Keywords

Place keywords here

Summary

Place summary here

References

Place references here

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Version | Date | Author | Initials | Review | Initials | Approval | Initials |
|  | feb. 2015 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |
| --- |
| State  draft  This is a draft report, intended for discussion purposes only. No part of this report may be relied upon by either principals or third parties. |

Contents

1 Introduction 1

2 Processes and model formulation 2

2.1 Domain and definitions 2

2.2 Hydrodynamics options 2

2.2.1 Stationary mode 2

2.2.2 Non-stationary (surfbeat) mode 2

2.2.3 Wave resolving mode 2

2.3 Short wave propagation 3

2.3.1 Wave action balance 3

2.3.2 Dissipation 4

2.3.3 Roller energy balance 7

2.4 Shallow water equations 8

2.4.1 Horizontal viscosity 8

2.4.2 Bed shear stress 9

2.5 Nonhydrostatic pressure correction 9

2.6 Groundwater flow 10

2.6.1 Hydrostatic 10

2.6.2 Non-hydrostatic 12

2.7 Sediment transport 17

2.7.1 Advection-diffusion 17

2.7.2 General parameters 17

2.7.3 Transport formulations 18

2.7.4 Wave asymmetry 20

2.8 Bottom updating 21

2.8.1 Due to sediment fluxes 21

2.8.2 Avalanching 21

2.8.3 Bed composition 22

3 Boundary conditions 23

3.1 Waves 23

3.1.1 Spectra 23

3.1.2 Non-spectra 24

3.1.3 Lateral boundary conditions 24

3.2 Shallow water equations 25

3.2.1 Absorbing-generating 25

3.2.2 River and point discharge 25

3.2.3 Ship motion 26

3.2.4 Lateral boundaries 26

3.2.5 Tide and surge 26

3.3 Sediment transport 27

4 Input description 28

4.1 General 28

4.2 Physical processes 29

4.3 Grid and bathymetry 30

4.4 Waves input 33

4.4.1 Stationary wave boundary conditions 35

4.4.2 Spectral wave boundary conditions 36

4.4.3 Boundary conditions for non-hydrostatic model 43

4.4.4 Special types of wave boundary conditions 43

4.4.5 Temporally and/or spatially varying wave boundary conditions 43

4.4.6 Notes on the generation of wave boundary conditions 44

4.5 Flow, tide and surge input 45

4.5.1 Flow boundary conditions 45

4.5.2 Time-varying tide/surge 47

4.6 Water level (dam break) 48

4.7 Wind input 49

4.8 Sediment input 49

4.9 Vegetation input 51

4.10 Discharge input 52

4.11 Drifters input 53

4.12 Ship input 54

4.13 Output selection 55

4.13.1 Output types 58

4.13.2 Output times 59

4.13.3 Output format 61

4.14 Time parameters 62

5 Bibliography 64

6 Appendices 66

6.1 Hands on exercises (based on basic XBeach exercises) 66

6.1.1 Dune erosion at Delfland, Netherlands (1D) 66

6.1.2 Nourishment scenarios near Kijkduin, Holland (1D) 67

6.1.3 Overwash at Santa Rosa Island , USA (2DH) 68

6.1.4 Yanchep perched beach and natural breakwater (2DH) 69

6.2 Advanced model coefficients 70

6.2.1 Wave numerics 70

6.2.2 Wave dissipation 70

6.2.3 Rollers 72

6.2.4 Wave-current interaction 72

6.2.5 Bed friction and viscosity 72

6.2.6 Flow numerics 73

6.2.7 Sediment transport 74

6.2.8 Sediment transport numerics 78

6.2.9 Quasi-3D sediment transport 78

6.2.10 Morphology 79

6.2.11 Bed update 80

6.2.12 Groundwater flow 81

6.2.13 Non-hydrostatic correction 82

6.2.14 Physical constants 84

6.2.15 Coriolis force 84

6.2.16 MPI 85

6.2.17 Output projection 85

6.3 Numerical implementation 87

6.3.1 Grid types 87

6.3.2 Grid set-up 87

6.3.3 Shallow water equations 87

6.3.4 Wave action balance 90

6.3.5 Shallow water equations 91

6.3.6 Groundwater flow 91

6.3.7 Sediment transport 96

6.3.8 Bottom updating schemes 97

6.3.9 Non-hydrostatic 98

# Introduction

Dano, Ad, Ap

# Processes and model formulation

## Domain and definitions

Dano - overnemen en nieuw plaatje curvi

## Hydrodynamics options

Dano

### Stationary mode

### Non-stationary (surfbeat) mode

### Wave resolving mode

## Short wave propagation

### Wave action balance

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



In which the wave action *A* is calculated as:



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



With *uL* and *vL* the cross-shore and alongshore depth-averaged Lagrangian velocities respectively (defined below), and the group velocity cg obtained from linear theory. If wave-current interaction is turned off (keyword: *wci=0*) then the last term in either equation is not taken into account. The propagation speed in θ-space is obtained from:



In *h* represents the total water depth and in this formulation bottom refraction (first term) and wave-current interaction (last two terms) are taken into account. If wave-current interaction is turned off (keyword: *wci=0*) then the last two terms are neglected.

The wave number *k* is obtained from the eikonal equations that is described in . In this formulation the subscripts refer to the direction of the wave vector components and *ω* represents the absolute radial frequency.



The wave number is then obtained from .



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



### Dissipation

The set of equations of the wave action balance closes with dissipation terms. In XBeach there are three dissipative terms: wave breaking, bottom friction and vegetation. Given the spatial distribution of the wave action (and therefore wave energy) the radiation stresses can be evaluated by using linear wave theory as described in:



#### Breaking

There are in four different wave breaking formulations (*Dw*) implemented in XBeach. The formulations are coded with the keyword *break*.

1. Non-stationary waves: formulation of Roelvink (1993a)
2. Stationary waves: formulation of Baldock et al. (1998)
3. Non-stationary waves: adaptation offirst formulation
4. Non-stationary waves: adaptation of first formulation,

according to Daly et al. ,2010

For the non-stationary (surf beat) approach the total wave energy dissipation, i.e. directionally integrated, due to wave breaking is modelled according to Roelvink (1993a). This is coded as *break=roelvink*. In *α* is applied as wave dissipation coefficient of O(1), *Qb* is the fraction breaking waves, *p* stands for the water density and *γ* is the breaker index. The total wave energy *Ew* is calculated by integrating over the wave direction per directional bin.



In variation of , one could also use the third wave breaking formulation, presented in . This formulation is somewhat different than the formulation of Roelvink (1993a). This is coded as *break=roelvink2.*



On top of that, Daly et al. (2010) developed a formulation presented in , which states that waves are fully breaking if the wave height exceeds a threshold (*γ*) and stop breaking if the wave height fall below another threshold (*γ2*). This is coded as *break=roelvink\_daly*.



In the stationary case Baldock et al. (1998) is applied, which is presented in . In this breaking formulation the fraction breaking waves *Qb* and breaking wave height *Hb* is calculated differently compared to the breaking formulations used for the non-stationary situation. In *α* is applied as wave dissipation coefficient, *frep* represents a representative intrinsic frequency and *y* is a calibration factor. The stationary wave breaking formulation is coded with *break=baldock*.



In either the non-stationary or stationary case the total wave dissipation is distributed proportionally over the wave directions with the formulation in .



#### Bottom friction

The bottom friction dissipation is modelled as



In the *fw* is the short-wave friction coefficient. This value only effects the wave action equation and is unrelated to bed friction in the flow equation. Studies conducted on reefs (e.g. Lowe et al., 2007) indicate that fw should be an order of magnitude (or more) larger than *cf*due to the dependency of wave frictional dissipation rates on the frequency of the motion. By default XBeach uses a *fw* value of 0.

#### Vegetation

The presence of aquatic vegetation within the area of wave propagation or wave breaking results in an additional dissipation mechanism for short waves. This is modelled using the approach of Mendez & Losada (2004), which was adjusted by Suzuki et al., (2011) to take into account vertically heterogeneous vegetation, see Van Rooijen et al. (2015). The short wave dissipation due to vegetation is calculated as function of the local wave height and several vegetation parameters. The vegetation can be schematized in a number of vertical elements with each specific properties. In this way the wave damping effect of vegetation such as mangrove trees, with a relatively dense root system but sparse stem area, can be modelled. The dissipation term is then computed as the sum of the dissipation per vegetation layer (Suzuki et al, 2011):



where *Dv,i*is the dissipation by vegetation in vegetation layer *i* and *nv* is the number of vegetation layers. The dissipation per layer is given by:



where is a (bulk) drag coefficient, *bv,i* is the vegetation stem diameter, *Nv,i* is the vegetation density, and *αi* is the relative vegetation height (= *hv*/ *h*) for layer *i*. In case only one vegetation layer is specified, the plants are assumed to be vertically uniform, which would for example typically apply in case of modelling sea grass.

**Damping of IG-waves and mean flow by vegetation**

The presence of aquatic vegetation within the area of wave propagation or wave breaking may not only result in short wave dissipation (section XXX), but also in damping of infragravity waves and/or mean flow. Since both long waves and mean flow are fully resolved with the nonlinear shallow water equations, the effect of vegetation can be modelled using a drag force (e.g. Dalrymple et al., 1984), which can be directly added to the momentum equations (Eq. XX, see also Van Rooijen et al., 2015):



Where *CD* is a drag coefficient, *bv* is the vegetation stem diameter, *N* is the vegetation density and *u* is the wave or current related velocity. To take into account the velocity due to mean flow and infragravity waves, we use the Lagrangian velocity (*uL)*here. The vegetation-induced time varying drag force is then calculated as the sum of the vegetation-induced drag force per vegetation layer:



where is a (bulk) drag coefficient, *bv,i* is the vegetation stem diameter, *Nv,i* is the vegetation density, and *αi* is the relative vegetation height (= *hv*/ *h*) for layer *i*.

### Roller energy balance

Dano

## Shallow water equations

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



In uS and vS represents the Stokes drift in x- and y-direction respectively (Phillips, 1977). The Strokes drift is calculated with in which the wave-group varying short wave energy *Ew* and direction are obtained from the wave-action balance.



The resulting GLM-momentum equations are given by:



In *τsx*and *τsy*are the wind shear stresses*, τbx*and *τby*are the bed shear stresses, *η* is the water level, *F* are the wave-induced stresses, *v* is the horizontal viscosity and *f* is the Coriolis coefficient. The shear stress terms are calculated with the Eulerian velocities as experienced by the bed and not with the GLM velocities. Also, the boundary conditions for the flow computations are expressed in function of Lagrangian and not Eulerian velocities.

### Horizontal viscosity

The horizontal viscosity (*vh*) is by default computed using the Smagorinsky (1963) model to account for the exchange of horizontal momentum at spatial scales smaller than the computational grid size, which is given as:



In *cS* is the Smagorinsky constant, set at 0.1 in all model simulations. It is also possible to use a user-defined value for the horizontal viscosity (keyword *smag = 0*).

### Bed shear stress

The mean current and long wave motions are solved within the depth-averaged momentum and continuity equations. The bed friction associated with mean currents and long wave is included via the formulation of the bed shear stress (*τb*). Using the approach of Ruessink et al. (2001) the bed shear stress is calculated with:



There are 4 different bed friction formulations (keyword: *bedfriccoef*) implemented into XBeach:

1. Chezy value (*C)* *bedfriccoef=chezy*
2. Dimensionless friction coefficient (*cf*) *bedfriccoef=cf*
3. White-Colebrook *bedfriccoef=white-colebrook*
4. Manning *bedfriccoef=manning*
5. White-Colebrook grainsize *bedfriccoef=white-colebrook-grainsize*

Internally XBeach calculates for every cell the dimensionless friction coefficient (*cf*) based on the input given by the user (keyword: *bedfriccoef=’value’*). In this calculation a distinction is made between a friction value in x- and y-direction. It is also possible to define a spatial varying bed friction. The users than needs to define a so called bed friction file (keyword: *bedfricfile=’file.txt’*).

## Nonhydrostatic pressure correction

For non-hydrostatic XBeach calculations (keyword *nonh=1*) depth-averaged flow due to waves and currents are computed using the non-linear shallow water equations, including a non-hydrostatic pressure. The depth-averaged normalized dynamic pressure (*q*) is derived in a method similar to a one-layer version of the SWASH model (Zijlema et al. 2011). The depth averaged dynamic pressure is computed from the mean of the dynamic pressure at the surface and at the bed by assuming the dynamic pressure at the surface to be zero and a linear change over depth. In order to compute the normalized dynamic pressure at the bed, the contributions of advective and diffusive terms to the vertical momentum balance are assumed to be negligible.



In *w* is the vertical velocity and *z* is the vertical coordinate. The vertical velocity at the bed is set by the kinematic boundary condition, in which :



Combining the Keller-box method (Lam and Simpson 1976), as applied by Stelling and Zijlema (2003) for the description of the pressure gradient in the vertical, the dynamic pressure at the bed can be described by:



Substituting in allows the vertical momentum balance at the surface to be described by:



In the subscript *s* refers to the location at the surface. The dynamic pressure at the bed is subsequently solved by combining and the local continuity equation:



Smit et al. (2010) have shown that the inclusion of the dynamic pressure described above reduces the relative dispersion and celerity errors in the non-linear shallow water equations of XBeach to less than 5% for values of kh<2.5 and allows for accurate modelling over wave transformation on dissipative beaches. In order to improve the computed location and magnitude of wave breaking, we apply the hydrostatic front approximation (HFA) of Smit et al. (2013), in which the pressure distribution under breaking bores is assumed to be hydrostatic. Following the recommendations of Smit et al. (2013), we consider hydrostatic bores if  and reform if .

Although this method greatly oversimplifies the complex hydrodynamics of plunging waves on, McCall et al. (2014) shows that the application of this method provides sufficient skill to describe dominant characteristics of the flow, without requiring computationally-expensive high-resolution discretisation of the vertical and surface tracking of overturning waves.

## Groundwater flow

### Hydrostatic

The hydrostatic groundwater module in XBeach utilizes the principle of Darcy flow and is therefore limited to laminar flow conditions. The module includes a vertical interaction flow between the surface water and groundwater. This flow is assumed to be a magnitude smaller than the horizontal flow and is not incorporated in the momentum balance. Darcy flow is described by the following relationship between the groundwater head gradient *dpgw/dx* and *dpgw/dy*, the permeability *k*, and the horizontal velocity, as can be seen in .



#### Determining groundwater head

The driving force behind groundwater flow according to Darcy is the groundwater head gradient. In the XBeach module, the groundwater head *pgw* is expressed in meter and basically there are two possibilities in the model:

1. There is no surface water, than the groundwater head is equal to the groundwater surface level *ηgw*
2. There is surface water and the groundwater surface level is just below the surface of the bed *zb*. This means the groundwater head is affected by the surface water head *zs*
   * 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 *dwetlayer* 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*.

#### 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 of m/s and is defined positive from the surface water to ground water and is given in terms of surface water for the continuity equation (i.e. 100% porosity).

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) exceeding the bed level is joins the surface water within the same numerical time step. The vertical velocity can therefore be calculated by:



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.



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*. 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 (*zs-zb*).

The thickness of the infiltration layer (*dinfiltration*) 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.



For numerical stability, the infiltration layer thickness is restricted to a minimum of one third of (*dinfiltration*), 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.

#### Mass balance

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



The effective depths through which horizontal ground water flow takes place (*hugw, hvgw*), are found by taking the mean difference between the groundwater level and bed of the aquifer (*zb,acquifir*) in the two surrounding points. 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*).

#### 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. The bed of the aquifer is set equal to or less than the regular bed level.
* At the offshore boundary: the groundwater head is set equal to the offshore surface water head.
* Bay side conditions: for cases in which a bay side water level is given explicitly with a tidal level record, the groundwater head on the bay side boundary is set equal to the bay side surface water head. In all other cases, the bay side groundwater head is kept at the initial value.
* Lateral boundary conditions: Neumann boundary conditions are applied to the groundwater head on the lateral boundaries:

The initial groundwater level is calculated from the initial groundwater head. The bed of the aquifer and the initial groundwater head must be specified.

