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

$$h_{mu,i,j} = \frac{1}{2}(h_{i,j} + h_{i+1,j}) \tag{3.44}$$

$$h_{mv,i,j} = \frac{1}{2}(h_{i,j} + h_{i,j+1}), \tag{3.45}$$

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

$$\begin{aligned} u\frac{\partial u^n}{\partial x_{i,j}} &= \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i-1,j}u_{i-1,j}}{h_{mu,i,j}}\frac{u^n_{i,j} - u^n_{i-1,j}}{x^n_{i,j} - x^n_{i-1,j}} && , u^n_{i,j} > 0 \\ u\frac{\partial u^n}{\partial x_{i,j}} &= \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i+1,j}u_{i+1,j}}{h_{mu,i,j}}\frac{u^n_{i+1,j} - u^n_{i,j}}{x^n_{i+1,j} - x^n_{i,j}} && , u^n_{i,j} < 0 \end{aligned} \tag{3.46}$$

$$
\begin{aligned}
v\frac{\partial u}{\partial y}^{n}_{i,j} &= \frac{1}{2}\frac{h_{u,i,j}v^{n}_{u,i,j}+h_{u,i,j-1}v^{n}_{u,i,j-1}}{h_{mu,i,j}}\frac{u^{n}_{i,j}-u^{n}_{i,j-1}}{y^{n}_{i,j}-y^{n}_{i,j-1}} \quad , v^{n}_{u,i,j}>0\\
v\frac{\partial u}{\partial y}^{n}_{i,j} &= \frac{1}{2}\frac{h_{u,i,j}v^{n}_{u,i,j}+h_{u,i,j+1}v^{n}_{u,i,j+1}}{h_{mu,i,j}}\frac{u^{n}_{i,j+1}-u^{n}_{i,j}}{y^{n}_{i,j+1}-y^{n}_{i,j}} \quad , v^{n}_{u,i,j}>0
\end{aligned}
\tag{3.47}
$$

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

$$
\begin{aligned}
v\frac{\partial v}{\partial y}^{n}_{i,j} &= \frac{1}{2}\frac{h^{n}_{v,i,j}v^{n}_{i,j}+h^{n}_{v,i,j-1}v^{n}_{i,j-1}}{h^{n}_{mv,i,j}}\frac{v^{n}_{i,j}-v^{n}_{i,j-1}}{y^{n}_{i,j}-y^{n}_{i,j-1}} \qquad , v^{n}_{i,j}>0\\
v\frac{\partial v}{\partial y}^{n}_{i,j} &= \frac{1}{2}\frac{h^{n}_{v,i,j}v^{n}_{i,j}+h^{n}_{v,i,j+1}v^{n}_{i,j+1}}{h^{n}_{mv,i,j}}\frac{v^{n}_{i,j+1}-v^{n}_{i,j}}{y^{n}_{i,j+1}-y^{n}_{i,j}} \qquad , v^{n}_{i,j}<0
\end{aligned}
\tag{3.48}
$$

$$
\begin{aligned}
u\frac{\partial v}{\partial x}^{n}_{i,j} &= \frac{1}{2}\frac{h_{v,i,j}u^{n}_{v,i,j}+h_{v,i-1,j}u^{n}_{v,i-1,j}}{h_{mv,i,j}}\frac{v^{n}_{i,j}-v^{n}_{i-1,j}}{x^{n}_{i,j}-x^{n}_{i-1,j}}, \quad u^{n}_{v,i,j}>0\\
u\frac{\partial v}{\partial x}^{n}_{i,j} &= \frac{1}{2}\frac{h_{v,i,j}u^{n}_{v,i,j}+h_{v,i+1,j}u^{n}_{v,i+1,j}}{h_{mv,i,j}}\frac{v^{n}_{i+1,j}-v^{n}_{i,j}}{x^{n}_{i+1,j}-x^{n}_{i,j}}, \quad u^{n}_{v,i,j}>0
\end{aligned}
\tag{3.49}
$$

The momentum equation is discretized as follows:

$$
\frac{u^{n+1}_{i,j}-u^{n}_{i,j}}{\Delta t} = -u\frac{\partial u}{\partial x}^{n}_{i,j} - v\frac{\partial u}{\partial y}^{n}_{i,j} - c_f\frac{u^{n}_{i,j}}{h^{n}_{u,i,j}}\sqrt{1.16u^{2}_{rms}+{u^{n}_{i,j}}^{2}+{v^{n}_{u,i,j}}^{2}} - g\frac{\eta^{n}_{i+1,j}-\eta^{n}_{i,j}}{x_{i+1,j}-x_{i,j}}+\frac{F_{x,i,j}}{\rho h_{u,i,j}} \tag{3.50}
$$

$$
\frac{v^{n+1}_{i,j}-v^{n}_{i,j}}{\Delta t} = -v\frac{\partial v}{\partial y}^{n}_{i,j} - u\frac{\partial v}{\partial x}^{n}_{i,j} - c_f\frac{v^{n}_{i,j}}{h^{n}_{v,i,j}}\sqrt{1.16u^{2}_{rms}+{u^{n}_{v,i,j}}^{2}+{v^{n}_{i,j}}^{2}} - g\frac{\eta_{i,j+1}-\eta_{i,j}}{y_{i,j+1}-y_{i,j}}+\frac{F_{y,i,j}}{\rho h_{v,i,j}} \tag{3.51}
$$

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

$$
\frac{\eta^{n+1}_{i,j}-\eta^{n}_{i,j}}{\Delta t} = -\frac{u^{n+1}_{i,j}h^{n}_{i,j}-u^{n+1}_{i-1,j}h^{n}_{i-1,j}}{x_{u,i,j}-x_{u,i-1,j}} - \frac{v^{n+1}_{i,j}h_{i,j}-v^{n+1}_{i,j-1}h^{n}_{i,j-1}}{y_{v,i,j}-y_{v,i,j-1}} \tag{3.52}
$$

Generalized Lagrangian Mean formulation

To account for the wave induced mass-flux and the subsequent (return) flow the shallow water equations are cast into a Generalized Lagrangian Mean (GLM) formulation (Walstra et al, 2000; Andrews and McIntyre, 1978). To that end the Eulerian shallow water velocity u^E is replaced with its lagrangian equivalent u^L:

$$u^L = u^E + u^S \quad and \quad v^L = v^E + v^S \tag{3.53}$$

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

$$u^S = \frac{E_w \cos\theta}{\rho hc} \quad and \quad v^S = \frac{E_w \sin\theta}{\rho hc} \tag{3.54}$$

Where the wave-group varying short wave energy and direction are obtained from the wave-action balance. The resulting GLM-momentum equations are given by:

$$\begin{aligned} \frac{\partial u^L}{\partial t} + u^L \frac{\partial u^L}{\partial x} + v^L \frac{\partial u^L}{\partial y} &= -\frac{\tau^E_{bx}}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \\ \frac{\partial v^L}{\partial t} + u^L \frac{\partial v^L}{\partial x} + v^L \frac{\partial v^L}{\partial y} &= -\frac{\tau^E_{by}}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \end{aligned} \tag{3.55}$$

For the x- and y-direction respectively. This operation shows that the GLM equations for the depth-averaged flow are very similar to the previously described Eulerian formulation, with the exception of the bottom shear stress terms that are calculated with the Eulerian velocities as experienced by the bed:

$$u^E = u^L - u^S \quad and \quad v^E = v^L - v^S \tag{3.56}$$

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

3.6 Sediment transport

Advection-diffusion scheme

The sediment transport is modeled with a depth-averaged advection diffusion equation [Gallapatti, 1983]:

$$\frac{\partial hC}{\partial t}+\frac{\partial hCu^E}{\partial x}+\frac{\partial hCv^E}{\partial y}+\frac{\partial}{\partial x}\left[D_h h\frac{\partial C}{\partial x}\right]+\frac{\partial}{\partial y}\left[D_h h\frac{\partial C}{\partial y}\right]=\frac{hC_{eq}-hC}{T_s} \quad (3.57)$$

Where C represents the depth-averaged sediment concentration which varies on the infragravity time scale. 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, sediment fall velocity w_s and a sediment transport depth factor f_{Ts}:

$$T_s=\max\left(f_{Ts}\frac{h}{w_s},0.2\right)s \quad (3.58)$$

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 and the equilibrium concentration, C_{eq}, thus representing the source term in the sediment transport equation.