### Non-hydrostatic

Groundwater flow in the swash and surf zone has been shown in previous numerical model (e.g., Li and Barry 2000; Lee et al. 2007) studies to be non-hydrostatic. Therefore, a requirement of the groundwater model is that it does not use the Dupuit–Forchheimer assumption of hydrostatic groundwater pressure. Although the requirement for non-hydrostatic pressure has the benefit of being a more accurate representation of reality, resolving the non-hydrostatic pressure field can be very computationally expensive.

In order to allow for a computationally efficient approximation of the non-hydrostatic groundwater pressure field, XBeach applies a quasi 3D-method to predict depth-averaged horizontal groundwater fluxes, vertical distribution of the groundwater pressure and the flow driven by groundwater-surface water pressure gradients (submarine exchange).

#### Equation of motions

Laminar flow of an incompressible fluid through a homogeneous medium can be described using the well-known Law of Darcy (1856). In which *K* is the hydraulic conductivity of the medium and *H* is the hydraulic head. However, in situations in which flow is not laminar, turbulent and inertial terms may become important, this relation is no longer valid. In these cases groundwater flow should be described using the extended Forchheimer equation.

In XBeach, however, a method comparable with the USGS MODFLOW-2005 groundwater model (Harbaugh 2005), in which the turbulent hydraulic conductivity is estimated based on the laminar hydraulic conductivity (*Klam*) and the Reynolds number at the start of turbulence (*Recrit*) (Halford 2000)



In the Reynolds number (*Re*) is calculated using the median grain size (*D5*0), the kinematic viscosity of water (*n*) and the groundwater velocity in the pores. Since the hydraulic conductivity in the turbulent regime is dependent on the local velocity, an iterative approach is taken to find the correct hydraulic conductivity and velocity.

#### Vertical groundwater head approximation

Since XBeach is depth-averaged, the model cannot compute true vertical profiles of the groundwater head and velocity. In order to improve the estimate of the groundwater head variation over the vertical, a quasi-3D modelling approach is applied, which is set by three conditions:

1. There is no exchange of groundwater between the aquifer and the impermeable layer below the aquifer
2. The groundwater head at the upper surface of the groundwater is continuous with the head applied at the surface
3. The vertical velocity is assumed to increase or decrease linearly from the bottom of the aquifer to the upper surface of the groundwater:

The vertical groundwater head approximation can be solved for the three imposed conditions by a parabolic function. The depth-average value of the groundwater head is used to calculate the horizontal groundwater flux and is found by integrating the groundwater head approximation over the vertical:



In the mean vertical ground water head (*H*) is calculated using the groundwater head imposed at the groundwater surface (*Hbc*), the groundwater head parabolic curvature coefficient (*β*) and the height of the groundwater level above the bottom of the aquifer (*hgw*).

#### Exchange with surface water

In the groundwater model there are three mechanisms for the exchange of groundwater and surface water: 1) submarine exchange, 2) infiltration and 3) exfiltration. The rate of exchange between the groundwater and surface water (*S*) is given in terms of surface water volume, and is defined positive when water is exchanged from the surface water to the groundwater. The groundwater and surface water are said to be in a connected state where and when the groundwater level reaches to the top of the bed and surface water exists above the bed. This state is described by a spatially and temporally varying logical *k*, which is true where groundwater and surface water are connected and false in all other situations.

Submarine exchange represents the high and low frequency infiltration and exfiltration through the bed due pressure gradients across the saturated bed. This process only takes place where the groundwater and surface water are connected. The rate of submarine exchange is determined by the vertical specific discharge velocity at the interface between the groundwater and surface water. The value of this velocity can be found using the vertical derivative of the approximated groundwater head at the groundwater-surface water interface.



Infiltration and exfiltration can only occur in locations where the groundwater and surface water are not connected. Infiltration takes place when surface water covers an area in which the groundwater level is lower than the bed level. The flux of surface water into the bed is related to the pressure gradient across the wetting front.



In the surface water-groundwater exchange flow of infiltration (*Sinf*) is calculated using the effective hydraulic conductivity (*K*), the surface water pressure at the bed () and a thickness of the wetting point (**).

Since the groundwater model is depth-averaged and cannot track multiple layers of groundwater infiltrating into the bed, the wetting front thickness is reset to zero when there is no available surface water, the groundwater exceeds the surface of the bed, or the groundwater and the surface water become connected. In addition, all infiltrating surface water is instantaneously added to the groundwater volume, independent of the distance from the bed to the groundwater table. Since the groundwater model neglects the time lag between infiltration at the beach surface and connection with the groundwater table a phase error may occur in the groundwater response to swash dynamics

Exfiltration occurs where the groundwater and surface water are not connected and the groundwater level exceeds the bed level (Figure 3.4b). The rate of exfiltration is related to the rate of the groundwater level exceeding the bed level.



#### Calculation of groundwater and surface water levels

The curvature coefficient (*β*) in the vertical groundwater head approximation is solved using the coupled equations for continuity and motion, thereby producing the depth-average horizontal groundwater head gradients and vertical head gradients at the groundwater surface, and subsequent depth-average horizontal and vertical specific discharge.

In areas where the groundwater and surface water are not connected, the groundwater level change is related to the vertical specific discharge and the infiltration and exfiltration fluxes:



In these same areas the surface water level is modified to account for infiltration and exfiltration:



In areas where the groundwater and surface water are connected, the groundwater level remains at the level of the bed, since the computed vertical velocity at the surface (w) is exactly equal and opposite to the submarine exchange (Ssub). The surface water level is modified to account for the submarine exchange with the groundwater:



In cases where there is not sufficient surface water to permeate into the bed to ensure the groundwater level remains at the bed level, a fractional time step approach is taken in which the area is considered to be connected while there is sufficient surface water, and considered unconnected once the surface water has drained away. A similar approach is taken when the groundwater level reaches the bed level during an infiltration event.

#### Boundary conditions

Since the groundwater dynamics are described by a parabolic equation, the system of equations requires boundary conditions at all horizontal and vertical boundaries, as well as an initial condition

* At the horizontal boundaries and bottom of the aquifer: a zero flux condition is imposed. This is based on the assumption that the groundwater head is constant
* At the surface of the groundwater
  + If connected: the head is set to the surface water head at the bed
  + If not-connected: the head is equal to the atmospheric pressure head
* The initial condition for the solution is specified by the model user in terms of the initial groundwater head.

## Sediment transport

### Advection-diffusion

Sediment concentrations in the water column are modelled using a depth-averaged advection-diffusion scheme with a source-sink term based on an equilibrium sediment concentration (Galappatti and Vreugdenhi, 1985):



In *C* represents the depth-averaged sediment concentration which varies on the wave-group time scale and *Dh* is the sediment diffusion coefficient. The entrainment of the sediment is represented by an adaptation time *Ts*, given by a simple approximation based on the local water depth *h* and sediment fall velocity *ws*. A small value of *Ts* corresponds to nearly instantaneous sediment response.



The entrainment or deposition of sediment is determined by the mismatch between the actual sediment concentration *C* and the equilibrium concentration *Ceq* thus representing the source term in the sediment transport equation.

### General parameters

In transport formulations the equilibrium sediment concentration *Ceq*(for both the bed load and the suspended load) is related to the velocity magnitude (*vmg*) and the orbital velocity (*urms*). This section elaborates how we calculate both terms.

First of all the velocity magnitude, if long wave stirring is turned on (keyword: *lws=1*), the velocity magnitude *vmg* is equal to the magnitude of the Eulerian velocity, as can be seen in .



If wave stirring is turned off (*keyword: lws=0*), the velocity magnitude will be current-averaged on time scale based on a certain factor *fcats* of the representative wave period *Trep*. By default *fcats* is 4.



Secondly, the root-mean-squared velocity, the urms is obtained from the wave group varying wave energy using linear wave theory. This formulation can be found in .



To take into account for wave breaking induced turbulence due to short waves, the orbital velocity is adjusted (van Thiel de Vries, 2009). In this formulation *kb* is the wave breaking induced turbulence due short waves. The turbulence is approximated with an empirical formulation in XBeach.



### Transport formulations

In the present version of XBeach, two sediment transport formulations are available. The formulae of the two formulations are presented in the following sections. For both methods the total equilibrium sediment concentration is calculated with .



#### Soulsby-Van Rijn

The Soulsby-Van Rijn transport equations are known as (Soulsby, 1997; van Rijn, 1984):



For which the bed-load and suspended load coefficient are calculated with:



In which the dimensionless sediment diameter (D\*) can be calculated with the following formulation:



The critical velocity defines at which depth averaged velocity sediment motion is initiated:



Finally the drag coefficient is calculated with:





#### Van Thiel-Van Rijn

The Van Thiel-Van Rijn transport equations are known as (van Rijn, 2007; van Thiel de Vries, 2009):



For which the bed-load and suspended load coefficient are calculated with:



The critical velocity is computed as weighted summation of the separate contributions by currents and waves (Van Rijn, 2007).



The critical velocity for currents is based on Shields (1936)



The critical velocity for waves is based on Komer and Miller (1975)



### Wave asymmetry

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



XBeach considers the wave energy of short waves as averaged over their length, and hence does not simulate the wave shape. A discretization of the wave skewness and asymmetry was introduced by Van Thiel de Vries (2009), to affect the sediment advection velocity. In this equation *ua* is calculated as function of wave skewness (*Sk*), wave asymmetry parameter (*Sk*), root-mean square velocity *urms* and a calibration factor *fua* (keyword: *facua*).



The skewness and asymmetry as parameterized as a function of the Ursell number by Ruessink et al. (2012).



## Bottom updating

### Due to sediment fluxes

Based on the gradients in the sediment transport the bed level changes according to:



In *ρ* is the porosity, *fmor* is a morphological acceleration factor of O(1-10) (Reniers et al., 2004) and *qx* and *qy* represent the sediment transport rates in x- and y-direction respectively. In order to take account for bed-slope effects on sediment transport a bed-slope correction factor *fslope* is introduced.



### 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 via the use of a critical bed slope for both the dry and wet area (keyword: *wetslp* and *dryslp*). It is considered that inundated areas are much more prone to slumping and therefore two separate critical slopes for dry and wet points are used. The default values are 1 and 0.3 respectively. When this critical slope is exceeded, material is exchanged between the adjacent cells to the amount needed to bring the slope back to the critical slope.



The change of the bed level within one time step is then given by . In this formulation a threshold of 0.05 m/s has been introduced to prevent the generation of large shockwaves.



### Bed composition

If the effect of different sediment fractions, sorting and armouring are of importance, a bed composition constituting multiple sediment fractions can be defined. Each sediment fraction is characterized by a median grain size (D50) and possible a D15 and D90 as well. When using multiple sediment fractions, multiple bed layers are needed as well to describe the vertical distribution of the sediment fractions in the bed.

By specifying multiple bed layers, XBeach can keep track of the different sediment fractions both in the horizontal and in the vertical. Coarse sediments may be deposited on top of fine sediment after which erosion of the coarse sediment is needed to expose the fine sediment again, effectively armouring the bed. Three types of bed layers are distinguished: 1) the top layer 2) the variable or “breathing” layer and 3) the bottom layers. The top layer is the only layer that interacts with the water column and can be eroded, but preserves it thickness. The bottom layers are layers of constant thickness that move with the top layer. A single variable or “breathing” layer is defined that adapts its thickness to the erosion and sedimentation of the bed. At least one of each type of bed layer is needed, which makes that at least three bed layers are required. For example: if a grid cell is eroded, particular fractions of sediment are removed from the top layer, but the top layer preserves its thickness and thus it takes the same volume of sediment, likely of different composition than the eroded sediment, from the layer below. If this layer is a bottom layer, the thickness is preserved and again the same volume of sediment is taken form a lower bed layer. This continues until the variable or “breathing” layer is reached. This layer adapts its thickness to the amount of erosion. If the thickness of the layer becomes too small, the variable layer is merged with an adjacent bottom layer and a new bottom layer is defined underneath the existing ones to ensure a constant number of bed layers. Reversely, if a grid cell is accreting, the thickness of the variable layer will be increased and with sufficient increase the variable layer will be split in two effectively creating a new bottom layer. The lowest existing bottom layer is then discarded to ensure a constant number of bed layers.

Each grid cell in XBeach holds its own sediment distribution and the sediment transport formulations used differentiate between fractions. Therefore the distribution of sediment may change over time and processes like armouring and sorting can be simulated. Due to the shifting of sediment between bed layers numerical mixing of sediment occurs. Choosing bed layer thicknesses that are in balance with the expected erosion and deposition during the simulation should keep the numerical mixing to a minimum. A bed layer thickness that is too large will result in relatively uniform behaviour, while a bed layer thickness that is too small will result in a lot of shifting and thus numerical mixing.

# Boundary conditions

## Waves

XBeach allows users to include two different options for wave boundary conditions in the model. These wave boundary conditions can be applied only at the upwave (seaward) boundary (keyword: *instat*). First of all, in 3.1.1 the method to specify wave spectra is discussed. Secondly, in 3.1.2 the method to apply non-spectra, such as stationary wave conditions or time-series is elaborated. In 3.1.3 the lateral boundary conditions for waves are discussed. There is currently not a possibility to force waves on the downwave (landward) boundary of a model.

### Spectra

The most-used wave boundary condition is a spectral type. XBeach allows the user to define these with three possibilities:

1. Parameterized spectrum: In this case the user can define a parameterized spectrum (typically of the JONSWAP or Pierson-Moscowitz type) in terms of the parameters such as *Hm0* or *fp*. (keyword *instat=jons*)
2. SWAN spectrum input: In this case the two-dimensional (frequency-direction) output by the spectral wave model SWAN ( .sp2 files) can be specified. (keyword *instat=swan*)
3. Formatted variance density spectrum: In this case a more general type (typically measured) spectrum can be specified. (keyword *instat=vardens*)

Upon specification of these keywords, XBeach will read the spectrum or spectral parameters from a separate file specified (keyword *bcfile= 'file.txt'*). The user must also state in params.txt the required duration for the boundary condition file (keyword *rt=<number>*) and the boundary condition file time step (keyword *dtbc=<number>*). If the record length 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. The default file time step is 0.5 s.

For option 2, 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 need to be specified (keyword *dthetaS\_XB=<number>*).

For the user-defined spectrum (option 3) it is important to note that the angles in the input file must be in the 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 in the file must be increasing

For all three options, XBeach offers the possibility of both a time-varying (*filelist*) and space-varying spectra (*loclist*). With *filelist* it is possible to define a sequence of sea-states, while with *loclist* it is possible to define the spectrum location containing the world x-coordinate and world y-coordinate of the location that the input spectrum should apply. More information about the input description can be found in @.

For option 1 there is an extra possibility for time-varying wave group conditions (keyword *instat=jons\_table*). In this case, each line in the spectrum definition file contains a parametric definition of a spectrum, like in a regular JONSWAP definition file, plus the duration for which that spectrum is used during the simulation.

If the user does not wish to recalculate boundary condition files or specifically wants to reuse the boundary condition files of another XBeach simulation, the keyword: *instat=reuse* should be used. No further wave boundary condition data need be given. 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.

### Non-spectra

XBeach also allows the user to define non-spectral wave boundary conditions. There are 4 possibilities to do so:

1. Stationary wave boundary condition. This means that a uniform and constant wave energy is specified, based on the given values of Hrms, Tm01, direction and power of the directional distribution function. Here there are multiple options
   1. Specify a single sea state without wave groups (keyword *instat=stat*)
   2. Specify a series of sea states without wave groups (keyword *instat=stat\_table*)
   3. Bichromatic (two short-wave components) waves (keyword *instat=bichrom*). In this case, XBeach will be forced with regular wave groups as the two short-wave components force one difference (infragravity) wave period. The user needs to specify not only variables of the stationary situation but also a wave period for the long wave. This wave period will be used to calculate the long wave based on the theory of Longuet-Higgins and Stewart (1964).
2. Time series of waves. The user can specify the variation in time of the wave energy.
   1. First-order time series of waves (keyword *instat=ts\_1*). XBeach will calculate the bound long wave based on the theory of Longuet-Higgins and Stewart (1964).
   2. Second-order time series of waves (keyword *instat=ts\_2*). The bound long wave is specified by the user via a long wave elevation.
3. Boundary conditions for non-hydrostatic model (keyword *instat=ts\_nonh*)

Specify the variation in time of the horizontal velocity, vertical velocity and the free surface elevation. Last two terms are optional.