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:

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

$$\left(\frac{\partial hCu^E}{\partial x}\right)_{i,j}=\frac{\left(h^nC^nu^{E,n+1}\right)_{i+1,j}-\left(h^nC^nu^{E,n+1}\right)_{i,j}}{x_{i+1,j}-x_{i,j}} \quad ,u_{i,j}^{E,n+1}<0 \quad (3.59)$$

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

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

$$\left(\frac{\partial hCv^E}{\partial y}\right)_{i,j}=\frac{\left(h^nC^nv^{E,n+1}\right)_{i,j+1}-\left(h^nC^nv^{E,n+1}\right)_{i,j}}{y_{i,j+1}-y_{i,j}} \quad ,v_{i,j}^{E,n+1}<0 \quad (3.60)$$

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

$$\left(\frac{\partial}{\partial x}\left(D_H h\frac{\partial C}{\partial x}\right)\right)_{i,j} = \frac{\left(D_H h C_{\partial x}\right)_{i+1,j} - \left(D_H h C_{\partial x}\right)_{i,j}}{x_{i+1,j} - x_{i,j}} \tag{3.61}$$

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

$$C_{\partial x} = \left(\frac{\partial C}{\partial x}\right)_{i,j} = \frac{C_{i+1,j} - C_{i,j}}{x_{i+1,j} - x_{i,j}} \tag{3.62}$$

And similarly for the y-direction:

$$\left(\frac{\partial}{\partial y}\left(D_H h\frac{\partial C}{\partial y}\right)\right)_{i,j} = \frac{\left(D_H h C_{\partial y}\right)_{i,j+1} - \left(D_H h C_{\partial y}\right)_{i,j}}{y_{i,j+1} - y_{i,j}} \qquad , v^E_{i,j} < 0 \tag{3.63}$$

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

$$\left(\frac{\partial C}{\partial y}\right)_{i,j} = \frac{C_{i,j+1} - C_{i,j}}{y_{i,j+1} - y_{i,j}} \tag{3.64}$$

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

$$\begin{aligned}\frac{h^{n+1}_{i,j}C^{n+1}_{i,j} - h^n_{i,j}C^n_{i,j}}{\Delta t} + \left[\frac{\partial hCu^E}{\partial x}\right]^n_{i,j} + \left[\frac{\partial hCv^E}{\partial y}\right]^n_{i,j} + \\ + \left[\frac{\partial}{\partial x}\left[D_h h\frac{\partial C}{\partial x}\right]\right]^n_{i,j} + \left[\frac{\partial}{\partial y}\left[D_h h\frac{\partial C}{\partial y}\right]\right]^n_{i,j} = \left[\frac{hC_{eq} - hC}{T_s}\right]^n_{i,j}\end{aligned} \tag{3.65}$$

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

$$(1-p)\frac{\partial z_b}{\partial t} + \frac{\partial S_x}{\partial x} + \frac{\partial S_y}{\partial y} = 0 \tag{3.66}$$

Where p is the porosity and S_x and S_y represent the sediment transport rates in *x*- and *y*-direction respectively, given by:

$$S^n_{x,i,j} = \left[\frac{\partial hCu^E}{\partial x}\right]^n_{i,j} + \left[\frac{\partial}{\partial x}\left[D_h h\frac{\partial C}{\partial x}\right]\right]^n_{i,j} - \left[f_{slope}Cu^E h\frac{\partial z_b}{\partial x}\right]^n_{i,j} \tag{3.67}$$

and

$$S^n_{y,i,j} = \left[\frac{\partial hCv^E}{\partial y}\right]^n_{i,j} + \left[\frac{\partial}{\partial y}\left[D_h h\frac{\partial C}{\partial y}\right]\right]^n_{i,j} - \left[f_{slope}Cv^E h\frac{\partial z_b}{\partial y}\right]^n_{i,j} \tag{3.68}$$

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

The bed-update is then approximated by:

$$\frac{z_{b,i,j}^{n+1} - z_{b,i,j}^{n}}{\Delta t} + \frac{f_{mor}}{(1-p)}\left[\frac{S_{x,i,j}^{n} - S_{x,i-1,j}^{n}}{\Delta x} + \frac{S_{y,i,j}^{n} - S_{y,i,j-1}^{n}}{\Delta y}\right] = 0 \qquad (3.69)$$

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

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 :

$$C_{eq} = \frac{A_{sb} + A_{ss}}{h}\left(\left(\left(u^{E}\right)^{2} + \left(v^{E}\right)^{2} + 0.018\frac{u_{rms}^{2}}{C_{d}}\right)^{0.5} - u_{cr}\right)^{2.4} \qquad (3.70)$$

Where sediment is stirred by the Eulerian mean and infragravity velocity in combination with the near bed short wave orbital velocity obtained from the wave-group varying wave energy as

$$\mathrm{u_{rms}} = \frac{\pi H_{rms}}{T_{rep}\sqrt{2}\sinh(kh + \delta H_{rms})} \qquad (3.71)$$

The combined mean/infragravity and orbital velocity have to exceed a threshold value, ucr, before sediment is set in motion. The drag coefficient, C_d , is due to flow velocity only (ignoring short wave effects).. 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 (θ_{sf} = 0.8 – 1.0):

$$u_{flow,stirring}^{2} = \min\left(\left(u^{E}\right)^{2} + \left(v^{E}\right)^{2}, \theta_{sf}\frac{gD_{50}\Delta}{c_{f}}\right) \qquad (3.72)$$

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).

3.7 Bottom updating

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:

$$\left|\frac{\partial z_b}{\partial x}\right| > m_{cr} \qquad (3.73)$$

Where the estimated bed slope is given by:

$$\frac{\partial z_b}{\partial x} = \frac{z_{b,i+1,j} - z_{b,i,j}}{\Delta x} \qquad (3.74)$$

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

$$\begin{aligned} \Delta z_b &= \min\left(\left(\left|\frac{\partial z_b}{\partial x}\right| - m_{cr} \right)\Delta x, \;\; 0.05\Delta t \right) \quad , \frac{\partial z_b}{\partial x} > 0 \\ \Delta z_b &= \max\left(-\left(\left|\frac{\partial z_b}{\partial x}\right| - m_{cr} \right)\Delta x, -0.05\Delta t \right) \quad , \frac{\partial z_b}{\partial x} < 0 \end{aligned} \qquad (3.75)$$

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:

$$\begin{aligned} z_{b,i,j}^{n+1} &= z_{b,i,j}^{n} + \Delta z_{b,i,j} \\ z_{b,i+1,j}^{n+1} &= z_{b,i+1,j}^{n} - \Delta z_{b,i,j} \end{aligned} \qquad (3.76)$$

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

$$\begin{aligned} z_{s,i,j}^{n+1} &= z_{s,i,j}^{n} + \Delta z_{b,i,j} \\ z_{s,i+1,j}^{n+1} &= z_{s,i+1,j}^{n} - \Delta z_{b,i,j} \end{aligned} \qquad (3.77)$$

Similar expressions are used for the subsequent avalanching in the y-direction.

3.8 Multiple sediment fractions

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 xxxxxxxx.

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:

$$c_{eq}(i) = frc(i,1)c_{eq}^{*}(i) \tag{3.78}$$

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. (3.57)) 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:

$$\Delta z_i = \frac{\Delta t}{1-n_p}\left[\frac{\partial S_{i,x}}{\partial x} + \frac{\partial S_{i,y}}{\partial y}\right] \tag{3.79}$$

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

$$frc^{n+1}(i,1) = \frac{\Delta z_i}{D_z} + \frac{D_z - \Delta z}{D_z}\, frc^{n}(i,1) \tag{3.80}$$

Given dz <= Dz, else:

$$frc^{n+1}(i,1) = \frac{\Delta z_i}{\Delta z} \tag{3.81}$$

And similarly for erosion:

$$frc^{n+1}(i,1) = \frac{\Delta z_i + D_z}{D_z}\, frc^{n}(i,1) - \frac{\Delta z}{D_z}\, frc^{n}(i,2) \tag{3.82}$$

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:

$$frc^{n+1}(i,j) = \frac{D_z + \Delta z}{D_z} frc^{n}(i,j) - \frac{\Delta z}{D_z} frc^{n}(i,j+1) \qquad (3.83)$$

For erosion and:

$$frc^{n+1}(i,j) = \frac{D_z - \Delta z}{D_z} frc^{n}(i,j) + \frac{\Delta z}{D_z} frc^{n}(i,j-1) \qquad (3.84)$$

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.

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 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 Figure 3.3). 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 Figure 3.3. 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.

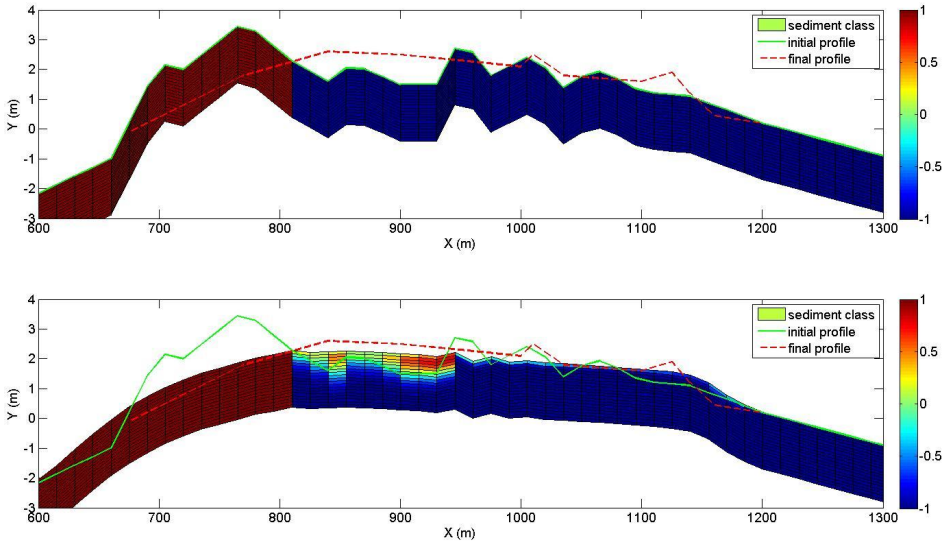

Figure 3.3 Top panel: Initial pre-hurricane bed elevation (green line) and sediment class distribution. A value of 1 corresponds to 100% of sediment class 1, a value of 0 to 50% of class 1 and 50 % of class 2, and a value of -1 corresponds to 100 % of sediment class two. Post-hurricane bed elevation (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.

3.9 Groundwater flow

3.9.1 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.

3.9.2 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}:

$$\left[p_{gw}\right]_{i,j}^{n} = \left[\eta_{gw}\right]_{i,j}^{n-1} \qquad \text{if } wetz_{i,j}^{n-1} = 0 \tag{3.85}$$

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:

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

$$fac_{i,j}^{n} = \frac{\left[z_b\right]_{i,j}^{n-1} - \left[\eta_{gw}\right]_{i,j}^{n-1}}{d_{wetlayer}} \qquad 0 \le fac \le 1 \tag{3.86}$$

3.9.3 Momentum balance

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

$$u_{gw} = -k_x \frac{dp_{gw}}{dx}$$
$$v_{gw} = -k_y \frac{dp_{gw}}{dy} \tag{3.87}$$

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

$$\left[\frac{dp_{gw}}{dx}\right]_{i,j}^{n} = \frac{p_{i+1,j}^{n} - p_{i,j}^{n}}{x_{z,i+1,j} - x_{z,i,j}}$$
$$\left[\frac{dp_{gw}}{dy}\right]_{i,j}^{n} = \frac{p_{i,j+1}^{n} - p_{i,j}^{n}}{y_{z,i,j+1} - y_{z,i,j}} \tag{3.88}$$

And horizontal flow is calculated by:

$$\left[u_{gw}\right]_{i,j}^{n} = -k_x \left[\frac{dp_{gw}}{dx}\right]_{i,j}^{n}$$
$$\left[v_{gw}\right]_{i,j}^{n} = -k_y \left[\frac{dp_{gw}}{dy}\right]_{i,j}^{n} \qquad (3.89)$$

3.9.4 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:

$$w_{i,j}^{n} = \left(\frac{\left[\eta_{gw}\right]_{i,j}^{n-1} - \left[z_b\right]_{i,j}^{n-1}}{\Delta t}\right) por \qquad \text{if} \quad \left[\eta_{gw}\right]_{i,j}^{n-1} \geq \left[z_b\right]_{i,j}^{n-1} \qquad (3.90)$$

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:

$$w = -k_z \left(\frac{dp}{dz} + 1\right) \qquad (3.91)$$

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 3.9.2. 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 ($z_s - z_b$).

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

$$w_{i,j}^{n} = \left((1 - fac) \frac{\left[\eta_{gw}\right]_{i,j}^{n-1} - \left[z_b\right]_{i,j}^{n-1}}{\Delta t} + fac_{i,j}^{n} \; k_z \left[\frac{z_s - z_b}{d_{infiltration}} \right]_{i,j}^{n-1} \right) por \tag{3.92}$$

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

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

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

$$w_{i,j}^{n} \le \frac{h_{i,j}^{n-1}}{\Delta t} \tag{3.93}$$

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.

$$\left[d_{infiltration} \right]_{i,j}^{n} = \left[d_{infiltration} \right]_{i,j}^{n-1} + \frac{w_{i,j}^{n} \Delta t}{por} \tag{3.94}$$

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.

$$\begin{aligned} \left[d_{infiltration} \right]_{i,j}^{n-1} &= \frac{1}{3} d_{wetlayer} && \text{if } wetz_{i,j}^{n-1} = 0 \\ \frac{1}{3} d_{wetlayer} &\le \left[d_{infiltration} \right]_{i,j}^{n-1} \le \left[z_b - \eta_{gw} \right]_{i,j}^{n-1} && \text{if } wetz_{i,j}^{n-1} = 1 \end{aligned} \tag{3.95}$$

3.9.5 Mass balance

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

$$\frac{d\eta_{gw}}{dt} + \frac{du_{gw}h_{ugw}}{dx} + \frac{dv_{gw}h_{vgw}}{dy} = \frac{w}{por} \tag{3.96}$$

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:

$$\begin{aligned} \left[h_{ugw} \right]_{i,j}^{n} &= \frac{\left[\eta_{gw} - z_{b,aquifer} \right]_{i,j}^{n} + \left[\eta_{gw} - z_{b,aquifer} \right]_{i+1,j}^{n}}{2} \\ \left[h_{vgw} \right]_{i,j}^{n} &= \frac{\left[\eta_{gw} - z_{b,aquifer} \right]_{i,j}^{n} + \left[\eta_{gw} - z_{b,aquifer} \right]_{i,j+1}^{n}}{2} \end{aligned} \tag{3.97}$$

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*).

$$\left.\begin{aligned} \left[u_{gw}h_{ugw} \right]_{i,j}^{n} &= \min\left(\left[u_{gw}h_{ugw} \right]_{i,j}^{n}, 0 \right) \\ \left[u_{gw}h_{ugw} \right]_{i-1,j}^{n} &= \max\left(\left[u_{gw}h_{ugw} \right]_{i-1,j}^{n}, 0 \right) \\ \left[v_{gw}h_{vgw} \right]_{i,j}^{n} &= \min\left(\left[v_{gw}h_{vgw} \right]_{i,j}^{n}, 0 \right) \\ \left[v_{gw}h_{vgw} \right]_{i,j-1}^{n} &= \max\left(\left[v_{gw}h_{vgw} \right]_{i,j-1}^{n}, 0 \right) \end{aligned}\right\} \text{ if } \quad \left[\eta_{gw} \right]_{i,j}^{n-1} \le \left[z_{b,aquifer} \right]_{i,j}^{n-1} + eps \tag{3.98}$$

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

$$\frac{\left[\eta_{gw}\right]_{i,j}^{n+1}-\left[\eta_{gw}\right]_{i,j}^{n}}{\Delta t}=-\frac{\left[u_{gw}h_{ugw}\right]_{i,j}^{n}-\left[u_{gw}h_{ugw}\right]_{i-1,j}^{n}}{x_{u,i,j}-x_{u,i-1,j}}-\frac{\left[v_{gw}h_{vgw}\right]_{i,j}^{n}-\left[v_{gw}h_{vgw}\right]_{i,j-1}^{n}}{y_{v,i,j}-y_{v,i,j-1}}+\frac{w_{i,j}^{n}}{por} \quad (3.99)$$

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, see also eq. (3.37):

$$\frac{\eta_{i,j}^{n+1}-\eta_{i,j}^{n}}{\Delta t}=-\frac{u_{i,j}^{n+1}h_{i,j}^{n}-u_{i-1,j}^{n+1}h_{i-1,j}^{n}}{x_{\mathrm{u,i,j}}-x_{\mathrm{u,i\text{-}1,j}}}-\frac{v_{i,j}^{n+1}h_{\mathrm{i,j}}-v_{i,j-1}^{n+1}h_{i,j-1}^{n}}{y_{\mathrm{v,i,j}}-y_{\mathrm{v,i,j\text{-}1}}}-w_{i,j}^{n} \quad (3.100)$$

3.10 Boundary conditions

3.10.1 Offshore flow boundary conditions

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.

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

$$u=2u_{i}-\sqrt{\frac{g}{h}}(z_{s}-z_{s0}) \quad (3.101)$$

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). This boundary condition assumes all incoming and outgoing waves propagate normal to the boundary. It is therefore only useful for 1D (flume like) simulations.

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 3.4a).

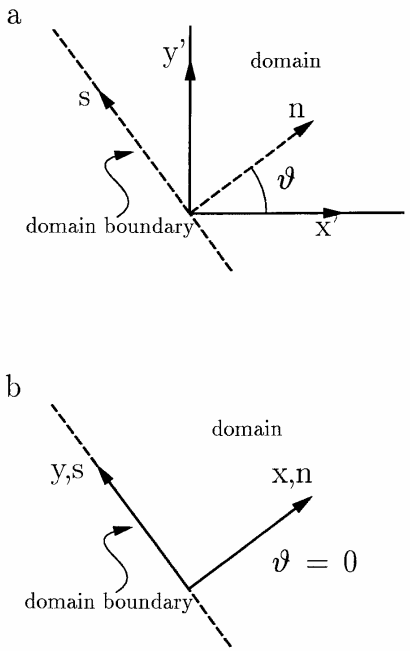

Figure 3.4 Coordinate system (a) for arbitrary angle υ between domain boundary and x-axis; (b) for υ=0

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 3.4b). The governing equations derived following Abbott (1979) and Verboom *et al.* (1981) then simplify to

$$\frac{\partial \beta^-}{\partial t} = -(u-c)\frac{\partial \beta^-}{\partial x} - v\frac{\partial \beta^-}{\partial y} + c\frac{\partial v}{\partial y} + g\frac{\partial h_0}{\partial x} + F_{\beta^-} \tag{3.102}$$

$$\frac{\partial \beta^+}{\partial t} = -(u+c)\frac{\partial \beta^+}{\partial x} - v\frac{\partial \beta^+}{\partial y} - c\frac{\partial v}{\partial y} + g\frac{\partial h_0}{\partial x} + F_{\beta^+} \tag{3.103}$$

$$\frac{\partial \gamma}{\partial t} = -u\frac{\partial \gamma}{\partial x} - v\frac{\partial \gamma}{\partial y} - g\frac{\partial \eta}{\partial y} + F_\gamma \tag{3.104}$$

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

$$\beta^- = u - 2c = u - 2\sqrt{g(h_0 + \eta)} \tag{3.105}$$

Here $\bar{u}$ 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

$$\gamma = v \tag{3.106}$$

The definition sketch in Figure 3.5 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 (3.102)-(3.104) originate from the right-hand side of the nonlinear shallow water equations, which imply that β- , β+, and γ are variables rather than constants.