1. No wave boundary conditions (keyword *instat=off)*

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

For option 1 there is an extra possibility for time-varying stationair conditions / sea states (keyword *instat=stat\_table*). In this case, each line in the spectrum definition file contains a parametric definition of a spectrum, like in a regular JONSWAP definition file, plus the duration for which that spectrum is used during the simulation.

If the user does not wish to recalculate boundary condition files or specifically wants to reuse the boundary condition files of another XBeach simulation, the keyword: *instat=reuse* should be used. No further wave boundary condition data need be given. 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.

### Lateral boundary conditions

Dano

## Shallow water equations

### Absorbing-generating

Typically, an offshore or lateral 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 (absorbing-generating), but there are also other possibilities implemented in XBeach (keyword: *front*). This method can be applied in 1D or 2D, is recommended and therefore the default value for XBeach.

In XBeach, there are two options with regard to the offshore absorbing-generating boundary condition. With the parameter setting *front=abs1d* a simple one-dimensional absorbing-generating boundary condition is activated. This option allows for a time-varying water level (surge and/or infragravity waves) to be specified at the boundary while allowing any waves propagating perpendicularly towards the boundary to be absorbed (i.e., passed through the boundary with a minimum of reflection. It is therefore only useful for 1D (flume like) simulations.

With option *front=abs2d* (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 obliquely-reflected waves to pass through the boundary. It is possible to account for situations with boundary-perpendicular and boundary-parallel currents. In order to differentiate between the particle velocities, the keyword *epsi* must be set. This parameter controls a simple Kalman-update filter which controls which part of the particle velocity is assumed to be part of the current and which part is wave-related. By default XBeach computes the value for epsi automatically using offshore boundary conditions (keyword: *epsi=-1*).

This option is the preferred one for 2D computations. For details on the formulation of the absorbing-generating boundary condition by Van Dongeren and Svendsen (1997).

There are three other possibilities implemented besides the absorbing-generating boundary conditions:

1. No flux wall (keyword: *front=wall*). This boundary condition type is a simple no flux boundary condition.
2. Water level specification (keyword: *front=wlevel*). This boundary sets the water level at a prescribed value. This can be constant or time-varying. With this option the outgoing long waves are not absorbed.
3. Boundary condition for the non-hydrostatic option (keyword*: front=nonh\_1d*). The user needs to provide a file containing time series for the velocity at the boundary.
4. Radiation boundary condition (keyword: *front=waveflume*).

### River and point discharge

The effect of a river outflow or other discharges can be simulated with XBeach. Multiple discharge locations can be designated. At a discharge location the discharge orifice is defined as well as the discharge time series in m3/s. The discharge orifice always constitutes an uninterrupted series of full grid abreast cell borders. It is not possible to define a discharge over half a grid cell nor is it possible to define a single discharge through grid cell borders that are either not adjacent or are not abreast.

At each time step the model sets the discharge and velocities at the grid cell borders that constitute the discharge orifice, which can be computed given the size of the discharge orifice and discharge time series. The discharge is positive in positive x or y direction. An exception is made when discharges are defined at the domain border. In that case the discharge is positive towards the domain (influx).

When a discharge is defined with a zero size orifice the discharge is assumed to be in vertical direction where a positive discharge is into the domain (influx). In these cases the discharge is linked to the closest grid cell center and at each time step mass according to the discharge time series is added. No momentum is added in case of a vertical discharge.

### Ship motion

Dano

### Lateral boundaries

Lateral boundaries are the boundaries perpendicular to the coastline. Usually these are artificial, because the model domain is limited but the physical coast will continue. At these boundaries (keywords: *left* & *right*) we need to prescribe information about the area beyond the numerical model domain in such a way that the boundary condition does not influence the results in an adverse way. One way to do this is to prescribe a so-called “no-gradient” or Neumann boundaries (XBeach default), which state that there is locally no change in surface elevation and velocity, but there are also other possibilities implemented into XBeach. This method is recommended and is therefore the default value for XBeach. Each lateral boundary is a separate condition, so it is possible to mix different type of lateral boundary per side.

Neumann boundary conditions are activated 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 when a shadow zone is taken into account. This means that regions where the boundary conditions are not fully enforced the results are not taken into account. Neumann boundaries can be individually defined (keyword: *left=neumann).*

There are two other possibilities implemented besides the absorbing-generating boundary conditions:

1. Simple no-flux boundary conditions can also be applied (keyword: *left=wall*). Wall boundary conditions are preferred over Neumann boundary conditions in 1D (cross-shore) models.
2. Velocity at the boundary will be calculated from NLSWE, but only include the advective terms. (keywords: *left=no\_advec*)

### Tide and surge

XBeach can take in up to four time-vary tidal signals to be applied to the four boundaries (offshore-left, backshore-left, backshore-right, offshore-right). A time-varying water level signal is read into XBeach by reading the specified file in zs0file. 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).

There are now four options for handling the tidal and/or surge contribution to the boundaries:

* Uniform water level (keyword: *tideloc=0*)
* One time-varying water level signal (keyword: *tideloc=1*)
* Two time-varying water level signals, which requires point of application indication. (keyword: *tideloc=2*)
* Four time-varying water level signals (keyword: *tideloc=4*)

For the option with a uniform water level the value specified in the params.txt is applied in the complete model domain (keyword: *zs0=’value’*). For the option with one time-varying water level signal the specified water level is applied (keyword: *zs0file = name\_of\_your\_time\_serie*) to the offshore boundary and a fixed value is applied at the backshore boundary (keyword: *zs0=value*). For the option with two time-varying water level signals two water level signals are read from the zs0file. Note: one tidal record is applied to both sea corners and one tidal record to both land corners. This means there is no alongshore variation. An alongshore variation can be applied when applying four time-varying water level signals.

## Sediment transport

Dano

# Input description

## General

Upon execution of the XBeach executable *xbeach.exe* the file *params.txt* in the current working directory will be read. The *params.txt* file contains keyword/value pairs that determine the parameter settings of XBeach. Each keyword/value pair may contain an actual model parameter or refers to another file with additional information on the model setup. If a *params.txt* file cannot be found then the execution of XBeach will be aborted.

In the *params.txt* file there can be a single keyword/value pair per line in any order. A keyword/value pair is separated by an equal sign (=). Each line containing an equal sign is interpreted as a keyword/value pair. Reversely, any lines without an equal sign are ignored and may be used for comments. Only a few keywords are required, others have default values that are used in case the keyword is not mentioned in the *params.txt* file.

The *params.txt* file contains grid and bathymetry info, wave input, flow input, morphological input, et cetera. This chapter describes the possibilities of the *params.*txt file and any files that can be referred to from the *params.txt* file. The tables in this chapter contain a description of the keywords, the default values, its units and recommended value ranges, while the formats for additional input files are described in the relevant sections. Keywords marked with an asterix (\*) are considered advanced options and should not be used for regular applications of XBeach.

In this chapter, any references to keywords refer to keywords in the *params.txt* file unless stated otherwise. Also any references to time indications are in seconds unless stated otherwise.

A typical *params.txt* file for a 1D XBeach model is:

params.txt

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% XBeach parameter settings input file %%%

%%% %%%

%%% date: 01-Jan-2015 12:00 %%%

%%% function: xb\_write\_params %%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Bed composition parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

D50 = 0.000105

%%% Grid parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

depfile = bed.dep

posdwn = 0

nx = 265

ny = 0

alfa = 0

vardx = 1

xfile = x.grd

yfile = y.grd

thetamin = -90

thetamax = 90

dtheta = 15

thetanaut = 0

%%% Model time %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tstop = 3600

%%% Physical constants %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

rho = 1000

depthscale = 1

%%% Tide boundary conditions %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

zs0file = tide.txt

tideloc = 2

%%% Wave boundary condition parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

instat = jons

%%% Wave-spectrum boundary condition parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%

bcfile = filelist.txt

random = 0

%%% Output variables %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

outputformat = netcdf

tint = 3600

tstart = 0

nglobalvar = 3

zb

zs

H

## Physical processes

XBeach supports a variety of physical processes from generic, like waves and flow, to very specific, like ship motions and point discharge. Each process can be switched on or off. The commonly used processes are turned on by default. The table below lists the keywords used to switch on or off physical processes in XBeach.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| avalanching | Turn on avalanching | 1 | 0 - 1 | - |  |
| bchwiz | Turn on beachwizard | 0 | 0 - 1 | - |  |
| flow | Turn on flow calculation | 1 | 0 - 1 | - |  |
| gwflow\* | Turn on groundwater flow | 0 | 0 - 1 | - |  |
| lwave | Turn on short wave forcing on NLSW equations and boundary conditions | 1 | 0 - 1 | - |  |
| morphology | Turn on morphology | 1 | 0 - 1 | - |  |
| nonh\* | Turn on non-hydrostatic pressure: 0 = NSWE, 1 = NSW + non-hydrostatic pressure compensation Stelling & Zijlema, 2003 | 0 | 0 - 1 | - |  |
| q3d\* | Turn on quasi-3D sediment transport | 0 | 0 - 1 | - |  |
| sedtrans | Turn on sediment transport | 1 | 0 - 1 | - |  |
| setbathy | Turn on timeseries of prescribed bathy input | 0 | 0 - 1 | - |  |
| ships\* | Turn on ship waves | 0 | 0 - 1 | - |  |
| single\_dir\* | Turn on stationary model for refraction, surfbeat based on mean direction | 0 | 0 - 1 | - |  |
| snells\* | Turn on Snell's law for wave refraction | 0 | 0 - 1 | - |  |
| swave | Turn on short waves | 1 | 0 - 1 | - |  |
| swrunup\* | Turn on short wave runup | 0 | 0 - 1 | - |  |
| vegetation\* | Turn on interaction of waves and flow with vegetation | 0 | 0 - 1 | - |  |

## Grid and bathymetry

XBeach’ spatial grid size is defined by the keywords *nx* and *ny*. The size of the computational grid will be *nx+1* by *ny+1* cells large. The initial bathymetry is provided using a separate file that is referred to by the *depfile* keyword. This file contains an initial bed level for each grid cell where each line corresponds to a transect in x-direction (cross-shore). The values are positive down by default, but this can be changed using the *posdwn* keyword.

Three main types of XBeach grids are supported: fast 1D, 1D and 2DH. Fast 1D grids have a single alongshore grid cell and thus a value *ny=0* and a single line in the *depfile*. The 1D grids have 3 alongshore grid cells and thus a value *ny=2* and three lines in the *depfile*. The 2DH grids have more than 3 alongshore grid cells, a value *ny>2* and as may lines in the *depfile*. In general, the bathymetry file has the following format:

bed.dep

<z 1,1> <z 2,1> <z 3,1> ... <z nx,1> <z nx+1,1>

<z 1,2> <z 2,2> <z 3,2> ... <z nx,2> <z nx+1,2>

<z 1,3> <z 2,3> <z 3,3> ... <z nx,3> <z nx+1,3>

...

<z 1,ny> <z 2,ny> <z 3,ny> ... <z nx,ny> <z nx+1,ny>

<z 1,ny+1> <z 2,ny+1> <z 3,ny+1> ... <z nx,ny+1> <z nx+1,ny+1>

XBeach spatial grids can be equidistant or non-equidistant. In the former case the grid size is defined by the keywords *dx* and *dy*. In the latter case the keyword *vardx* should be set to *1* and x- and y-coordinates of the grid cells should be provided through the files referenced by the *xfile* and *yfile* keywords. These files take exactly the same format as the *depfile* file where all coordinates along the x-direction are on one line and each line represents a cell in y-direction. XBeach grids are defined in a coordinate system of choice and can be either rectangular or curvilinear grids.

Delft3D grids created with tools like RFGRID are also supported. To use Delft3D grids, choose *gridform=delft3d* and provide a grid file via the keyword *xyfile*. The format of Delft3D grids is not described here, but can be found in the Delft3D manual. Also forced updating of bathymetries is supported as described in section 6.2.11 Bed update.

Apart for the spatial grid, XBeach also uses a directional grid for short waves and rollers. The grid is determined by a minimum and maximum angle and a directional bin size using the keywords *thetamin*, *thetamax* and *dtheta* respectively. The *thetamin* and *thetamax* angles are either defined according to the Cartesian convention (angle w.r.t. the computational x-axis) or according to the nautical convention (angle w.r.t. deg. N, so from W is 270 deg. N). The convention is chosen using the keyword *thetanaut*.

Examples of typical input for a non-equidistant, fast 1D XBeach model, together with the *params.txt* example at the start of this chapter, are:

bed.dep

0.00 0.09 0.46 0.80 1.14 1.46 1.77 ... 29.14 29.12 29.10 29.07 29.06 29.05

x.grd

0.00 17.38 34.77 52.15 69.54 ... 1403.32 1407.88 1412.44 1417.00 1421.56

y.grd

0.00 0.00 0.00 0.00 0.00 0.00 0.00 ... 0.00 0.00 0.00 0.00 0.00 0.00 0.00

All keywords related to grid and bathymetry input are listed in the following table:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| alfa | Angle of x-axis from East | 0.0 | 0.0 - 360.0 | deg |  |
| depfile | Name of the input bathymetry file |  |  | <file> |  |
| dtheta | Directional resolution | 10.0 | 0.1 - 20.0 | deg |  |
| dtheta\_s | Directional resolution in case of stationary refraction | 10.0 | 0.1 - 20.0 | deg |  |
| dx | Regular grid spacing in x-direction | -1.0 | 0.0 - 1000000000.0 | m |  |
| dy | Regular grid spacing in y-direction | -1.0 | 0.0 - 1000000000.0 | m |  |
| gridform | Grid definition format | xbeach | xbeach, delft3d |  |  |
| nx | Number of computational cell corners in x-direction | 50 | 2 - 10000 | - |  |
| ny | Number of computational cell corners in y-direction | 2 | 0 - 10000 | - |  |
| posdwn | Bathymetry is specified positive down (1) or positive up (-1) | 1.0 | -1.0 - 1.0 | - |  |
| thetamax | Higher directional limit (angle w.r.t computational x-axis) | 90.0 | -180.0 - 180.0 | deg |  |
| thetamin | Lower directional limit (angle w.r.t computational x-axis) | -90.0 | -180.0 - 180.0 | deg |  |
| thetanaut | Switch to specify thetamin and thetamax in nautical convention rather than cartesian | 0 | 0 - 1 | - |  |
| vardx | Switch for variable grid spacing | 0 | 0 - 1 | - |  |
| xfile | Name of the file containing x-coordinates of the calculation grid |  |  | <file> |  |
| xori | X-coordinate of origin of axis | 0.0 | -100000000.0 - 1000000000.0 | m |  |
| xyfile | Name of the file containing Delft3D xy-coordinates of the calculation grid |  |  | <file> |  |
| yfile | Name of the file containing y-coordinates of the calculation grid |  |  | <file> |  |
| yori | Y-coordinate of origin of axis | 0.0 | -1000000000.0 - 1000000000.0 | m |  |

## Waves input

An XBeach model is generally forced by waves on its offshore boundary. These waves are described by the wave boundary conditions discussed in this section. The details of the wave motions within the model are described by the wave numerics in terms of the wave action balance (see 6.2.1 Wave numerics) and wave-current interaction (6.2.4 Wave-current interaction), wave dissipation model (see 6.2.2 Wave dissipation) and wave roller model (6.2.3 Rollers).