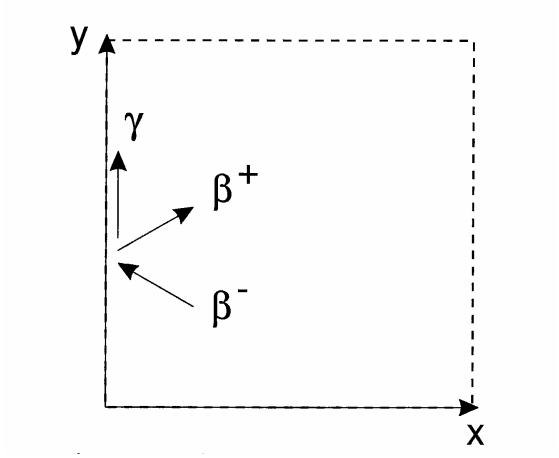

Figure 3.5 Definition sketch of the characteristics.

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 (3.102) which discretized in XBeach (similarly as the *x*-momentum equation) reads

$$\frac{\beta_{i,j}^{n+1}-\beta_{i,j}^{n}}{\Delta t}=-(u-c)\frac{\partial \beta}{\partial x}^{n}_{i,j}-v\frac{\partial \beta}{\partial y}^{n}_{i,j}+c\frac{\partial v}{\partial y}^{n}_{i,j}+g\frac{h_{\mathrm{i+1,j}}^{n}-h_{\mathrm{i,j}}^{\mathrm{n}}}{x_{\mathrm{i+1,j}}-x_{\mathrm{i,j}}}+\frac{F_{x,i,j}}{\rho h_{u,i,j}} \qquad (3.107)$$

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 (3.105) to lowest order as

$$\beta^{-}=u-2\sqrt{g(h_0+\eta)}=u-2\sqrt{gh_0}\left(1+\frac{1}{2}\frac{\eta}{h_0}\right) \qquad (3.108)$$

We further have the identities

$$\begin{aligned} u &= u_i + u_r \\ \eta &= \eta_i + \eta_r \\ u_i &= \sqrt{gh_0}\,\eta_i \cos\theta_i \\ u_r &= -\sqrt{gh_0}\,\eta_r \cos\theta_r \end{aligned} \qquad (3.109)$$

where the last two identities assume a wave propagating in shallow water with constant form 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. Inserting these identities into (3.108) and re-arranging gives

$$u_r=\left(\frac{\cos\theta_r}{\cos\theta_r+1}\right)\left[\beta^{-}+2\sqrt{gh_0}-u_i\left(\frac{\cos\theta_i-1}{\cos\theta_i}\right)\right] \qquad (3.110)$$

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

$$\theta_r=\arctan\left(\frac{u_r}{v_r}\right)=\arctan\left(\frac{u_r}{v-v_i}\right) \qquad (3.111)$$

Eqs. (3.110) and (3.111) are then solved iteratively to yield both u_r and θ_r. The final boundary condition is then the total velocity u $u = u_i + u_r$ at the boundary at the time level *n+1*.

3.10.2 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

3. 1D absorbing boundary condition

This boundary condition type is activated using “back = 0’ and is analoguous to (3.101) without the possibility of specifying an incoming particle velocity.

4. wall

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

5. 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).

3.10.3 Lateral flow boundary conditions

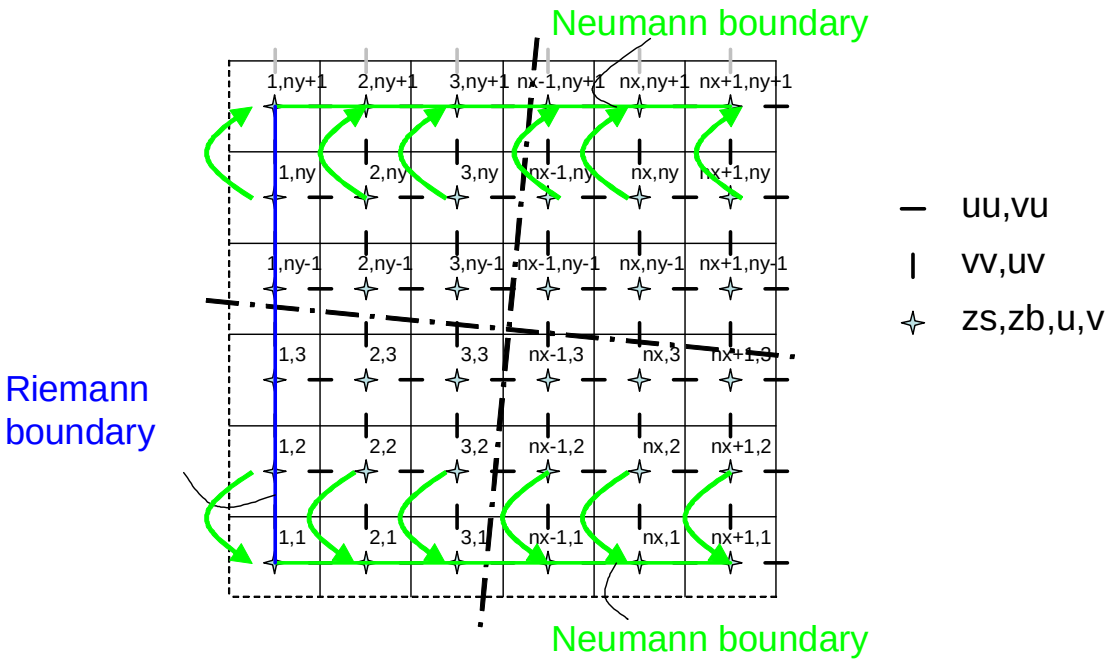

Figure 3.6 Stencil for Neumann-type boundary conditions.

For the lateral boundaries so-called Neumann boundaries can be used (“left = 0” and/or “right = 0”), 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” and/or “right = 1”. Wall boundary conditions are more effective than Neumann boundary conditions in 1D models.

3.10.4 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=7”.

- 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.

$$e_0(\vartheta) = E_{mean}\frac{\cos^m\left(\vartheta-\vartheta_m\right)}{\sum_{\vartheta_{\min}}^{\vartheta_{\max}}\cos^m\left(\vartheta-\vartheta_m\right)\Delta\vartheta} \quad ,\ \left|\vartheta-\vartheta_m\right|<\pi/2$$
$$E_{mean} = \frac{1}{8}\rho g H_{rms}^2 \tag{3.112}$$

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

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

$$e_0(\vartheta) = E_{mean}\frac{\cos^m\left(\vartheta-\vartheta_m\right)}{\sum_{\vartheta_{\min}}^{\vartheta_{\max}}\cos^m\left(\vartheta-\vartheta_m\right)\Delta\vartheta} * \frac{1}{2}\left(1+\cos\left(2\pi\left(\frac{t}{T_{long}}-\frac{y}{L_{long}}\right)\right)\right),$$
$$L_{long} = \frac{c_g T_{long}}{\sin(\vartheta_m)} \tag{3.113}$$

- 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.

$$e_0(\vartheta, y) = E_{t-\tau(y)}\frac{\cos^m\left(\vartheta-\vartheta_m\right)}{\sum_{\vartheta_{\min}}^{\vartheta_{\max}}\cos^m\left(\vartheta-\vartheta_m\right)\Delta\vartheta},$$
$$\tau(y) = \frac{y\sin(\vartheta_m)}{c_g} \tag{3.114}$$

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 5.5 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 section5.5.

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

The spectrum is determined by the peak period, wave height, spectral peakedness, mean angle and directional spreading (see section 5.5 for details on the input values). The routine follows the 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:

$$\eta(0,y,t)=\sum_{i=1}^{K} B_i \cos\left(k_i \sin(\theta_i)\, y - 2\pi f_i t + \varphi_i\right) \tag{3.115}$$

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 3.7. 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.

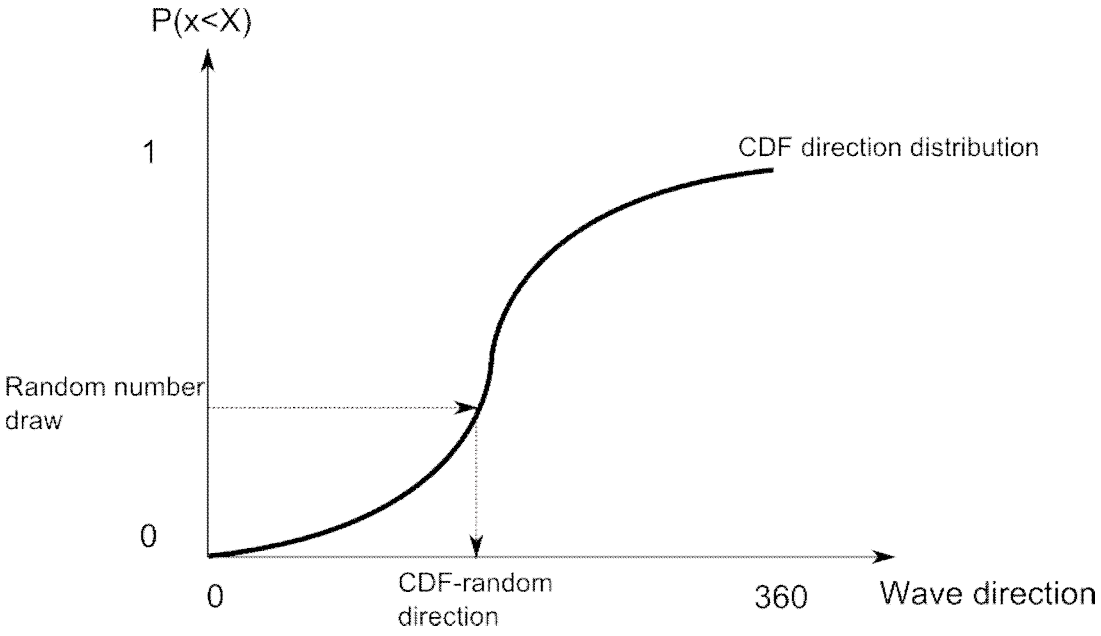

Figure 3.7 Random wave angles are chosen using a random number generator and the cumulative distribution function of the directional spreading of the input spectrum

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 (3.115) 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:

$$E_{\theta_i}(y,t) = \frac{1}{2}\rho g A_{\theta_i}(y,t) \tag{3.116}$$

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:

$$f_3 = f_2 - f_1 \tag{3.117}$$

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:

$$k_3 \equiv \left|\vec{k_1} - \vec{k_2}\right| = \sqrt{k_1^2 + k_2^2 - 2k_1k_2\cos(\theta_3)} \tag{3.118}$$

$$c_{g3} = \frac{2\pi f_3}{k_3} \tag{3.119}$$

$$\varphi_3 = \varphi_2 - \varphi_1 + \pi \tag{3.120}$$

Note that equation (3.120) 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:

$$\theta_3 = \arctan\left(\frac{k_2\sin\theta_2 - k_1\sin\theta_1}{k_2\cos\theta_2 - k_1\cos\theta_1}\right) \tag{3.121}$$

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

$$E_3(f_3) = 2\int_{\Delta f}^{\infty}\int_0^{2\pi}\int_0^{2\pi} D^2\left(f + f_3, -f, |\theta_1 - \theta_2| + \pi\right) \cdot E(f + f_3, \theta_1)E(f, \theta_2)\, d\theta_2 d\theta_1 df \tag{3.122}$$

Where in equation (3.122) 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:

$$D_{surface} = D_{bed} \frac{\cosh\left(k_3 h\right)}{\cosh\left(k_1 h\right)\cosh\left(k_2 h\right)} \tag{3.123}$$

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

$$A_3 = \sqrt{2E_3 df} \tag{3.124}$$

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:

$$q_x(0,t) = IFFT\left[\sum_{i=1}^{K} \frac{A_{3,i}}{2} e^{-i\varphi_{3,i}} c_{g3,i} \cos\theta_{3,i}\right] \tag{3.125}$$

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

$$q_x(y,t) = IFFT\left[\sum_{i=1}^{K} \frac{A_{3,i}}{2} e^{-i\varphi_{3,i}} c_{g3,i} \cos\theta_{3,i} e^{-ik_{y3,i}y}\right] \tag{3.126}$$

The cross-shore flux (3.126) and the wave energy (3.116) are specified along the boundary.

3.9.5 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).

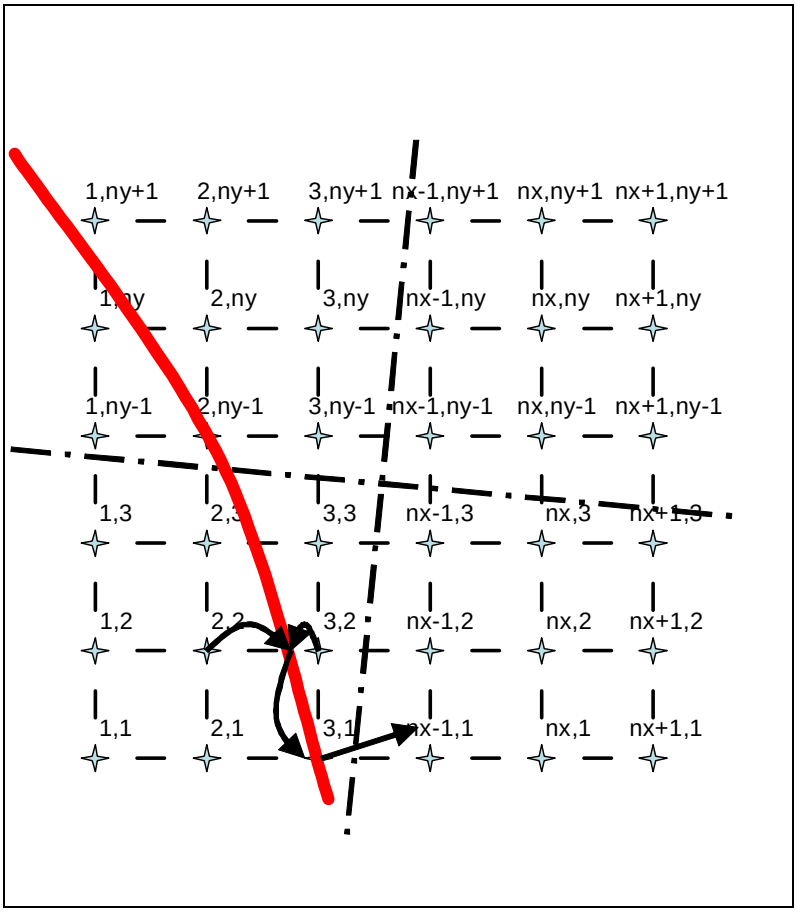

Figure 3.8 Example of interpolation at the lateral boundary.

3.10.5 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:

$$\begin{aligned}
\left[\eta_{gw}\right]_{1,j}^{n} &= \min\left(\left[\eta_{gw}\right]_{1,j}^{n},\left[z_b\right]_{1,j}^{n}\right)\\
\left[\eta_{gw}\right]_{nx+1,j}^{n} &= \min\left(\left[\eta_{gw}\right]_{nx+1,j}^{n},\left[z_b\right]_{nx+1,j}^{n}\right)\\
\left[\eta_{gw}\right]_{i,1}^{n} &= \min\left(\left[\eta_{gw}\right]_{i,1}^{n},\left[z_b\right]_{i,1}^{n}\right)\\
\left[\eta_{gw}\right]_{i,ny+1}^{n} &= \min\left(\left[\eta_{gw}\right]_{i,ny+1}^{n},\left[z_b\right]_{i,ny+1}^{n}\right)
\end{aligned} \tag{3.127}$$

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

$$z_{b,aquifer} = \min\left(z_{b,aquifer}, z_b - eps\right) \tag{3.128}$$

Offshore boundary condition

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

$$\left[p_{gw}\right]_{i,1}^{n} = \left[z_s\right]_{i,1}^{n} \tag{3.129}$$

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:

$$\left[p_{gw}\right]_{i,ny+1}^{n} = \left[z_s\right]_{i,ny+1}^{n} \tag{3.130}$$

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

$$\left[p_{gw}\right]_{i,ny+1}^{n} = \left[p_{gw}\right]_{i,ny+1}^{1} \tag{3.131}$$

Lateral boundary conditions

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

$$\left[p_{gw} \right]_{i,1}^{n} = \left[p_{gw} \right]_{i,2}^{n}$$
$$\left[p_{gw} \right]_{i,ny+1}^{n} = \left[p_{gw} \right]_{i,ny}^{n} \tag{3.132}$$

Initial conditions

The bed of the aquifer and the initial groundwater head must be specified, see section 5.10 for a description. The initial groundwater level is calculated from the initial groundwater head.

4 Compiling the model

The source code has been extensively tested in a version compiled under Compaq Visual Fortran version 6.6. 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 open 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.

params.txt
- depth file
- (optional) x- and y-grid files
- (optional) water level time series
- (optional) incident wave energy and long wave time series
- (optional) spectral input (time series)

5.2 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	2	10000	-	
dx	grid size in x	0	-1e9	1e9	m	
dy	grid size in y	0	-1e9	1e9	m	
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	-360	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

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.

5.3 Wave input

5.3.1 Action balance

For simple simulations with stationary waves or regular wave groups the wave parameters Hrms, Trep and dir0 can be entered directly in the *params.txt* file; for more advanced options see section 5.5. 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) .
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 3.3.

keyword	description	default value	minimum value	maximum value	unit	remarks
dir0	mean wave direction (Nautical convention)	270	180	360	deg	instat=0-3 only
Hrms	rms wave height	1	0	10	m	instat=0,1 only
wavint	interval between stationary wave module calls	1	1	3600	s	*instat* = 0 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	10	1	20	s	overrules

	representative period					value for Tm01
Tlong	long wave / wave group period	80	20	300	s	*instat* = 1 only
thetamin	lower directional limit	-80	-180	180	deg	angle w.r.t computational x-axis
thetamax	upper directional limit	80	-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	-	
taper	time to spin up wave boundary conditions	100	0	1000	s	not for stationary waves
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
wci	wave current interaction option	0	0	1	-	*instat* = 0 only
scheme	Switch numerical schemes for wave action balance	1	1	2	-	1 = Upwind 2 = Lax Wendroff

5.3.2 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 model will be suitable.

keyword	description	default value	minimum value	maximum value	unit	remarks
break	option breaker model	3	1	3	-	1=roelvink, 2 = baldock, 3 = roelvink adapted
gamma	breaker parameter in Baldock or Roelvink formulation	0.6	0.4	0.9	-	
alpha	wave dissipation	1	0.5	2	-	

	coefficient					
n	power in roelvink dissipation model	5	5	20	-	

5.3.3 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 value	minimum value	maximum value	unit	remarks
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	-	