XBeach supports a variety of wave boundary condition types that are divided in two main groups: stationary and spectral boundary conditions. The *instat* keyword can be used to select one particular type of wave boundary conditions. Table XXX gives an overview of all types of wave boundary conditions available for XBeach. Figure XXX can be used to help you determine what type of wave boundary conditions is appropriate for your case. Each wave boundary condition type is explained in the following subsections. Note that most spectral wave boundary conditions can vary both in space and time using a *FILELIST* and/or *LOCLIST* construction as described in 4.4.5 Temporally and/or spatially varying wave boundary conditions.

Table XXX Overview of wave boundary conditions supported by XBeach

|  |  |
| --- | --- |
| *instat* | description |
| off | no wave boundary condition |
| stat | stationary wave boundary condition (sea state) |
| bichrom | bichromatic (two wave component) waves |
| ts\_1 | first-order timeseries of waves (generated outside XBeach) |
| ts\_2 | second-order timeseries of waves (generated outside XBeach) |
| jons | wave groups generated using a parametric (Jonswap) spectrum |
| swan | wave groups generated using a SWAN 2D output file |
| vardens | wave groups generated using a formatted file |
| nonh | boundary conditions for nonhydrostatic option |
| reuse | reuse of wave conditions |
| stat\_table | a sequence of stationary conditions (sea states) |
| jons\_table | a sequence of time-varying wave groups |

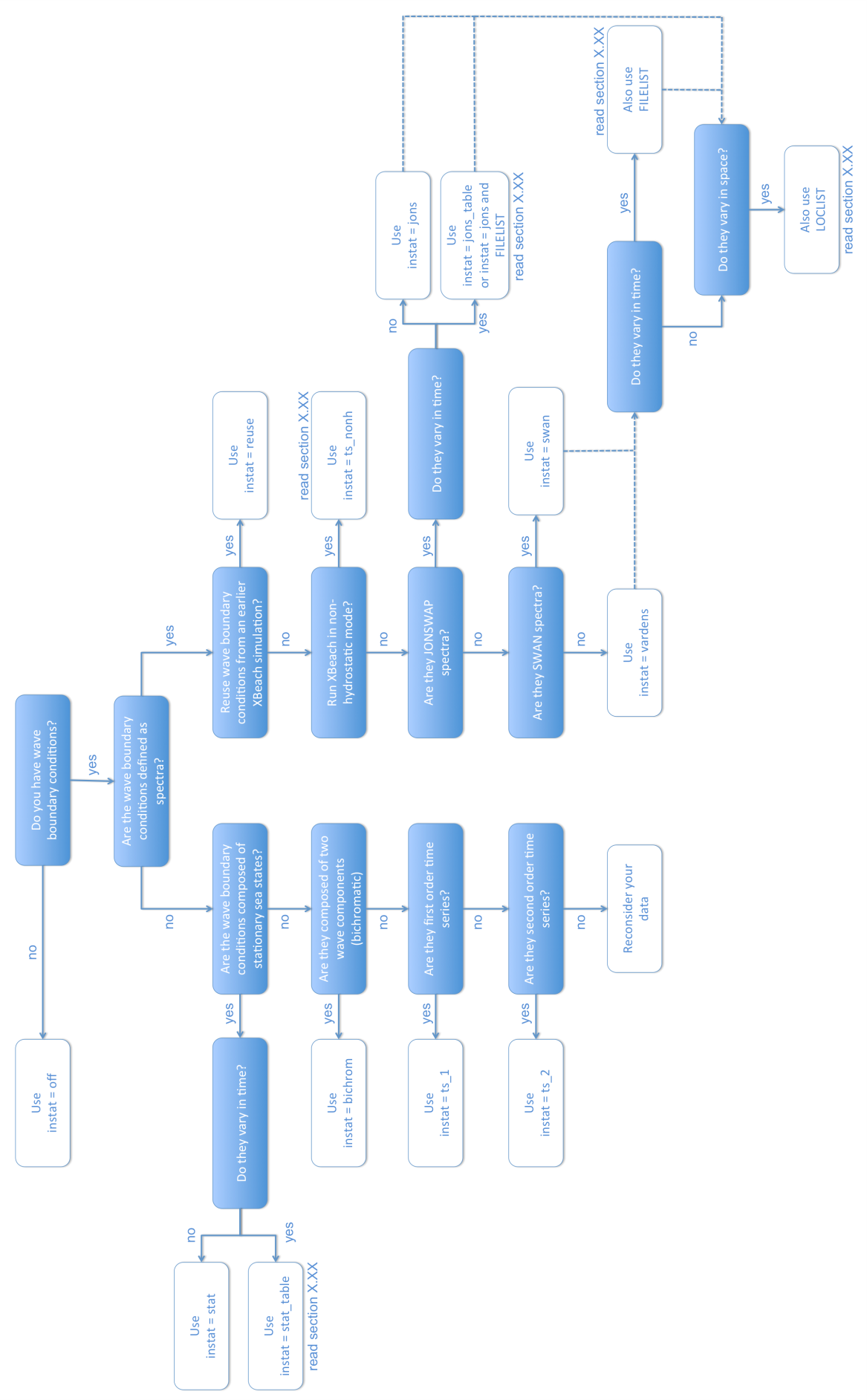


Figure XXX Decision tree for selecting the appropriate type of wave boundary conditions

### Stationary wave boundary conditions

Stationary wave boundary conditions are enabled using *instat* values *stat*, *bichrom*, *ts\_1*, *ts\_2* or *stat\_table*. The conditions aregenerally defined within the *params.txt* file directly using the keywords described in the table below. In addition, in case of *instat* values *ts\_1* or *ts\_2* the file *bc/gen.ezs* should be present that describes the infragravity wave forcing generated outside of XBeach. The format of this file is as follows:

bc/gen.ezs

<time 1> <zs 1> <Hrms 1>

<time 1> <zs 2> <Hrms 2>

<time 2> <zs 3> <Hrms 3>

...

Only in case of *instat=stat\_table* the time-varying stationary wave boundary conditions are fully described in an external file referenced by the *bcfile* keyword. The *bcfile* keyword is part of the spectral wave boundary condition input and also the referenced file is designed for time-varying spectral input in the form of JONSWAP spectra. In stationary mode only the relevant data from this file is used and irrelevant data like *gamma* and *dfj* are discarded. See 4.4.2.1 JONSWAP wave spectra for a description of the file format.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| Hrms | Hrms wave height for instat = stat, bichrom, ts\_1 or ts\_2 | 1.0 | 0.0 - 10.0 | m |  |
| Tlong | Wave group period for case instat = bichrom | 80.0 | 20.0 - 300.0 | s |  |
| Trep | Representative wave period for instat = stat, bichrom, ts\_1 or ts\_2 | 10.0 | 1.0 - 20.0 | s |  |
| dir0 | Mean wave direction for instat = stat, bichrom, ts\_1 or ts\_2 (nautical convention) | 270.0 | 180.0 - 360.0 | deg |  |
| instat | Wave boundary condition type | bichrom | stat, bichrom, ts\_1, ts\_2, jons, swan, vardens, reuse, ts\_nonh, off, stat\_table, jons\_table |  |  |
| lateralwave | Switch for lateral boundary at left | neumann | neumann, wavecrest, cyclic |  |  |
| m | Power in cos^m directional distribution for instat = stat, bichrom, ts\_1 or ts\_2 | 10 | 2 - 128 | - |  |
| nmax\* | Maximum ratio of cg/c for computing long wave boundary conditions | 0.8 | 0.5 - 1.0 | - |  |
| taper | Spin-up time of wave boundary conditions, in morphological time | 100.0 | 0.0 - 1000.0 | s |  |

### Spectral wave boundary conditions

Spectral wave boundary conditions are enabled using *instat* values *jons*, *swan*, *vardens* or *jons\_table*. The conditions are defined in separate files referenced from the *params.txt* file using the *bcfile* keyword. A spectral wave boundary condition describes a spectrum *shape* that XBeach uses to generate a (random) wave time series. The length and resolution of the generated time series is determined by the keywords *rt* and *dtbc* respectively. XBeach will reuse the generated time series until the simulation is completed. The resolution of the time series should be enough to accurately represent the bound long wave, but need not be as small as the time step used in XBeach.

An overview of all keywords relevant for spectral wave boundary conditions is given in the table below. The necessary file formats for each type of spectral wave boundary condition is explained in the following subsections.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| Tm01switch\* | Switch to enable Tm01 rather than Tm-10 | 0 | 0 - 1 | - |  |
| bcfile | Name of spectrum file |  |  | <file> |  |
| correctHm0\* | Switch to enable Hm0 correction | 1 | 0 - 1 | - |  |
| dtbc\* | Timestep used to describe time series of wave energy and long wave flux at offshore boundary (not affected by morfac) | 1.0 | 0.1 - 2.0 | s |  |
| dthetaS\_XB\* | The (counter-clockwise) angle in the degrees needed to rotate from the x-axis in SWAN to the x-axis pointing East | 0.0 | -360.0 - 360.0 | deg |  |
| fcutoff\* | Low-freq cutoff frequency for instat = jons, swan or vardens boundary conditions | 0.0 | 0.0 - 40.0 | Hz |  |
| instat | Wave boundary condition type | bichrom | stat, bichrom, ts\_1, ts\_2, jons, swan, vardens, reuse, ts\_nonh, off, stat\_table, jons\_table |  |  |
| nonhspectrum\* | Spectrum format for wave action balance of nonhydrostatic waves | 0 | 0 - 1 | - |  |
| nspectrumloc\* | Number of input spectrum locations | 1 | 1 - par%ny+1 | - |  |
| nspr\* | Switch to enable long wave direction forced into centres of short wave bins | 0 | 0 - 1 | - |  |
| oldnyq\* | Switch to enable old nyquist switch | 0 | 0 - 1 | - |  |
| random\* | Switch to enable random seed for instat = jons, swan or vardens boundary conditions | 1 | 0 - 1 | - |  |
| rt | Duration of wave spectrum at offshore boundary, in morphological time | min(3600.d0 | par%tstop - None | s |  |
| sprdthr\* | Threshold ratio to maximum value of S above which spectrum densities are read in | 0.08 | 0.0 - 1.0 | - |  |
| trepfac\* | Compute mean wave period over energy band: par%trepfac\*maxval(Sf) for instat jons, swan or vardens; converges to Tm01 for trepfac = 0.0 and | 0.01 | 0.0 - 1.0 | - |  |
| wbcversion\* | Version of wave boundary conditions | 3 | 1 - 3 | - |  |

#### JONSWAP wave spectra

JONSWAP spectrum input is enabled using *instat=jons*. A JONSWAP wave spectrum is parametrically defined in a file that is referenced using the *bcfile* keyword. This file contains a single parameter per line in arbitrary order. The parameters that can be defined are listed in Table XXX. All variables are optional. If no value is given, the default value as specified in the table is used. It is advised not to specify the keyword *dfj* and allow XBeach to calculate the default value.

A typical JONSWAP definition file looks as follows:

jonswap.txt

Hm0 = 0.8

fp = 0.125

mainang = 285.

gammajsp = 3.3

s = 10.

fnyq = 0.3

It is possible to use an alternative file format for time-varying JONSWAP spectra. To enable this option use the *instat* value *jons\_table*. In this case, each line in the spectrum definition file contains a parametric definition of a spectrum, like in a regular JONSWAP definition file, plus the duration for which that spectrum is used during the simulation. XBeach does not reuse time-varying spectrum files. Therefore the total duration of all spectra should at least match the duration of the simulation. The name of the file can be chosen freely, but the file format is fixed as follows and all parameters should be present in all lines:

jonswap.txt

<Hm0> <Tp> <mainang> <gammajsp> <s> <duration> <dtbc>

Note that we refer to the keywords used in a regular JONSWAP definition file in this example, with three differences: 1) the peak period rather than the peak frequency is defined 2) the duration is added (similar to *rt* in *params.txt*) 3) the time resolution is added (similar to *dtbc* in *params.txt*). The duration and boundary condition time step in this file overrules *rt* and *dtbf* in *params.txt*. This format is also used for time-varying stationary wave boundary conditions as described in 4.4.1 Stationary wave boundary conditions. As an example, the JONSWAP spectrum definition file presented above would look as follows if the significant wave height should be increased with 0.2 m every hour:

jonswap.txt

0.8 8. 285. 3.3 10. 0.3 3600. 0.05

1.0 8. 285. 3.3 10. 0.3 3600. 0.05

1.2 8. 285. 3.3 10. 0.3 3600. 0.05

A more generic way of providing time-varying spectral wave boundary conditions is using a FILELIST construction as described in 4.4.5 Temporally and/or spatially varying wave boundary conditions. This approach is compatible with all spectral wave boundary condition types as well as spatially varying boundary conditions as described in the same section.

Table XXX Overview of available keywords in JONSWAP definition file

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| keyword | description | default | minimum | maximum |
| Hm0 | Hm0 of the wave spectrum, significant wave height [m] | 0.0 | 0.0 | 5.0 |
| fp | Peak frequency of the wave spectrum [s-1] | 0.08 | 0.0625 | 0.4 |
| gammajsp | Peak enhancement factor in the JONSWAP expression [-] | 3.3 | 1.0 | 5.0 |
| s | Directional spreading coefficient, cosine law [-] | 10. | 1.0 | 1000. |
| mainang | Main wave angle (nautical convention) [°] | 270. | 180. | 360. |
| fnyq | Highest frequency used to create JONSWAP spectrum [s-1] | 0.3 | 0.2 | 1.0 |
| dfj | Step size frequency used to create JONSWAP spectrum [s-1] | fnyq/200 | fnyq/1000 | fnyq/20 |

#### SWAN wave spectra

XBeach can read standard SWAN 2D variance density or energy density output files (\*.sp2 files) as specified in the SWAN v40.51 manual. This option is enabled using *instat=swan* in *params.txt* and a reference to the spectrum file via the keyword *bcfile*. XBeach assumes the directional information in the SWAN file is according to the nautical convention. If the file uses the Cartesian convention for directions, the user must specify the angle in degrees to rotate the x-axis in SWAN to the x-axis in XBeach (by the Cartesian convention). This value is specified in *params.txt* using the keyword *dthetaS\_XB*.

Note that time-varying and spatially varying SWAN spectra can be provided using the FILELIST and LOCLIST constructions as described in 4.4.5 Temporally and/or spatially varying wave boundary conditions.

An example of a 2D SWAN spectrum is given below:

swan.txt

SWAN 1 Swan standard spectral file

$ 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

0 0 0 0 0 0 0 0 0 0 0 0

#### Variance density spectra

2D spectral information that is not in SWAN format can be provided using a formatted variance density spectrum file and *instat=vardens*. The spectrum file itself is again referenced using the keyword *bcfile*. The contents of the file must adhere to a specific format:

vardens.txt

<number of frequencies (n)>

<frequency 1>

<frequency 2>

<frequency 3>

...

<frequency n-1>

<frequency n>

<number of directions (m)>

<directions 1>

<directions 2>

<directions 3>

...

<directions m-1>

<directions m>

<variance density 1,1> <variance density 2,1> ... <variance density m,1>

<variance density 1,2> <variance density 2,2> ... <variance density m,2>

...

<variance density 1,n> <variance density 2,n> ... <variance density m,n>

Note that the directions must defined according to the Cartesion convention and in the coordinate system used by XBeach. In this coordinate system 0° corresponds to the direction of the x-axis, while 90° corresponds to the direction of the y-axis. Also, the directions must be defined in increasing order. Time-varying and spatially varying variance density spectra can be provided using the FILELIST and LOCLIST constructions as described in 4.4.5 Temporally and/or spatially varying wave boundary conditions.

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

vardens.txt

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

### Boundary conditions for non-hydrostatic model

If XBeach is run as a non-hydrostatic model, which is essentially the nonlinear shallow water equations with dispersion terms and without a wave-action driver, appropriate wave boundary conditions need to be supplied. This option is enabled by *instat=nonh*.

TODO: nonh wbc description

### Special types of wave boundary conditions

Two special types of wave boundary conditions are available that makes XBeach skip the generation of new wave time series. The first is *instat=off* which simply does not provide any wave forcing on the model and hence no wave action in the model.