5.4 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	friction coefficient	0.003	0	0.1	-	
C	Chezy coefficient	sqrt(g/cf)	20	100	m1/2/s	alternative for cf, overrules cf if set
nuh	horizontal background viscosity	0.5	0	1	m2/s	
nuhfac	viscosity coefficient for roller induced turbulent horizontal viscosity	0	0	1	-	
nuhv	additional shear	1	1	20	-	svendsen

	dispersion factor					and putrevu (1994)
lat	latitude	0	0	90	deg N	
wearth	angular velocity of earth	1/24	0	1	1/hour	

5.5 Wave boundary conditions

5.5.1 User input

The user input options include *instat* = 0,1,2,3,4, 5, 6 or 7 in *params.txt*. The meaning of each is summarised below:

- Instat 0: Stationary wave conditions specified by Hrms, Trep, dir0
- Instat 1: Regular bichromatic wave groups specified by Hrms, Trep, dir0, Tlong
- Instat 2: Read in unidirectional time series of wave energy from file *bc\gen.ezs*, direction given by dir0
- Instat 3: Reads in unidirectional time series of wave energy and incident long wave from file *bc\gen.ezs*, direction given by dir0
- Instat 4: Standard JONSWAP spectrum, based on user-input spectrum coefficients
- Instat 5: Unmodified SWAN 2D spectrum output file
- Instat 6: Formatted variance-density spectrum file
- Instat 7: Reuse boundary condition files from an earlier XBeach simulation

The new module in XBeach converts the input spectral data into boundary condition time series that can be read by the main XBeach program. Boundary condition time series that are created from input spectral data are non-stationary. Due to the randomness used to generate the time series, see *Theory*, no two time series generated from the same input data will be identical. They will however share the same statistical properties. The user can avoid random variations between simulations by reusing the boundary condition time series of an earlier XBeach simulation.

The user has the option to carry out the simulation using one wave spectrum during the entire simulation period. In this case the spectral properties of the generated waves will be constant in time. Alternatively, the user may choose to vary the spectral properties of the boundary conditions in time. In this case the spectral properties of the generated waves depend on a time series of input spectral data.

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.

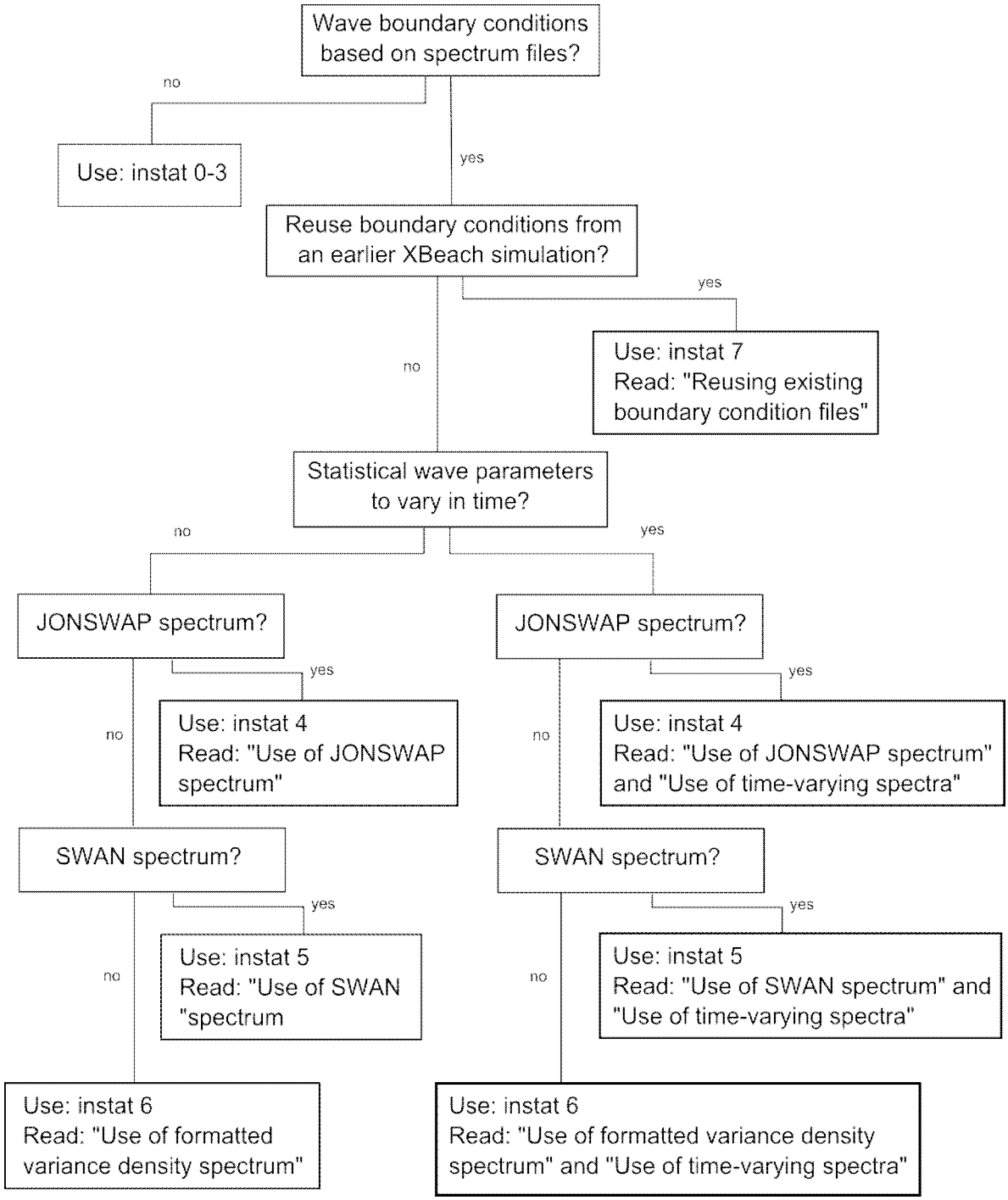

Figure 5.1 Reading guide

5.5.2 Use of JONSWAP spectrum

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.*

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
```

5.5.3 Use of SWAN spectrum

The new XBeach module 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 attempt to 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.*

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 =*'.

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
```

5.5.4 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
```

5.5.5 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.

5.5.6 Use of time-varying spectra

The new XBeach module allows the user to specify time-varying wave spectra on the offshore boundary. This is done by feeding in several input data files such as those used for instat 4, 5 or 6, and specifying the duration for which these spectra should occur.
To make use of this option, the user must specify the instat value (4, 5 or 6) associated with the input data type for the wave boundary conditions. XBeach will then attempt to read a list of input data filenames from a separate file specified by '*bcfile =*' in *params.txt*. This keyword is the same keyword as used for non-time-varying spectra. In order for XBeach to differentiate between time-varying and non-time-varying wave spectra, the file must have the following format.

The first word in the file must be the keyword 'FILELIST'. In the following lines, each line contains the duration of this wave spectrum condition in seconds (similar to *'rt'* in *params.txt*), the required time step in this boundary condition file in seconds (similar to *'dtbf'* in *params.txt*) and the name of the input data file used to generate these boundary conditions. The duration and boundary condition time step in this file overrules *'rt'* and *'dtbf'* in *params.txt*. XBeach will now not reuse any boundary condition time series, so the user must ensure that the total record length is greater than or equal to the simulation time.

A typical input file contains the following:

```
FILELIST
1800 0.2 jonswap1.inp
1800 0.2 jonswap1.inp
1350 0.2 jonswap2.inp
1500 0.2 jonswap3.inp
1200 0.2 jonswap2.inp
3600 0.2 jonswap4.inp
```

Note: It is not possible to use a mix of JONSWAP, SWAN and Variance Density files. It is also not possible to vary *dthetaS_XB* between files in case of non-nautical SWAN output.

5.5.7 Examples

In order to clarify the instructions given above, a number of examples are given. In the first example, the user wishes to use a constant JONSWAP spectrum. In order to save calculation time and memory use, the user decides to recycle the boundary condition file every 30 minutes. Wave input parameters in the *params.txt* file are given below:

```
Boundary condition options
instat   = 4

Required for instat 4,5,6
bcfile   = jonswap.inp

if file is reused, else this information in file-list:
rt       = 1800
dtbc     = 0.2
```

The file *jonswap.inp* contains information as described in the section “use of one JONSWAP input file”. As the user has only one input file, *rt* and *dtbc* must be specified in *params.txt*.

In the second example the user wishes to use a constant JONSWAP spectrum, but does not want to reuse the same boundary condition file as in the previous example. This can be achieved by using the multiple files option, but only using one JONSWAP data file. For instance in *params.txt*:

```
Boundary condition options
instat   = 4

Required for instat 4,5,6
bcfile   = jonswaplist.txt
```

And in the file jonswaplist.txt:

```
FILELIST
1800 0.2 jonswap.inp
1800 0.2 jonswap.inp
1800 0.2 jonswap.inp
1800 0.2 jonswap.inp
1800 0.2 jonswap.inp
1800 0.2 jonswap.inp
```

Note that in this case if the simulation time is greater than 3 hours (10800 seconds), XBeach will abort due to lack of input data.

In the third example the user wishes to use one SWAN 2D spectrum file. The output of the SWAN model is in nautical degrees, so dthetaS_XB does not need to be specified. Wave input parameters in the *params.txt* file are given below:

```
Boundary condition options
instat   = 5

Required for instat 4,5,6
bcfile   = swan14a.sp2

if file is reused, else this information in file-list:
rt       = 3600
dtbc     = 0.5
```

The file swan14a.sp2 contains information as described in the section “use of one SWAN input file”. As the user has only one input file, *rt* and *dtbc* must be specified in *params.txt*. If the simulation time is more than one hour (3600 seconds), the boundary condition file is reused.

In the final example, the user wishes to reuse the exact boundary conditions of the simulation in the previous example. In this case the user copies the files *ebcflist.bcf* and *qbcflist.bcf* and E_reuse.bcf and q_reuse.bcf to the current directory (the directory with the XBeach executable). The *params.txt* would now contain the following options for the wave boundary conditions:

```
Boundary condition options
instat   = 7
```

5.5.8 Explanation of spectral input/output

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.

```
---------------------------------------------------
Grid input

nx       = 104
ny       = 124
xori     = 0.
yori     = 0.
alfa     = 0.
thetamin = -90.
thetamax = 90.
dtheta   = 20
depfile  = depth.grd
posdwn   = -1
vardx    = 1

Required if vardx 1
xfile    = x.grd
yfile    = y.grd

Required if vardx 0
dx       = 1.
dy       = 1.

---------------------------------------------------
Numerics input
CFL      = 0.2
eps      = 0.01

---------------------------------------------------
Time input
tstart   = 0
tint     = 1.
tstop    = 1800
taper    = 20

---------------------------------------------------
General constants
rho      = 1000
g        = 9.81

---------------------------------------------------
Wind boundary conditions
rhoa     = 1.25
Cd       = 0.002
windv    = 0.
windth   = 90.

---------------------------------------------------
Water level boundary conditions
tideloc  = 1

Required if tideloc > 0
zs0file  = zs0input.bcf
tidelen  = 2
paulrevere = 1

Required if tideloc 0
zs0      = 4.

---------------------------------------------------
Boundary wave conditions absorbing
front    = 1
left     = 0
right    = 0
back     = 0

---------------------------------------------------
```

Continued on next page …

```
Boundary wave conditions generating
instat   = 4
ARC      = 1

Required for instat 0,1,2,3
dir0     = 270.

Required for instat 0,1
Hrms     = 0.4
Tm01     = 6.0
m        = 200

Required for instat 0
waveint    = 1

Required for instat 1
Tlong    = 31.2

Required for instat 5 (if SWAN is Cartesian):
Angle in Cartesian degrees to rotate the x-axis in SWAN to the x-axis in XBeach
dthetaS_XB  = 0.

Required for instat 4,5,6:
Name of boundary condition file
bcfile   = jonswap.inp

Required for instat 4,5,6 if spectra are not time-varying:
Record length for the boundary condition file
rt       = 1800.
Boundary condition file time step
dtbc     = 0.2

-----------------------------------------------------
Wave calculation options
break    = 1
wci      = 0
roller   = 1
beta     = 0.1
gamma    = 0.55
gammax   = 5.
alpha    = 1.
delta    = 0.0
n        = 10.
order    = 2

--------
```

5.6 Flow boundary conditions

The input parameters below refer to functionality described in section 3.10.1. With these options you can mimic the seaward, lateral and basin boundary conditions at a real stretch of coast or in a flume. Default values give weakly reflective boundary conditions at offshore and basin sides and Neumann boundary conditions at lateral sides. With ARC on or off you can simulate a flume with or without active wave absorption; with order at 1 or 2 you can generate waves without resp. with bound long waves. With the keywords *left* and *right* set to 1 you can turn the sides of the flume or basin into walls.

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	1	-	0 = 1D absorbing, 1 = 2D absorbing
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
back	bayside boundary condition	2	0	2	-	0 = 1D absorbing, 1 = wall, 2 = 2D absorbing
epsi	weighting factor between steady flow and particle velocity	0	0	1	-	0=waves, 1=steady flow for seaward and bayward boundaries only
left	left lateral boundary condition	0	0	1	-	0 = Neumann, 1 = wall
right	right lateral boundary condition	0	0	1	-	0 = Neumann, 1 = wall
carspan	free long wave input	0	0	1	-	0 = use cg (default); 1 = use sqrt(gh) in *instat* = 3

						for c&g tests
fcutoff	low freq cutoff frequency for boundary conditions	0	0	40	Hz	
sprdthr	threshold above which spec dens are read in	0.08	0	1	-	

5.7 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	-	
tidelen	length of tidal record	0	0	1000000	-	.

5.8 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)	90	-180	180	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.

5.9 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 wav-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	5	0.4	5	-	
hmin	threshold water depth for concentration and return flow	0.01	0.001	1	m	
eps	threshold depth for drying and flooding	0.1	0.001	1	m	
umin	threshold velocity upwind scheme	0.1	0.001	5	m/s	
Hwci	depth below which wci is not applied	0.01	0.001	1	m	

5.10 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	type	default value	description
qwflow	integer	0	Turns the groundwater module on (1) or off (2)
kx	real	1E-4	The Darcy permeability coefficient of the aquifer in x-direction [ms^{-1}]
ky	real	kx	The Darcy permeability coefficient of the aquifer in y-direction [ms^{-1}]
kz	real	kx	The Darcy permeability coefficient of the aquifer in z-direction [ms^{-1}]
dwetlayer	real	0.2d0	Thickness of the interaction layer $d_{wetlayer}$ [m]
aquiferbot	real	minval(zb)-3.d0	The bed level of the aquifer is set to the value of *aquiferbot* across the entire domain [m]
aquiferbotfile	character	-	The initial bed level of the aquifer is

			read from the file named by the keyword *aquiferbotfile*. The file has the same dimensions as the bathymetry file. This keyword overrides the *aquiferbot* keyword
gw0	real	0.d0	The initial groundwater head is set to the value of *gw0* across the entire domain [m]
gw0file	character	-	The initial groundwater head is read from the file named by the keyword *gw0file*. The file has the same dimensions as the bathymetry 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
```

5.11 Hydrodynamic simulation

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 5.15. 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	tint	0.01	100000		supercedes tint if

	values					specified
tintm	time interval output mean global values	ting	0.01	100000		
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
tstop	stop time simulation	2000	1	1000000	s	
CFL	maximum courant number	0.2	0.1	0.9	-	actual CFL number varies during calculation
nonh	nonhydrostatic option	0	0	1	-	not implemented yet

5.12 Sediment transport

form	equilibrium sed. conc. formulation	1	1	3	-	1 = Soulsby van rijn, 2 = Van Rijn 2008 (not tested yet)
dico	diffusion coefficient	1	0	10	m2/s	
z0	zero flow velocity level in Soulsby van Rijn (1997) sed.conc. expression	0.006	0.0001	0.05	m	
facsl	bed slope factor	0	0	1.6		
hswitch	water depth at interface from wetslp to dryslp	0.1	0.01	1		
rhos	density of sediment	2650	2400	2800	kg/m3	
struct	option for hard structures	0	0	1		0 = no revetment, 1 = multiple sediment classes

ngd	number of sediment classes	1	1	3	-	
nd	number of sediment class layers	1	1	20	-	
dzg	thickness of sediment class layers	0.1	0.01	1	-	
D50	D50 grain diameter first (or only) class of sediment	0.0002	0.00005	0.001	m	
D90	D90 grain diameter first (or only) class of sediment	0.0003	0.00005	0.001	m	
D502	D50 grain diameter second class of sediment	0	0	0.001	m	
D902	D90 grain diameter second class of sediment	0	0	0.001	m	
D503	D50 grain diameter third class of sediment	0	0	0.001	m	
D903	D90 grain diameter third class of sediment	0	0	0.001	m	
sedcal1	calibration factor for sediment class 1	1	0	10		
sedcal2	calibration factor for sediment class 2	1	0	10		
sedcal3	calibration factor for sediment class 3	1	0	10		
tsfac	max value for fall velocity	0.1	0.01	1	[-]	
thetanum	option to switch between central and upwind scheme	1	0.5	1	[-]	
smax	max shields value for overwash	-1	0	3	[-]	

5.13 Morphological updating

morfac	morphological factor	0	0	1000	-	
morstart	start time of morphological	120	0	10000	s	

	updates					
por	porosity	0.4	0.3	0.5	-	
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		

5.14 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.

5.14.1 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.

5.14.2 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.*

5.14.3 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*.

5.14.4 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.

5.15 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.

5.15.1 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*.

5.15.2 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.

5.15.3 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

5.15.4 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*.