The second is *instat=reuse* which makes XBeach reuse wave time series that were generated during a previous simulation. This can be a simulation using the same or a different model as long as the computational grids are identical. In order to reuse boundary conditions, all relevant files should be copied to the current working directory of the model (where the *params.txt* file is located). Relevant files are the *ebcflist.bcf* and *qbcflist.bcf* files and all files referenced therein. Generally, the referenced files have *E\_* and *q\_* prefixes. No further wave boundary condition data need be given in *params.txt*.

### Temporally and/or spatially varying wave boundary conditions

Time-varying spectral wave boundary conditions can be defined by feeding in multiple spectrum definition files rather than a single definition file. In addition, the duration for which these spectra should occur needs to be defined.

To make use of this option, the user must specify a regular *instat* value for spectral wave boundary conditions (*jons*, *swan* or *vardens*), but instead of referencing a single spectrum definition file using the *bcfile* keyword, an extra file listing all spectrum definition files is now referenced.

The first word in this extra 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 spectral definition 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 does not reuse time-varying spectrum files. Therefore the total duration of all spectra should at least match the duration of the simulation.

A typical input file contains the following:

filelist.txt

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

Similar to time-varying spectral wave boundary conditions, also spatially varying wave boundary conditions can be defined using a similar construction. In order to apply spatially varying spectra on the offshore boundary, the user must specify set the keywords *wbcversion =3* and *nspectrumloc=ns* in *params.txt* where *ns* is the number of locations in which a spectrum is defined. By default the number of defined spectra is one.

Similar to time-varying spectral wave boundary conditions, its spatially varying sibling uses an extra file listing all relevant spectrum definition files. The first word in this extra file must be the keyword *LOCLIST*. This line should be followed by one line per spectrum definition location containing the world x-coordinate and world y-coordinate of the location that the input spectrum should apply, and the name of the file containing spectral wave information.

A typical input file for a run with three JONSWAP spectra contains the following:

loclist.txt

LOCLIST

0. 0. jonswap1.inp

0. 100. jonswap2.inp

0. 200. jonswap3.inp

Note that it is not possible to use a mix of JONSWAP, SWAN and variance density files in either a *FILELIST* or a *LOCLIST* construction. It is also not possible to vary *dthetaS\_XB* between files in case of non-nautical SWAN spectra. However, it is possible to combine *FILELIST* and *LOCLIST* files by referencing *FILELIST* files from the *LOCLIST* file. In this case all *FILELIST* files should adhere to the same time discretisation, so the duration and timestep values should be constant over al *FILELIST* files as well as the number of wave spectra definitions.

The manner in which a time series of short wave energy and bound long wave flux is calculated per offshore boundary point for spatially varying spectra is described in REF. The user is reminded that along the offshore boundary of the model, the wave energy, rather than the wave height, is interpolated linearly between input spectra without consideration of the physical aspects of the intermediate bathymetry. In cases with large gradients in wave energy, direction or period, the user should specify sufficient wave spectra for the model to accurately represent changes in offshore wave conditions.

### Notes on the generation of wave boundary conditions

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

Whether or not the wave spectrum of the boundary conditions changes over time, the XBeach module requires a record length during which the current wave spectral parameters are applied. 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. The XBeach module tapers the beginning and end of the boundary condition file. 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 may affect the model results.

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.

## Flow, tide and surge input

An XBeach model needs flow boundary conditions on all boundaries of the model domain. Moreover, each boundary may be imposed by tides or surges. The flow boundary conditions and time-varying tide or surge input are discussed in this section. The details on how the flow is computed within the model are described in the sections on bed friction and viscosity parameters (see 6.2.5 Bed friction and viscosity) and flow numerics (6.2.6 Flow numerics).

### Flow boundary conditions

Flow boundary conditions need to be specified on all sides of the domain. We will differentiate between the offshore, lateral and landward boundaries that are set using the keywords *front*, *back* and *left*/*right* respectively. Table XXX to XXX give an overview of the available flow boundary condition types for each of these boundaries.

The keyword *freewave* can be used to switch from bound to free propagation of long waves, which can be useful when time series of free long waves incident on the offshore boundary need to be specified. The file *bc/gen.ezs* can be used to describe the free long waves at the offshore boundary as discussed in 4.4.1 Stationary wave boundary conditions.

Table XXX Overview of available offshore flow boundary condition types

|  |  |
| --- | --- |
| *front* | description |
| abs1d | absorbing-generating (weakly-reflective) boundary in 1D |
| abs2d | absorbing-generating (weakly-reflective) boundary in 2D |
| wall | no flux wall |
| wlevel | water level specification (from file) |
| nonh\_1d | boundary condition for non-hydrostatic option |
| waveflume | ??? |

Table XXX Overview of available landward flow boundary condition types

|  |  |
| --- | --- |
| *back* | description |
| wall | no flux wall |
| abs1d | absorbing-generating (weakly-reflective) boundary in 1D |
| abs2d | absorbing-generating (weakly-reflective) boundary in 2D |
| wlevel | water level specification (from file) |

Table XXX Overview of available lateral flow boundary condition types

|  |  |
| --- | --- |
| *back* | description |
| wall | no flux wall |
| neumann | Neumann boundary condition (constant water level gradient) |
| neumann\_v | ??? |
| no\_advec | ??? |

The table below gives an overview of all keywords related to the flow boundary conditions:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| ARC\* | Switch for active reflection compensation at seaward boundary | 1 | 0 - 1 | - |  |
| back | Switch for boundary at bay side | abs\_2d | wall, abs\_1d, abs\_2d, wlevel |  |  |
| epsi\* | Ratio of mean current to time varying current through offshore boundary | -1.0 | -1.0 - 0.2 | - |  |
| freewave\* | Switch for free wave propagation 0 = use cg (default); 1 = use sqrt(gh) in instat = ts\_2 | 0 | 0 - 1 | - |  |
| front | Switch for seaward flow boundary | abs\_2d | abs\_1d, abs\_2d, wall, wlevel, nonh\_1d, waveflume |  |  |
| left | Switch for lateral boundary at ny+1 | neumann | neumann, wall, no\_advec, neumann\_v |  |  |
| nc\* | Smoothing distance for estimating umean (defined as nr of cells) | par%ny+1 | 1 - par%ny+1 | - |  |
| order\* | Switch for order of wave steering, 1 = first order wave steering (short wave energy only), 2 = second oder wave steering (bound long wave corresponding to short wave forcing is added) | 2.0 | 1.0 - 2.0 | - |  |
| right | Switch for lateral boundary at 0 | neumann | neumann, wall, no\_advec, neumann\_v |  |  |
| tidetype\* | Switch for offfshore boundary, velocity boundary or instant water level boundary | velocity | instant, velocity |  |  |

### Time-varying tide/surge

XBeach can take in up to four time-varying tidal (or surge) signals to be applied to the four boundaries. The number of tidal signals is determined by the keyword *tideloc* that can take the values 0, 1, 2 or 4. Specifying three tidal signals is not an option. Setting *tideloc=0* disables the time-varying tide/surge option. In this case a constant and uniform water level is used specified by the keyword *zs0*.

The length of the tidal signals is determined by the keyword *tidelen*. This is the number of water levels specified in the file referenced with the *zs0file* keyword. The tidal signal will be interpolated to the local time step of the XBeach simulation; therefore the resolution of the signals only needs to be enough to resolve the water level phenomenon of interest (i.e. tide variations, surge event). The tidal signals are not reused, therefore the signal should be at least as long as the simulation time.

The *zs0file* file must adhere to the following format where the last three columns are optional depending on the value of *tideloc* and *tlen* represents the value of *tidelen*:

tide.txt

<time 1> <zs 1,1> [<zs 2,1> [<zs 3,1> <zs 4,1>]]

<time 2> <zs 1,2> [<zs 2,2> [<zs 3,2> <zs 4,2>]]

<time 3> <zs 1,3> [<zs 2,3> [<zs 3,3> <zs 4,3>]]

...

<time tlen> <zs 1,tlen> [<zs 2,tlen> [<zs 3,tlen> <zs 4,tlen>]]

With the options discussed above we can either impose a uniform and constant water level, a single, two or the maximum number of four time-varying boundary conditions. In the first and last case there is no need to specify the location at which the boundary conditions are imposed. In the first case, the boundary conditions are uniform and in the latter case each signal is imposed on a different corner of the model domain and spatially interpolated along the boundaries. When four tidal signals are provided the first signal is imposed to the left offshore boundary seen from sea (x=1,y=1) and the others according to a clockwise rotation. Therefore the columns in the *zs0file* must follow the order of: (x=1,y=1), (x=1,y=N), (x=N,y=N), (x=N,y=1).

In case of a single tidal signal, the signal is imposed on both offshore corners of the domain, while a constant water level defined by the keyword *zs0* is imposed on the landward corners. In case of two tidal signals there are two options available: 1) the first signal is imposed on the offshore boundary and the second on the landward boundary or 2) the first signal is imposed on the left lateral boundary and the second on the right lateral boundary. The choice between the two options is made using the keyword *paulrevere* where a value *0* indicates the first option and a value *1* indicates the second option. Also in the case of two tidal signals the signals are spatially interpolated along the boundaries.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| paulrevere | Specifies tide on sea and land or two sea points if tideloc = 2 | land | land, sea |  |  |
| tideloc | Number of corner points on which a tide time series is specified | 0 | 0 - 4 | - |  |
| zs0 | Inital water level | 0.0 | -5.0 - 5.0 | m |  |
| zs0file | Name of tide boundary condition series |  |  | <file> |  |

## Water level (dam break)

Water levels can be imposed on the model boundaries as explained in 4.5.2 Time-varying tide/surge after which the shallow water equations force the water body in the model domain. Specific applications may require the initialisation of the entire water body in the model domain at the start of the simulation. For example, an initial significant gradient in the water level that “collapses” at the start of the simulation may simulate a dam break. The initialisation of the water level in the model domain is governed by the keywords listed in the table below.

The keyword *zsinitfile* references an external file describing the initial water levels in the entire model domain. The file should adhere to the same format as the bathymetry input file described in 4.3 Grid and bathymetry.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| hotstartflow\* | Switch for hotstart flow conditions with pressure gradient balanced by wind and bed stress | 0 | 0 - 1 | - |  |
| zs0 | Inital water level | 0.0 | -5.0 - 5.0 | m |  |
| zsinitfile | Name of inital water level file |  |  | <file> |  |

## Wind input

Stationary winds can parametrically defined using the keywords *windv* and *windth* that represent the wind velocity and direction (nautical convention) respectively. Time-varying winds can be defined in an external file referenced by the *windfile* keyword. The file should adhere to the format indicated below. The total length of the time series is automatically determined and should be at least as long as the simulation time.

wind.txt

<time 1> <windv 1> <windth 1>

<time 2> <windv 2> <windth 2>

<time 3> <windv 3> <windth 3>

...

The table below gives an overview of all keywords related to the wind:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| Cd\* | Wind drag coefficient | 0.002 | 0.0001 - 0.01 | - |  |
| rhoa\* | Air density | 1.25 | 1.0 - 2.0 | kgm^-3 |  |
| windfile | Name of file with non-stationary wind data |  |  | <file> |  |
| windth | Nautical wind direction, in case of stationary wind | 270.0 | -360.0 - 360.0 | deg |  |
| windv | Wind velocity, in case of stationary wind | 0.0 | 0.0 - 200.0 | ms^-1 |  |

## Sediment input

The sediment input determines the (initial) composition of the bed and the detail in which processes related to sediment sorting are resolved. This is different from how the sediment transport processes are handled in the model itself and that are described in 6.2.7 Sediment transport and 6.2.8 Sediment transport numerics.

The simplest situation is an XBeach simulation with uniform sediment. In this case it is sufficient to specify the uniform grain size using the keyword *D50* indicating the median grain size. The effects of a specific sediment distribution can be parametrically defined by additionally specifying values for *D15* and *D90* and optionally the bed composition can be fine-tuned by specifying the porosity and sediment density using the keywords *por* and *rhos* respectively. In this simple case no sorting of sediment will be simulated.

If the effect of different sediment fractions, sorting and armouring are of importance, multiple sediment fractions can be defined. The number of sediment fraction is determined by the keyword *ngd*. For each sediment fraction a value for *D50,* and optionally *D15* and *D90*, should be defined separated by a space. Moreover, when using multiple sediment fractions, multiple bed layers are needed as well. The number of bed layers can be defined using the keyword *nd*.

Three types of bed layers are distinguished: 1) the top layer 2) the variable or “breathing” layer and 3) the bottom layers. At least one of each type of bed layer is needed, which makes that at least three bed layers are required (see **Error! Reference source not found.** **Error! Reference source not found.**). Each bed layer has a thickness. Choosing bed layer thicknesses that are in balance with the expected erosion and deposition during the simulation should keep the numerical mixing to a minimum. A bed layer thickness that is too large will result in relatively uniform behaviour, while a bed layer thickness that is too small will result in a lot of shifting and thus numerical mixing. The bed layer thicknesses are determined by the three keywords *dzg1*, *dzg2* and *dzg3* for the top, variable and bottom layers respectively.

Apart from the discretization of the grain size distribution and the vertical structure of the bed, the initial bed composition needs to be defined. The bed composition is defined using external files that are not explicitly referenced from *params.txt*, but are assumed to be located in the working directory of the model (next to *params.txt*). There is one file for each sediment fraction specified by *ngd*. The file corresponding to the first sediment fraction is named *gdist1.inp*, the second *gdist2*.inp, et cetera.

The bed composition files hold information on how much sediment of a specific fraction is in each grid cell and bed layer at the start of the simulation. The values are a volumetric fraction that implies that they should add up to unity over all fractions. For example, if a specific grid cell is filled with the first sediment fraction only, the value corresponding to this grid cell will be one in the *gdist1.inp* file and zero in all others. Alternatively, if we defined five sediment fractions and a specific grid cell is filled equally with all fractions, the value corresponding to this grid cell will be 1/5 = 0.2 in all files. The *gidst<N>.inp* files are formatted comparable to the bathymetry files (see 4.3 Grid and bathymetry), but now holds values over the three dimensions x (nx+1), y (ny+1) and the bed layers (nd). The file format is as follows:

gdist1.inp

<p 1,1,1> <p 1,2,1> <p 1,3,1> ... <p 1,nx,1> <p 1,nx+1,1>

<p 1,1,2> <p 1,2,2> <p 1,3,2> ... <p 1,nx,2> <p 1,nx+1,2>

<p 1,1,3> <p 1,2,3> <p 1,3,3> ... <p 1,nx,3> <p 1,nx+1,3>

...

<p 1,1,ny> <p 1,2,ny> <p 1,3,ny> ... <p 1,nx,ny> <p 1,nx+1,ny>

<p 1,1,ny+1> <p 1,2,ny+1> <p 1,3,ny+1> ... <p 1,nx,ny+1> <p 1,nx+1,ny+1>

...

<p 2,1,1> <p 2,2,1> <p 2,3,1> ... <p 2,nx,1> <p 2,nx+1,1>

<p 2,1,2> <p 2,2,2> <p 2,3,2> ... <p 2,nx,2> <p 2,nx+1,2>

<p 2,1,3> <p 2,2,3> <p 2,3,3> ... <p 2,nx,3> <p 2,nx+1,3>

...

<p 2,1,ny> <p 2,2,ny> <p 2,3,ny> ... <p 2,nx,ny> <p 2,nx+1,ny>

<p 2,1,ny+1> <p 2,2,ny+1> <p 2,3,ny+1> ... <p 2,nx,ny+1> <p 2,nx+1,ny+1>

...

<p nd,1,1> <p nd,2,1> <p nd,3,1> ... <p nd,nx,1> <p nd,nx+1,1>

<p nd,1,2> <p nd,2,2> <p nd,3,2> ... <p nd,nx,2> <p nd,nx+1,2>

<p nd,1,3> <p nd,2,3> <p nd,3,3> ... <p nd,nx,3> <p nd,nx+1,3>

...

<p nd,1,ny> <p nd,2,ny> <p nd,3,ny> ... <p nd,nx,ny> <p nd,nx+1,ny>

<p nd,1,ny+1> <p nd,2,ny+1> ... <p nd,nx,ny+1> <p nd,nx+1,ny+1>

The table below gives an overview of all keywords related to working with multiple sediment fractions and bed layers:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| D15 | D15 grain size per grain type | 1 | size(par%D15 - None | m |  |
| D50 | D50 grain size per grain type | 1 | size(par%D50 - None | m |  |
| D90 | D90 grain size per grain type | 1 | size(par%D90 - None | m |  |
| dzg1\* | Thickness of top sediment class layers | par%dzg1 | 0.01 - 1.0 | m |  |
| dzg2\* | Nominal thickness of variable sediment class layer | par%dzg1 | 0.01 - 1.0 | m |  |
| dzg3\* | Thickness of bottom sediment class layers | par%dzg1 | 0.01 - 1.0 | m |  |
| nd\* | Number of computational layers in the bed | 3 | 3 - 1000 | - |  |
| ngd | Number of sediment classes | 1 | 1 - 20 | - |  |
| por | Porosity | 0.4 | 0.3 - 0.5 | - |  |
| rhos | Solid sediment density (no pores) | 2650.0 | 2400.0 - 2800.0 | kgm^-3 |  |
| sedcal\* | Sediment transport calibration coefficient per grain type | 1 | size(par%sedcal - None | - |  |
| ucrcal\* | Critical velocity calibration coefficient per grain type | 1 | size(par%ucrcal - None | - |  |

## Vegetation input

Short wave dissipation and flow interaction due to vegetation is supported. The user can define multiple vegetation species. The number of species is set by the keyword *nveg*. In the file referenced by the keyword *veggiefile* each species is given a name. The properties of each species are summarized in yet another file with the name of the species. This properties file defines the vegetation parameters *ah*, *Cd*, *bv* and *N* that represent the vegetation height, drag coefficient, stem diameter and vegetation density respectively. An example of vegetation definition files is:

veggiefile.txt

cauliflour

corn

cauliflower

ah = 0.2

Cd = 1.0

bv = 0.3

N = 10

corn

nsec = 3

ah = 1.0 0.4 0.3

Cd = 2.0 2.0 2.0

bv = 0.05 0.05 0.1

N = 150 150 150

The *nsec* keyword in the species property file allows the user to define multiple height segments of the species with different properties. The total height of the species is the sum of all *ah* values where the first value corresponds to the lowest segment. Another file referenced by the *veggiemapfile* keyword indicates in what grid cell which vegetation species can be found. The format of this file is similar to the bathymetry files described in 4.3 Grid and bathymetry, but the values are integers referring to a species where 1 refers to the first listed species, 2 to the second, et cetera. A zero indicates no vegetation.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| nveg | Number of vegetation species | -123 |  | - |  |
| veggiefile | Name of vegetation species list file |  |  | <file> |  |
| veggiemapfile | Name of vegetation species map file |  |  | <file> |  |

## Discharge input

Discharge of water at the model boundaries or directly in the model domain is defined along specific grid sections. The keywords *ndischarge* and *ntdischarge* define the number of discharge sections and the length of the discharge time series respectively. The *disch\_loc\_file* keyword references a file that defines the discharge sections. Each line in this file corresponds to a grid section and each line contains four numbers being the start and end coordinates of the section. The file is formatted as follows, where *ndisch* refers to the keyword *ndischarge*:

disch\_loc.txt

<x\_start 1> <y\_start 1> <x\_end 1> <y\_end 1>

<x\_start 2> <y\_start 2> <x\_end 2> <y\_end 2>

<x\_start 3> <y\_start 3> <x\_end 3> <y\_end 3>

...

<x\_start ndisch> <y\_start ndisch> <x\_end ndisch> <y\_end ndisch>

The world coordinates specified in this file must be chosen such that they are close to the desired grid cell borders, since the grid cell borders are eventually used as discharge section. Discharge sections can be located along grid cell borders that are either oriented cross-shore or alongshore, but not a combination of the two. In a regular grid this implies that either the start and end x-coordinates are equal or the start and end y-coordinates are equal. Alternatively, both are equal. In this case a vertical discharge from above is assumed, rather than a horizontal discharge. Vertical discharges only add mass and no momentum to the water body.

The keyword *disch\_timeseries\_file* references a file defining the time series imposed on the discharge locations. The file lists the timings in the first column and a discharge value in m3/s for each discharge section as follows, where *ntdisch* refers to the keyword *ntdischarge*:

disch\_timeseries.txt

<t 1> <Q 1,1> <Q 2,1> ... <Q ndisch,1>

<t 2> <Q 1,2> <Q 2,2> ... <Q ndisch,2>

<t 3> <Q 1,3> <Q 2,3> ... <Q ndisch,3>

...

<t ntdisch> <Q 1,ntdisch> <Q 2,ntdisch> ... <Q ndisch,ntdisch>

Discharges defined at the domain borders are positive in direction towards the domain (influx). Discharges defined in the domain itself are positive in direction of the positive x or y direction. Vertical discharges are positive into the domain (influx).

The table below gives an overview of all keywords related to discharges:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| disch\_loc\_file\* | Name of discharge locations file |  |  | <file> |  |
| disch\_timeseries\_file\* | Name of discharge timeseries file |  |  | <file> |  |
| ndischarge\* | Number of discharge locations | par%ndischarge | 0 - 100 | - |  |
| ntdischarge\* | Length of discharge time series | par%ntdischarge | 0 - 100 | - |  |

## Drifters input

Drifters can be deployed during the model simulation by specifying the number of drifters using the keyword *ndrifter* and the location, start and end time of the drifter deployment in a separate file referenced by the *drifterfile* keyword. The file format is as follows:

drifter.txt

<x 1> <y 1> <t\_start 1> <t\_end 1>

<x 2> <y 2> <t\_start 2> <t\_end 2>

<x 3> <y 3> <t\_start 3> <t\_end 3>

...

<x ndrifter> <y ndrifter> <t\_start ndrifter> <t\_end ndrifter>

The table below gives an overview of all keywords related to drifters:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| drifterfile | Name of drifter data file |  |  | <file> |  |
| ndrifter | Number of drifers | par%ndrifter | 0 - 50 | - |  |

## Ship input

Ship waves can be simulated by defining the ships geometries and trajectories in a collection of files. The user can define multiple ships. The number of ships is set by the keyword *nship*. In the file referenced by the keyword *shipfile* each ship is given a name. The properties of each ship are summarized in yet another file with the name of the ship. This properties file defines the parameters discretization of the ships geometry using the keywords *dx, dy, nx* and *ny*. The geometry is then defined in a separate file referenced by the keyword *shipgeom* from the ship properties file. This file contains depth values of the hull discretized by the *dx, dy, nx,* and *ny* parameters. The center of gravity of the ship is also defined in the ship properties file using the keywords *xCG, yCG* and *zCG*. The ships trajectory is defined in a file referenced from the ship properties file by the keyword *shiptrack*. Each line in this file contains a time, x- and y-coordinate indicating the ships trajectory. In case the option *flying* is enabled, also a z-coordinate is defined indicating the vertical position of the ship. The two keywords *compute\_force* and *compute\_motion* enable the computation of forces on the ship and the ships motions due to wave forcing respectively. An example of ship definition files is:

shipfile.txt

pannamax

small

pannamax

dx = 10

dy = 10

nx = 30

ny = 10

shipgeom = pannamax\_geom.txt

xCG = 120

yCG = 50

zCG = 30

shiptrack = pannamax\_track.txt

flying = 1

compute\_force = 1

compute\_motion = 1

pannamax\_geom.txt

<z 0,0> <z 1,0> <z 2,0> <z 3,0> ... <z nx,0> <z nx+1,0>

<z 0,1> <z 1,1> <z 2,1> <z 3,1> ... <z nx,1> <z nx+1,1>

...

<z 0,ny> <z 1,ny> <z 2,ny> <z 3,ny> ... <z nx,ny> <z nx+1,ny>

<z 0,ny+1> <z 1,ny+1> <z 2,ny+1> <z 3,ny+1> ... <z nx,ny+1> <z nx+1,ny+1>

pannamax\_track.txt

<t 1> <x 1> <y 1> <z 1>

<t 2> <x 2> <y 2> <z 2>

<t 3> <x 3> <y 3> <z 3>

...

small

dx = 2

dy = 2

nx = 20

ny = 4

shipgeom = small\_geom.txt

xCG = 20

yCG = 40

zCG = 1.5

shiptrack = small\_track.txt

flying = 0

small\_track.txt

<t 1> <x 1> <y 1>

<t 2> <x 2> <y 2>

<t 3> <x 3> <y 3>

...

| Keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| nship\* | Number of ships | -123 |  | - |  |
| shipfile | Name of ship data file |  |  | <file> |  |

## Output selection

Output selection determines what data computed by XBeach is written to a file in terms of location and time and in what format. The output types, output times and output formats supported by XBeach are explained in more detail in the following subsections. The table below gives an overview of all keywords related to model output:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| globalvars\* | Mnems of global output variables, not per se the same size as nglobalvar (invalid variables, defaults) | 'abc' |  | - |  |
| meanvars\* | Mnems of mean output variables (by variables) | 'abc' |  | - |  |
| ncfilename\* | Xbeach netcdf output file name |  |  | <file> |  |
| ncross\* | Number of output cross sections | 0 | 0 - 50 | - |  |
| nglobalvar | Number of global output variables (as specified by user) | -1 | -1 - 20 | - |  |
| nmeanvar | Number of mean, min, max, var output variables | 0 | 0 - 15 | - |  |
| npoints | Number of output point locations | 0 | 0 - 50 | - |  |
| npointvar | Number of point output variables | 0 | 0 - 50 | - |  |
| nrugauge | Number of output runup gauge locations | 0 | 0 - 50 | - |  |
| nrugdepth\* | Number of depths to compute runup in runup gauge | 1 | 1 - 10 | - |  |
| outputformat\* | Output file format | fortran | fortran, netcdf, debug |  |  |
| pointtypes\* | Point types (0 = point, 1 = rugauge) | > NULL() |  | - |  |
| pointvars\* | Mnems of point output variables (by variables) | 'abc' |  | - |  |
| rugdepth\* | Minimum depth for determination of last wet point in runup gauge | 1 | size(par%rugdepth - None | m |  |
| timings\* | Switch enable progress output to screen | 1 | 0 - 1 | - |  |
| tintc\* | Interval time of cross section output | 1.0 | 0.01 - 100000.0 | s |  |
| tintg | Interval time of global output | 1.0 | 0.01 - 100000.0 | s |  |
| tintm | Interval time of mean, var, max, min output | par%tstop-par%tstart | 1.0 - par%tstop-par%tstart | s |  |
| tintp | Interval time of point and runup gauge output | 1.0 | 0.01 - 100000.0 | s |  |
| tscross\* | Name of file containing timings of cross section output | None | None - None | - |  |
| tsglobal\* | Name of file containing timings of global output | None | None - None | - |  |
| tsmean\* | Name of file containing timings of mean, max, min and var output | None | None - None | - |  |
| tspoints\* | Name of file containing timings of point output | None | None - None | - |  |
| tstart | Start time of output, in morphological time | 1.0 | 0.0 - 1000000.0 | s |  |

### Output types

XBeach supports four different types of output: 1) instantaneous spatial output 2) time-averaged spatial output 3) fixed point output or 4) run-up gauge output. In principle any variable in XBeach can be outputted as long as it is part of the *spaceparams* structure defined in *spaceparams.tmpl* in the XBeach source code. An overview of all currently supported parameters in this file is presented in REF.

The amount of output variables used for each type is determined by the keywords *nglobalvar*, *nmeanvar*, *npoints* and *nrugauge*. Each of these keywords takes a number indicating the number of parameters or locations that should be written to file. If any of the keywords is set to zero, the output type is effectively disabled. If *nglovalvar* is set to *-1* then a standard set of output variables is used, being *H, zs, zs0, zb, hh, u, v, ue, ve, urms, Fc, Fy, ccg, ceqsg, ceqbg, Susg, Svsg, E, R, D* and *DR*. If *nglobalvar* is not set it defaults to *-1*. The lines in the *params.txt* file immediately following these keywords determine what parameters or locations are used, as will be explained in more detail in the following subsections.

#### Instantaneous spatial output

Instantaneous spatial output describes the instantaneous state of variables across the entire model domain at various points in time. To make use of this option the user must specify the number of output variables required using the *nglobalvar* keyword in *params.txt*, immediately followed by the names of the requested variables on a separate line each. The output of three instantaneous grids can look as follows:

params.txt

nglobalvar = 3

zs

zb

H

#### Time-averaged spatial output

Time-averaged spatial output describes the time-averaged state of variables across the entire model domain at various points in time. The user can define the averaging period in *params.txt* (see 4.13.2 Output times). To make use of this option the user must specify the number of output variables required using the *nmeanvar* keyword in *params.txt*, immediately followed by the names of the requested variables on a separate line each. The output of two time-averaged grids may look as follows:

params.txt

nmeanvar = 2

u

v

#### Fixed point output

Fixed point output allows the user to select one or more locations for which a time series of data is stored. This output describes a time-series of one or more 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*, immediately followed by one line per output location describing the location coordinates, the number of output variables requested at that location and the names of the output variables to be included. The location coordinates are given separately as x-coordinate and y-coordinate and in world coordinates. XBeach will link the output location to the nearest computational point. The variable names, including the last name, must be followed by a hash symbol (#). Fixed point output significantly reduces the amount of data written to file in each time step and is therefore particularly suitable for high temporal resolution output.

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

params.txt

npoints = 2

0. 800. 6 H#zs#zb#D#u#v#

2000. 1600. 6 H#zs#zb#D#u#v#

#### Run-up gauge output

Run-up gauge output describes a time-series of a number of variables at the (moving) waterline. In this case XBeach scans in an x-directional transect defined by the user for the location of the waterline. Output information is recorded for this point. This is particularly useful to keep track of run-up levels in cross-shore transects.

The definition of run-up gauges is similar to the definition of fixed point output. The user needs to specify the number of run-up gauges using the *nrugauge* keyword in *params.txt*, immediately followed by one line per output location describing the location coordinates, the number of output variables requested at that location and the names of the output variable to be included as with fixed point output. The only difference is that XBeach will link the output location to the nearest computational cross-shore transect rather than just the nearest computational point.

### Output times

The user may determine the output times for regular spatial output variables, time averaged spatial variables and point location variables individually. Run-up gauge output and fixed point output are given at the same moments 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 external file containing times at which output should be given or a combination of both.

#### Output at fixed intervals

The user should define a point in time after the start of the simulation at which the first output is generated for fixed interval output. 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 instantaneous spatial output is given by the *tintg* keyword. The keywords for the interval of time-averaged spatial output and point output are *tintm* and *tintp* respectively, where *tintp* is used both for fixed point and run-up gauge output. Note that *tintg*, *tintm* and *tintp* supersede the older *tint* parameter that is valid for all types of output. 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*. An example of the definition of fixed intervals is given below.

params.txt

tstart = 100.

tintg = 100.

tintp = 2.

tintm = 3600.

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

#### 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. The user specifies the name of the output time series file for instantaneous spatial output using the *tsglobal* keyword. The keywords for time series files for time-averaged spatial output and point output are *tsmean* and *tspoint* respectively, where *tspoint* is again used for both fixed point and run-up gauge output. All time series files must contain on the first line the number of output times followed by every output time on a new line. An example of such irregular output time definition is given below.

params.txt

tsglobal= timeseries1.txt

tspoints = timeseries2.txt

tsmean= timeseries3.txt

timeseries1.txt

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.

In the case of instantaneous spatial output and point output, the first output is given at the first stated point in time. In the case of time-averaged spatial 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 done over every interval.

#### 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.

param.txt

tstart = 100.

tintg = 100.

tspoints = timeseries2.txt

tintm = 3600.

### Output format

XBeach supports two types of output: 1) Fortran binary and 2) netCDF. The output format used is determined by the keyword *outputformat*. The use of netCDF output might nowadays be more convenient since all output (and input) is stored in a single, easy accessible file. Also the netCDF file format is compatible with many programming languages (e.g. Matlab, Python) as well as many visualisation tools (e.g. QuickPlot, Morphan). It should be noted that the support for output types in netCDF could be limited for recent functionalities of the XBeach model.