5.16 Example

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

5.17 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.

```
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

5.18 Variable keywords

The keywords to define which model variables are to be output are givenin the tables below.

Keyword	Unit	Description
c :	[m/s]	wave celerity
cg :	[m/s]	group velocity
D :	[W/m^2]	dissipation
DR :	[W/m2]	roller energy dissipation
dzav :	[m]	total bed level change due to avalanching
dzbdt :	[m/s]	rate of change bed level
E :	[Nm/m^2]	wave energy
Fx :	[N/m^2]	wave force x-direction
Fy :	[N/m^2]	wave force y-direction
H :	[m]	wave height
hh :	[m]	water depth
k :	[rad/m]	wave number
kb :	[m^2/s^2]	near bed turbulence intensity due to depth induced breaking
maxzs :	[m]	maximum elevation in simulation
minzs :	[m]	minimum elevation in simulation
n :	[-]	ratio group velocity/wave celerity
R :	[Nm/m^2]	roller energy
sedero :	[m]	cum. sedimentation/erosion
sigm :	[rad/s]	mean frequency
Sxx :	[N/m]	radiation stress
Sxy :	[N/m]	radiation stress
Syy :	[N/m]	radiation stress

Keyword	Unit	Description
thetamean :	[rad]	mean wave angle
u :	[m/s]	(GLM) x-velocity cell centre (for output)
ue :	[m/s]	Eulerian mean x-velocity cell centre (for output)
uoff :	[m/s]	offshore directed peak orbital velocity
uon :	[m/s]	onshore directed peak orbital velocity
urms :	[m/s]	orbital velocity
usd :	[m/s]	return flow due to roller after breaker delay
ust :	[m/s]	Stokes drift
ustr :	[m/s]	return flow due to roller
uwf :	[m/s]	x-comp. Stokes drift
v :	[m/s]	(GLM) y-velocity cell centre (for output)
ve :	[m/s]	Eulerian mean y-velocity cell centre (for output)
vmag :	[m/s]	velocity magnitude in cell centre
vwf :	[m/s]	y-comp. Stokes drift
wetz :	[-]	mask wet/dry eta-points
zb :	[m]	bed level
zs :	[m]	water level
zs0 :	[m]	water level due to tide alone

Add some kind of output for c, ceq and Su, Sv, summed across all sediment classes?
Add sediment distribution in bed

6 Distribution and maintenance

XBeach.org
Repos.deltares.nl/xbeach/trunk

7 Tutorial

7.1 Profile mode

For many applications it is useful to start by running in profile mode. This is certainly true for more or less uniform coasts. The way the model operates is that it assumes longshore gradients to be zero. Numerically this is done by copying water levels and wave energy from row 2 to row 1 and from row 2 to row 3; the number of *cells* in y-direction must be ny=2, which gives 3 grid *points* in y-direction.

7.1.1 Creating grid and depth information

An easy way to create grid and depth for XBeach given a table of (x,zb) values is by using a Maatlab script like below. The script reads the table from the file *profile.dat*; based on a given dx (5 m in our case) it creates a computational grid *xdata* and interpolates the depth data onto the grid as *zdata*. It then writes the block of grid data form the *params.txt* file to a file *griddata.txt*; this info can be cut and pasted into the *params.txt* file. The depth data is written to the file 'profile.dep'. It is written three times, once for each grid row.

```
dx=5;
profname='profile.dat';

xz=load(profname)
xdata=xz(:,1);
zdata=xz(:,2);

xgrid=[xdata(1):dx:xdata(end)];
zgrid=interp1(xdata,zdata,xgrid);

fi=fopen('griddata.txt','wt');
fprintf(fi,'nx      = %3i \n',length(xgrid)-1);
fprintf(fi,'ny      = %3i \n',2);
fprintf(fi,'dx      = %6.1f \n',dx);
fprintf(fi,'dy      = %6.1f \n',dx);
fprintf(fi,'xori    = %10.2f \n',xdata(1));
fprintf(fi,'yori    = %10.2f \n',0);
fprintf(fi,'alfa    = %10.2f \n',0);
fprintf(fi,'depfile = profile.dep \n')
fprintf(fi,'posdwn  = -1 \n')
fclose(fi);

fi=fopen('profile.dep','wt');
for i=1:3
    fprintf(fi,'%7.3f ',zgrid);
    fprintf(fi,'\n');
```

```
end
fclose(fi);
```

Next, we cut and paste the contents of *griddata.txt* into an example *params.txt* file and start editing the rest of the file.

```
nx      = 200
ny      =   2
xori    =       0.00
yori    =       0.00
alfa    =       0.00
depfile = profile.dep
posdwn  = -1
vardx   = 0
dx      =    5.0
dy      =    5.0
```

Note that only lines with an '=' are considered, any other lines are comments. We start by specifying the theta-grid:

```
thetamin = -90
thetamax = 90
dtheta   =  180
thetanaut= 0
```

This means we have a single directional bin that covers the whole directional spectrum, since in our 1D case we cannot represent longshore variations.

In stationary mode, waves may be obliquely incident and a longshore current will be generated. In instationary mode (with wave energy varying on the wave group scale) the wave direction must be shore-normal.
We specify the duration of the run, the CFL number that is used to determine the automatic timestep, we select the numerical scheme for wave propagation and for morphology updating and set a number of limiters.

```
---------------------------------------------------
Time input
---------------------------------------------------
tstop    = 1800
---------------------------------------------------
Numerics input
---------------------------------------------------
CFL      = 0.9
scheme   = 1
thetanum = 1
---------------------------------------------------
Limiters
---------------------------------------------------
gammax   = 2
hmin     = .01
eps      = .001
umin     = 0.
hwci     = 0.1
```

We continue with the boundary conditions; the seward boundary and landward boundary are 2D Riemann boundaries, the side boundaries Neumann,

```
--------------------------------------------------
Boundary numerics
--------------------------------------------------
front    = 1
back     = 2
left     = 0
right    = 0
```

We set a simple tidal boundary by specifying *tideloc*=0 and *zs0*=0. Note that the rest of the text is ignored as long as no '=' is used.

```
--------------------------------------------------
Boundary tide options
--------------------------------------------------
tideloc  = 0

if tideloc 0
   zs0  = 0
else if tideloc >0
   zs0file
   tidelen
   paulrevere
```

We set a negligible taper time and select *instat=4* to specify a parametric Jonswap spectrum. We use the default options for wave generation, with reflection compensation (ARC) and including incident bound waves.(order=2)

```
--------------------------------------------------
Wave generating boundary options
--------------------------------------------------
taper   = 1.e-3
instat  = 4
ARC     = 1
order   = 2
```

We set some wave calculation and flow calculation options:

```
--------------------------------------------------
Wave calculation options
--------------------------------------------------
break   = 1
gamma   = 0.55
n       = 10
--------------------------------------------------
Flow calculation options
--------------------------------------------------
cf      = 0.003
nuhfac  =1
```

Then some sediment transport parameters:

```
--------------------------------------------------
Sediment transport calculation options
--------------------------------------------------
form    =1
```

```
smax    =0.8
tsfac   =0.1
D50     =.0002
D90     =.0003
rhos    =2650
ngd     = 1
```

We specify the morphological parameters as follows:

```
-----------------------------------------------------
Morphological calculation options
-----------------------------------------------------
morfac    = 10
morstart  = 300
por       = 0.4
dryslp    = 1.0
wetslp    = 0.3
hswitch   = 0.1
```

And finally we specify the output requests:

```
-----------------------------------------------------
Output options
-----------------------------------------------------
tstart    = 0
tintg     = 1
tintp     = 1
tintm
tsglobal
tspoints
tsmean
nglobalvar
-----------------------------
npoints = 1
150 2 4 H#zs#zb#u#
nrugauge=  1
2 4 H#zs#zb#u#
```

8 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 Blank params.txt

```
--------------------------------------------------
General constants
--------------------------------------------------
rho
g

--------------------------------------------------
Grid input
--------------------------------------------------
nx
ny
xori
yori
alfa
depfile
posdwn
vardx

if vardx 0 (regular sized grid)
   dx
   dy
else if vardx 1 (varying dx and dy)
   xfile
   yfile

thetamin
thetamax
dtheta
thetanaut

--------------------------------------------------
Time input
--------------------------------------------------
tstop

--------------------------------------------------
Numerics input
--------------------------------------------------
CFL
scheme
thetanum

--------------------------------------------------
Limiters
--------------------------------------------------
gammax
hmin
eps
umin
hwci

--------------------------------------------------
Boundary numerics
--------------------------------------------------
front
back
left
right

--------------------------------------------------
Advanced wave boundary options
```

```
--------------------------------------------------
Boundary tide options
--------------------------------------------------
tideloc

if tideloc 0
   zs0
else if tideloc >0
   zs0file
   tidelen
   paulrevere

--------------------------------------------------
Wave generating boundary options
--------------------------------------------------
taper
instat
ARC
order

if instat 0,1,2
   dir0
   Hrms
   Trep
   m

   if instat 0
      waveint

   if instat 1
      Tlong

else if instat 4,5,6
   bcfile

   if instat 4,5,6 and bcfile is a spectrum file (not a list file)
      rt
      dtbc

   if instat 5 and SWAN spectrum output is carthesian
      dthetaS_XB

--------------------------------------------------
Wind options
--------------------------------------------------
rhoa
Cd
windv
windth

--------------------------------------------------
Coriolis options
--------------------------------------------------
lat
wearth

--------------------------------------------------
Wave calculation options
--------------------------------------------------
wci
break
```

```
--------------------------------------------------
Flow calculation options
--------------------------------------------------
Choose between C or cf
C
cf

nuh
nuhfac
nuhv

--------------------------------------------------
Sediment transport calculation options
--------------------------------------------------
form

smax
tsfac
dico
z0
facsl
sedcal1
struct

D50
D90
rhos
ngd

if ngd>1
   nd
   dzg
   D502
   D902
   D503
   D903
   sedcal2
   sedcal3

--------------------------------------------------
Morphological calculation options
--------------------------------------------------
morfac
morstart
por
dryslp
wetslp
hswitch

--------------------------------------------------
Output options
--------------------------------------------------
tstart
tintg
tintp
tintm
tsglobal
tspoints
tsmean

nglobalvar
```