#### Fortran binary

Output files in Fortran binary format are bare matrix dumps of XBeach’ computational matrices. At each output time, one such matrix block is added to the output file. These files can generally be read by binary read functions, like *fread* in Matlab and the *struct* package in Python.

Output files written in Fortran binary format are given the name *<variable>.dat*, for example *zs.dat*, for instantaneous spatial output. 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. Time-averaged spatial output is stored similarly, but the file names have a suffix indicating the type of averaging *<variable>\_mean.dat*. For time-averaged spatial output also the variance, minimum and maximum values are stored using the suffixes \_*var*, \_*min* and \_*max* respectively.

All data corresponding to fixed point locations will be stored in files called *point<NNN>.dat*. *<NNN>* represents a number between 001 and 999 corresponding to the order in which the points are declared in *params.txt*. The data files are plain text and contain one row for each 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 instantaneous values of the variables at the given point. The order of the variables is equal to the order in which they are defined for that point in *params.txt*. Data corresponding to run-up gauge locations are stored in a the same format as fixed point output, but the files are named *rugau<NNN>.dat*.

An extra file called *dims.dat* is always written at the start of the simulation in Fortran binary output mode. This file contains the dimensions of the XBeach model. It simply states the following dimensions in order: *nt* (number of output time steps)*, nx* (number of grid cells in x-direction)*, ny* (number of grid cells in y-direction)*, ngd* (number of sediment fractions)*, nd* (number of bed layers)*, ntp* (number of point output time steps)*, ntm* (number of time-averaged output time steps). Subsequently, the irregular time series are stored, if applicable: *tsglobal* (irregular output times)*, tspoints* (irregular point output times)and *tsmean* (irregular time-averaged output times). Similarly, a file *xy.dat* is written containing the x- and y- coordinates of the full computational grid.

#### netCDF

All data in netCDF output is stored in a single output file. By default this file is named *xboutput.nc*, but this name can be chosen freely using the keyword *ncfilename*. The netCDF file holds all output data, dimensions and input data in a single file. It should be noted that netCDF files hold a single time axis. Time dimensions can therefore be different compared to the Fortran binary output, since the latter supports different resolutions for point output and spatial output. The temporal unit can be specified in the *params.txt* file using the keyword *tunits*. This unit does not affect calculations and is only used for output. An example of the layout of the netcdf file is given below:

xboutput.nc (structure only, no real contents)

netcdf xboutput {

dimensions:

x = 565 ;

y = 101 ;

wave\_angle = 9 ;

bed\_layers = 3 ;

sediment\_classes = 1 ;

inout = 2 ;

globaltime = 2 ;

tidetime = 435 ;

tidecorners = 2 ;

windtime = 2 ;

variables:

double x(x) ;

x:units = "m" ;

x:long\_name = "local x coordinate" ;

double y(y) ;

y:units = "m" ;

y:long\_name = "local y coordinate" ;

double globaltime(globaltime) ;

globaltime:units = "s" ;

double H(globaltime, y, x) ;

H:units = "m" ;

H:long\_name = "wave height" ;

double zs(globaltime, y, x) ;

zs:units = "m" ;

zs:long\_name = "water level" ;

double zb(globaltime, y, x) ;

zb:units = "m" ;

zb:long\_name = "bed level" ;

double ue(globaltime, y, x) ;

ue:units = "m/s" ;

## Time parameters

In all XBeach simulations the hydrodynamic simulation starts at time 0. Model output can be postponed until the time specified by the keyword *tstart* (see 4.13 Output selection). The simulation stops at the time specified by *tstop*. The time step used in the hydrodynamic simulation is determined based on a given maximum Courant number using the keyword *CFL*. The table below gives an overview of all keywords related to time management:

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| CFL | Maximum Courant-Friedrichs-Lewy number | 0.7 | 0.1 - 0.9 | - |  |
| tstop | Stop time of simulation, in morphological time | 2000.0 | 1.0 - 1000000.0 | s |  |
| tunits\* | Time units in udunits format (seconds since 1970-01-01 00:00:00.00 +1:00) | 's' |  | - |  |

# Bibliography

Holthuijsen, L., Booij, N., & Herbers, T. (1989). A prediction model for stationary, short-crested waves in shallow water with ambient currents. Coastal Engineering, 13(1):23-54.

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

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

Daly, C., Roelvink, J. A., Van Dongeren, A., Van Thiel de Vries, J. S. M., & McCall, R. (2010). Short wave breaking effects on low frequency waves. Proceedings of 32nd International Conference on Coastal Engineering, (1), 1–13.

Walstra, D. J. R., Roelvink, J., and Groeneweg, J. (2000). 3D calculation of wave-driven cross-shore currents. In Proceedings 27th International Conference on Coastal Engineering, pages 1050-1063

Phillips, O. (1977). The dynamics of the upper ocean. Cambridge University Press, page 366

Andrews, D. G.; McIntyre, M. E. (1978a), "An exact theory of nonlinear waves on a Lagrangian-mean flow", Journal of Fluid Mechanics 89 (4): 609-646

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

Ruessink, B.G., Ramaekers, G. and van Rijn, L.C., 2012. On the parameterization of the free-stream non-linear wave orbital motion in nearshore morphodynamic models. Coastal Engineering, 65, 56-63.

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 Rijn, L. (2007). Unified View of Sediment Transport by Currents and Waves . part I, II, III, IV.Journal of Hydraulic Engineering, (June):649–667.

Van Thiel de Vries, J. S. M. (2009). Dune erosion during storm surges. PhD thesis, Delft University of Technology, Delft.

Shields, A. (1936). Anwendung der ahnlichkeits-Mechanik und der Turbulenz-forschung auf die Geschiebebewegung. Preussische Versuchanstalt fur Wasserbrau und Schiffbau, 26:524-526

Soulsby, R. (1997). Dynamics of Marine Sands. Thomas Telford Publications, London. ISBN 0 7277 2584 x.

van Rijn, L.C. (1984). Sediment transport, part iii: Bed forms and alluvial roughness. Journal of Hydraulic Engineering, 110(12):1733-1754

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.

Komar, P.D. and Miller, M.C. (1975). On the comparison between the threshold of sediment motion under waves and unidirectional currents with a discussion of the practical evaluation of the threshold; reply. Journal of Sedimentary Research, 45(1):362-367

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

Dalrymple, R.A., Kirby, J.T., Hwang, P.A. (1984). Wave diffraction due to areas of energy dissipation. J. Waterw. Port Coast. Ocean Eng. 110, 67–79.

Mendez, F.M., Losada, I.J. (2004). An empirical model to estimate the propagation of random breaking and nonbreaking waves over vegetation fields. Coastal Engineering 51,103–118.

Suzuki, T., Zijlema, M., Burger, B., Meijer, M.C., Narayan, S. (2011). Wave dissipation by vegetation with layer schematization in SWAN. Coastasl Engineering 59, 64-71.

Van Rooijen, A.A., Van Thiel de Vries, J.S.M., McCall, R.T., Van Dongeren, A.R., Roelvink, J.A., Reniers, A.J.H.M. (2015). Modeling of wave attenuation by vegetation with XBeach. E-proceedings of the 36th IAHR World Congress 28 June – 3 July, 2015, The Hague, The Netherlands.

# Appendices

## Hands on exercises (based on basic XBeach exercises)

### Dune erosion at Delfland, Netherlands (1D)

The first case we will run is a relative simple 1D case. It concerns a profile along the Dutch coast and the hydraulic boundary conditions are based on the 1953 storm surge that caused substantial flooding in the Netherlands.

You can work on the following assignments:

1. Go to the folder “Examples\DelflandStorm” and double click the file “run\_model.bat”. The simulation will start. The model will run for a few minutes, but in the meantime you can already work on question 2 to 5.
2. Open params.txt in which you specify the model input files and settings. Check the number of grid-points in x-direction (keyword: *nx*) and y-direction (keyword: *ny*). Check the filenames in which you specify the wave conditions (keyword: *bcfile*) and the storm surge level (SSL) (keyword: *zs0file*).
3. Do the wave conditions change during the simulation? What is/are the wave height(s) and wave period(s) applied in the simulation?
4. Does the storm surge level change during the simulation? What is the maximum surge height in the simulation.Surge height is defined with respect to the mean sea level (MSL)?
5. What is the simulation time (keyword: *tstop*)? Do we apply a morphological acceleration factor (keyword: *morfac*)? What variables are stored as output and with what time interval? How much hydrodynamic time is simulated?
6. Probably the simulation has finished. When you start the model, it generates a file named XBlog.txt. Open this file and check what is stored in the file. What was the total simulation time?
7. To check out the simulation results we make use of the Quickplot tool (A brief tutorial is attached to this document). You can start Quickplot via the Delft3D environment we installed (Start 🡪 Programs 🡪 Deltares 🡪 Delft3D 🡪 Delft3D). In the Delft 3D menu choose Utilities 🡪 Quickplot. Choose Files of type “NetCDF files and GRIB files” and open “xboutput.nc” in the simulation folder.
8. Use the Quickplot tutorial and try to make an animation in which you plot short wave height (H), water level (including long wave variations, zs) and bed level (zb) as function of time.
9. Plot the offshore water level as function of time. Also open the file “tide.tek” (Tekal data files format), which contains the imposed surge level. Did the model correctly simulate the imposed surge level?
10. Copy all model files to a new folder named “superfast”. Edit params.txt and set ny=0 (instead of ny=2), and run the model. What is the simulation time compare to the original simulation?
11. Compare simulation results for the “superfast” and “default” simulation. Are these the same? What option will you use in the future?

### Nourishment scenarios near Kijkduin, Holland (1D)

This case concerns the exploration of a nourishment strategy near Kijkduin along the Holland coast in the Netherlands. At this location a mega nourishment of 21 Mm3 named the Sand Engine was constructed. In this case we will explore to what extent nourishments can reduce the (dune and beach) erosion during a storm event.

You can work on the following assignments:

1. Go to the folder “Examples\Nourishment case” and double click on the file “runall.bat”. This batch file will run three simulations sequentially in which the profile configuration varies and corresponds with the undisturbed profile (folder reference), a shoreface nourishment (folder shoreface) and a beach nourishment (folder beach) respectively. Each model will run for a few minutes. While running you can already answer question 2 to 6.
2. For the reference case open the params.txt in which you specify model input files and settings. Check the number of grid-points in x-direction (keyword: *nx*) and y-direction (keyword: *ny*). How many directional wave bins are defined and what is their width (keywords: *thetamin*, *thetamax*, *dtheta*).
3. Do the wave conditions change during the simulation? What is/are the wave height(s) and wave period(s) applied in the simulation?
4. Does the storm surge level change during the simulation? What is the maximum surge height in the simulation.Surge height is defined with respect to the mean sea level (MSL)?
5. What is the simulation time (keyword: *tstop*)? Do we apply a morphological acceleration factor (keyword: *morfac*)? What variables are stored as output and with what time interval? How much hydrodynamic time is simulated?
6. Probably the simulation has finished. When you start the model, it generates a file named XBlog.txt. Open this file and check what is stored in the file. What was the total simulation time?
7. Inspect the initial bathymetries of each simulation with QUICKPLOT. Choose Files of type “NetCDF files and GRIB files” and open “xboutput.nc” in the simulation folder).
   1. At what cross-shore position were the shoreface nourishment and beach nourishment placed?
   2. What is the (average) thickness of the nourishments?
   3. Is the volume of the nourishments comparable?
   4. Plot the reference profile with markers; does the grid resolution vary in cross-shore direction?
8. Use the Quickplot tutorial and try to make an animation in which you plot short wave height (H), water level (including long wave variations, zs) and bed level (zb) as function of time.
9. Plot the offshore water level as function of time. Also open the file “tide.tek” (Tekal data files format), which contains the imposed surge level. Did the model correctly simulate the imposed surge level?
10. Inspect the final bathymetries of each simulation.
    1. What is the dune face retreat in the three simulations you have carried out?
    2. Where does the eroded sediment form the dunes deposit?
    3. What nourishment type is most effective in reducing the impact of a storm and do you have an explanation for this?
11. In the folder “banquette” you find a final simulation in which a special beach nourishment type is evaluated named a banquette. This beach nourishment has a highly elevated flat area that connects to the dune foot on which beach restaurants can be build.
    1. Run the model and compare in Quickplot the banquette design with the beach nourishment design we have evaluated before. Do you expect more or less erosion?
    2. Check your hypothesis by comparing the final profile of the banquette simulation to the other simulations.
    3. What would be your approach to further reduce beach and dune erosion?

### Overwash at Santa Rosa Island , USA (2DH)

This case concerns overwash at Santa Rosa island in the Gulf of Mexico during hurricane Ivan in 2004.

You can work on the following assignments.

1. For the reference case open the params.txt in which you specify model input files and settings. Check the number of grid-points in x-direction (keyword: *nx*) and y-direction (keyword: *ny*). How many directional wave bins are defined and what is their width (keywords: *thetamin*, *thetamax*, *dtheta*).
2. In this simulation the grid is specified in Delft3D format. Open Quickin in the Delft 3D menu (Grid 🡪 Quickin) and use the brief tutorial to read in the grid and bathymetry. Does the grid resolution vary in cross-shore direction? And in longshore direction? What are the minimum dx and dy? Why can the grid be coarse offshore?
3. How many wave conditions do we apply in this simulation? What is the offshore mean wave direction? Does the surge level change in the simulation?
4. What is the simulation time (hydrodynamic and morphologic)?
5. Inspect the model results and make an animation of the short wave height (H) and the water levels (including long wave, zs). Describe what is happening.

* For the water levels set the color limits manual between -0.5 and 3.5.

1. Make an animation of cumulative sedimentation/erosion. Describe what is happening.

* For the sedimentation/erosion set the color limits manual between -3 and 3

1. Look at the mean flow field. Plot the flow field in colored vectors. Where are the flow velocities highest and what is the direction of the flow (cross-shore or longshore)? Is there (also) a longshore current present and what is its intensity?

If you have time left feel free to:

* Narrow or broaden the imposed spectrum by changing the parameter directional spreading (*s*) in ‘jonswap.inp’ (you could for example set s = 100 and s = 2 respectively). Make animations of the instantaneous short wave height to see what is happening to the size of the wave groups.
* Design a nourishment in Quickin to reduce the impact of the storm on Santa Rosa Island. Change the depth file in params.txt to make a simulation with the updated bathymetry.

### Yanchep perched beach and natural breakwater (2DH)

This case is an example of a beach 60km north of Perth most commonly known as Yanchep lagoon. Many beaches in WA like Yanchep are fronted by shallow reef and here we are investigating the effects of the reef on the morphodynamics.

You can work on the following assignments:

1. Go to the folder “Examples\YanchepBeach” and double click the file “run\_model.bat”. The simulation will start (and will run about 15 minutes).
2. Meanwhile, inspect the bathymetry file and the structure file (using Quickin). What is the depth in the lagoon? Is the reef enclosing the lagoon below or above the model initial water level? What is the wave height at the boundary condition?
3. Use Quickplot and try to make an animation in which you plot short wave height (H), water level (including long wave variations) (zs) and Eulerian velocities (ue and ve) as function of time.What happens in the lagoon?
4. Use Quickplot and try to make an animation of cumulative sedimentation/erosion. What happens in the lagoon?
5. How is the lagoon affected by the mean water level? Increase or decrease the mean water level condition (‘tide.tx’), run the model again (maybe for a shorter time by reducing keyword: *tstop*). How are the circulation and sediment transport affected?
6. What would happen if the lagoon was open at the southern end? Open the structure file (keyword: ne\_layer=’reef.dep’) with the Quickin tool and modify it to allow the southern end of the lagoon to be eroded. Modify the param.txt file to use this new structure file and run the model. Alternatively, remove the reef from the bathymetry and rerun the model without the structure file, by setting the keyword *struct=0*.

If you still have time;

* Reefs are very rough what happens in the model when the friction is increased? Reduce the Chezy roughness and increase the value of *fw*. Rerun the model what do you observe?
* Is wave/current interaction (keyword: *wci=1*) switched on? Rerun the model with the wave/current switch on/off. Compare the output with model you ran previously. How much effect do you see on the morphology?

## Advanced model coefficients

In @ the main input parameters and files required by XBeach to start a simulation are explained. It explained how the user can switch on and off specific processes and how the user can define the model initial and boundary conditions. XBeach offers, however, many more parameters to fine-tune the simulation of different processes. These parameters are listed in the following subsections grouped by process. Most parameters are not relevant for the average XBeach user. Parameters marked with an asterix (\*) are considered advanced options that are recommended to stay untouched unless you know what you are doing.

### Wave numerics

The parameters listed in the table below involve the numerical aspects of the wave action balance that solves the wave propagation in the model. The keyword *scheme* can be used to set the numerical scheme. By default a higher-order upwind scheme is used to minimize numerical dissipation.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| maxerror\* | Maximum wave height error in wave stationary iteration | 5e-05 | 1e-05 - 0.001 | m |  |
| maxiter\* | Maximum number of iterations in wave stationary | 500 | 2 - 1000 | - |  |
| scheme\* | Numerical scheme for wave propagation | upwind\_2 | upwind\_1, lax\_wendroff, upwind\_2 |  |  |
| wavint | Interval between wave module calls (only in stationary wave mode) | 60.0 | 1.0 - 3600.0 | s |  |

### Wave dissipation

The parameters listed in the table below involve the wave dissipation process. For instationary model runs use either *break=roelvink1*, *roelvink2* or *roelvink\_daly.* Note that the standard value *gamma=0.55* and *n=10* was calibrated for option *break=roelvink1*. For *break=roelvink2* the wave dissipation is proportional to H3/h instead of H2; this affects the calibration. For stationary runs the *break=baldock* option is suitable. The *break=roelvink\_daly* option is a model in which waves start and stop breaking. Reducing *gammax* will reduce wave heights in very shallow water, probably 2 is a reasonable value.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| alpha\* | Wave dissipation coefficient in Roelvink formulation | 1.0 | 0.5 - 2.0 | - |  |
| break | Type of breaker formulation | roelvink2 | roelvink1, baldock, roelvink2, roelvink\_daly, janssen |  |  |
| breakerdelay\* | Switch to enable breaker delay model | 1 | 0 - 1 | - |  |
| delta\* | Fraction of wave height to add to water depth | 0.0 | 0.0 - 1.0 | - |  |
| facrun\* | Calibration coefficient for short wave runup | 1.0 | 0.0 - 2.0 | - |  |
| facsd\* | Fraction of the local wave length to use for shoaling delay depth | 1.0 | 0.0 - 2.0 | - |  |
| fw\* | Bed friction factor | 0.0 | 0.0 - 1.0 | - |  |
| fwcutoff | Depth greater than which the bed friction factor is not applied | 1000.0 | 0.0 - 1000.0 | - |  |
| gamma | Breaker parameter in Baldock or Roelvink formulation | 0.55 | 0.4 - 0.9 | - |  |
| gamma2 | End of breaking parameter in Roelvink Daly formulation | 0.3 | 0.0 - 0.5 | - |  |
| gammax\* | Maximum ratio wave height to water depth | 2.0 | 0.4 - 5.0 | - |  |
| n\* | Power in Roelvink dissipation model | 10.0 | 5.0 - 20.0 | - |  |
| shoaldelay\* | Switch to enable shoaling delay | 0 | 0 - 1 | - |  |

### Rollers

The parameters listed in the table below involve the wave roller model. Using the roller model will give a shoreward shift in wave-induced setup, return flow and alongshore current. This shift becomes greater for lower *beta* values.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| beta\* | Breaker slope coefficient in roller model | 0.1 | 0.05 - 0.3 | - |  |
| rfb\* | Switch to feed back maximum wave surface slope in roller energy balance, otherwise rfb = par%Beta | 0 | 0 - 1 | - |  |
| roller\* | Switch to enable roller model | 1 | 0 - 1 | - |  |

### Wave-current interaction

The parameters listed in the table below involve the process of wave-current interaction. With the switch *wci* one can turn off or on the wave-current interaction, The wave current interation will result in a feedback of currents on the wave propagation. *hwci* limits the computation of wave-current interaction in very shallow water where the procedure may not converge.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| cats\* | Current averaging time scale for wci, in terms of mean wave periods | 4.0 | 1.0 - 50.0 | Trep |  |
| hwci\* | Minimum depth until which wave-current interaction is used | 0.1 | 0.001 - 1.0 | m |  |
| wci | Turns on wave-current interaction | 0 | 0 - 1 | - |  |

### Bed friction and viscosity

The parameters listed in the table below involve the settings for bed friction and viscosity influencing the flow in XBeach. The bed friction is influenced by the dimensionless friction coefficient *cf* or other formulation like the dimensional Chézy or Manning. The bed friction formulation applied needs to be determined with the keyword *bedfriction*. It is possible both the define one value (keyword: bedfriccoef) or to apply , spatially varying values for the bed friction. A spatial varying friction can be provided through an external file referenced via the keyword *bedfricfile*. The file has the same format as the bathymetry file explained in 4.3 (Grid and bathymetry).

The horizontal viscosity is composed of 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. It is also possible to use a user-defined value for the horizontal viscosity (keyword *smag = 0*)

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| C | Chezy coefficient | 55.0 | 20.0 - 100.0 | m^0.5s^-1 |  |
| bedfriccoef | Bed friction coefficient | 0.01 | 3.5e-05 - 0.9 | - |  |
| bedfricfile | Bed friction file (only valid with values of C) |  |  | <file> |  |
| bedfriction | Bed friction formulation | chezy | chezy, cf, white-colebrook, manning, white-colebrook-grainsize |  |  |
| cf\* | Friction coefficient flow | 0.003 | 0.001 - 0.1 | - |  |
| nuh | Horizontal background viscosity | 0.1 | 0.0 - 1.0 | m^2s^-1 |  |
| nuhfac\* | Viscosity switch for roller induced turbulent horizontal viscosity | 1.0 | 0.0 - 1.0 | - |  |
| nuhv\* | Longshore viscosity enhancement factor, following Svendsen (?) | 1.0 | 1.0 - 20.0 | - |  |
| smag\* | Switch for smagorinsky subgrid model for viscocity | 1 | 0 - 1 | - |  |

### Flow numerics

The parameters listed in the table below involve the numerical aspects of the shallow water equations that solve the water motions in the model. Especially in very shallow water some processes need to be limited to avoid unrealistic behavior. For example *hmin* prevents very strong return flows or high concentrations and the*eps* determines whether points are dry or wet and can be taken quite small.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| eps | Threshold water depth above which cells are considered wet | 0.005 | 0.001 - 0.1 | m |  |
| eps\_sd | Threshold velocity difference to determine conservation of energy head versus momentum | 0.5 | 0.0 - 1.0 | m/s |  |
| hmin | Threshold water depth above which Stokes drift is included | 0.2 | 0.001 - 1.0 | m |  |
| oldhu\* | Switch to enable old hu calculation | 0 | 0 - 1 | - |  |
| secorder\* | Use second order corrections to advection/non-linear terms based on MacCormack scheme | 0 | 0 - 1 | - |  |
| umin | Threshold velocity for upwind velocity detection and for vmag2 in equilibrium sediment concentration | 0.0 | 0.0 - 0.2 | m/s |  |

### Sediment transport

The parameters listed in the table below involve the process of sediment transport. The keywords *facAs* and *facSk* determine the effect of the wave form on the sediment transport, this is especially important in the nearshore. The *facua* is an alias setting in which both parameters can be varied at once. The wave form model itself is selected using the keyword *waveform*. Processes like short- and long-wave stirring and turbulence can be switched on or off using the keywords *sws, lws*  and *lwt*. Several options for calibrating the sediment transport formulations are available as well as keywords to incorporate the bed slope effect.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| BRfac\* | Calibration factor surface slope | 1.0 | 0.0 - 1.0 | - |  |
| Tbfac\* | Calibration factor for bore interval Tbore: Tbore = Tbfac\*Tbore | 1.0 | 0.0 - 1.0 | - |  |
| Tsmin\* | Minimum adaptation time scale in advection diffusion equation sediment | 0.5 | 0.01 - 10.0 | s |  |
| bdslpeffdir | Modify the direction of the sediment transport based on the bed slope | none | none, talmon |  |  |
| bdslpeffdirfac | Calibration factor in the modification of the direction | 1.0 | 0.0 - 2.0 | - |  |
| bdslpeffini | Modify the critical shields parameter based on the bed slope | none | none, total, bed |  |  |
| bdslpeffmag | Modify the magnitude of the sediment transport based on the bed slope, uses facsl | roelvink\_total | none, roelvink\_total, roelvink\_bed, soulsby\_total, soulsby\_bed |  |  |
| bed\* | Calibration factor for bed transports | 1 | 0 - 1 | - |  |
| betad\* | Dissipation parameter long wave breaking turbulence | 1.0 | 0.0 - 10.0 | - |  |
| bulk\* | Switch to compute bulk transport rather than bed and suspended load separately | 0 | 0 - 1 | - |  |
| dilatancy | Switch to reduce critical shields number due dilatancy | 0 | 0 - 1 | - |  |
| facAs\* | Calibration factor time averaged flows due to wave asymmetry | 0.1 | 0.0 - 1.0 | - |  |
| facDc\* | Option to control sediment diffusion coefficient | 1.0 | 0.0 - 1.0 | - |  |
| facSk\* | Calibration factor time averaged flows due to wave skewness | 0.1 | 0.0 - 1.0 | - |  |
| facsl\* | Factor bedslope effect | 1.6 | 0.0 - 1.6 | - |  |
| facua\* | Calibration factor time averaged flows due to wave skewness and asymmetry | 0.1 | 0.0 - 1.0 | - |  |
| fallvelred | Switch to reduce fall velocity for high concentrations | 0 | 0 - 1 | - |  |
| form | Equilibrium sediment concentration formulation | vanthiel\_vanrijn | soulsby\_vanrijn, vanthiel\_vanrijn |  |  |
| jetfac\* | Option to mimic turbulence production near revetments | 0.0 | 0.0 - 1.0 | - |  |
| lws\* | Switch to enable long wave stirring | 1 | 0 - 1 | - |  |
| lwt\* | Switch to enable long wave turbulence | 0 | 0 - 1 | - |  |
| pormax | Max porosity used in the experession of Van Rhee | 0.5 | 0.3 - 0.6 | - |  |
| reposeangle | Angle of internal friction | 30.0 | 0.0 - 45.0 | deg |  |
| rheeA | A parameter in the Van Rhee expression | 0.75 | 0.75 - 2.0 | - |  |
| smax\* | Maximum Shields parameter for equillibrium sediment concentration acc. Diane Foster | -1.0 | -1.0 - 3.0 | - |  |
| sus\* | Calibration factor for suspensions transports | 1 | 0 - 1 | - |  |
| sws\* | Switch to enable short wave and roller stirring and undertow | 1 | 0 - 1 | - |  |
| tsfac\* | Coefficient determining Ts = tsfac \* h/ws in sediment source term | 0.1 | 0.01 - 1.0 | - |  |
| turb\* | Switch to include short wave turbulence | bore\_averaged | none, wave\_averaged, bore\_averaged |  |  |
| turbadv\* | Switch to activate turbulence advection model for short and or long wave turbulence | none | none, lagrangian, eulerian |  |  |
| waveform | Wave shape model | vanthiel | ruessink\_vanrijn, vanthiel |  |  |
| z0\* | Zero flow velocity level in Soulsby and van Rijn (1997) sediment concentration | 0.006 | 0.0001 - 0.05 | m |  |

### Sediment transport numerics

The parameters listed in the table below involve the numerical aspects of sediment transport that are all considered advanced options. For example the maximum allowed sediment concentration can be varied with the keyword cmax. It is however not recommend to vary these settings.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| cmax\* | Maximum allowed sediment concentration | 0.1 | 0.0 - 1.0 | - |  |
| sourcesink\* | Switch to enable source-sink terms to calculate bed level change rather than suspended transport gradients | 0 | 0 - 1 | - |  |
| thetanum\* | Coefficient determining whether upwind (1) or central scheme (0.5) is used. | 1.0 | 0.5 - 1.0 | - |  |

### Quasi-3D sediment transport

The parameters listed in the table below involve the tuning of quasi-3D sediment transport, if enabled. The most important setting is the kmax in which the user specifies the number of layers used in the quasi 3D sediment model.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| kmax\* | Number of sigma layers in Quasi-3D model; kmax = 1 is without vertical structure of flow and suspensions | 1 | 1 - 1000 | - |  |
| sigfac\* | Dsig scales with log(sigfac) | 1.3 | 0.0 - 10.0 | - |  |
| vicmol\* | Molecular viscosity | 1e-06 | 0.0 - 0.001 | - |  |
| vonkar\* | Von Karman constant | 0.4 | 0.01 - 1.0 | - |  |

### Morphology

The parameters listed in the table below involve the morphological processes. The *dryslp* and *wetslp* keyword define the critical avalanching slope above and below water respectively. If the bed exceeds the relevant critical slope it collapses and slides downward (avalanching). To reduce the impact of these land slides the maximum bed level change due to avalanching is limited by the *dzmax* value. Which of the two slopes is applied to a grid cell is determined by the *hswitch* keyword.

The keyword *morfac* enables the user to decouple the hydrodynamical and the morphological time. This is suitable for situations where the morphological process is much slower than the hydrodynamic process. The factor defined by the *morfac* keyword is applied to all morphological change. A *morfac=10* therefore results in 10 times more erosion and deposition in a given time step than usual. The simulation time is however then shortened with the same factor to obtain an approximate result more quickly. The user can prevent the simulation time to be adapted to the *morfac* value by setting *morfacopt* to zero. The keywords *morstart* and *morstop* let the user enable the morphological processes in XBeach only for a particular period during the (hydrodynamic) simulation. These options can beuseful if a spin-up time is needed for the hydrodynamics.

The *struct* and *ne\_layer* keywords enable the user to specify non-erodible structures in the model. To switch on non-erodible structures use *struct=1*. The location of the structures is specified in an external file referenced by the *ne\_layer* keyword. The file has the same format as the bathymetry file explained in 4.3 (Grid and bathymetry). The values of the file define the thickness of the erodible layer on top of the non-erodible layer. A *ne\_layer* file with only zeros therefore defines a fully non-erodible bathymetry and a file with only tens means a erodible layer of 10 meters. Only at the grid cells where the value in the *ne\_layer* file is larger than zero erosion can occur. Non-erodible layers are infinitely deep and thus no erosion underneath these layers can occur.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| dryslp | Critical avalanching slope above water (dz/dx and dz/dy) | 1.0 | 0.1 - 2.0 | - |  |
| dzmax\* | Maximum bed level change due to avalanching | 0.05 | 0.0 - 1.0 | m/s/m |  |
| hswitch\* | Water depth at which is switched from wetslp to dryslp | 0.1 | 0.01 - 1.0 | m |  |
| morfac | Morphological acceleration factor | 1.0 | 0.0 - 1000.0 | - |  |
| morfacopt\* | Switch to adjusting output times for morfac | 1 | 0 - 1 | - |  |
| morstart | Start time morphology, in morphological time | 120.0 | 0.0 - 10000000.0 | s |  |
| morstop | Stop time morphology, in morphological time | 2000.0 | 0.0 - 10000000.0 | s |  |
| ne\_layer | Name of file containing depth of hard structure |  |  | <file> |  |
| struct | Switch for enabling hard structures | 0 | 0 - 1 | - |  |
| wetslp | Critical avalanching slope under water (dz/dx and dz/dy) | 0.3 | 0.1 - 1.0 | - |  |

### Bed update

The parameters listed in the table below involve the settings for the bed update process especially in the case multiple sediment fractions and bed layers are involved. The *frac\_dz, split* and *merge* keywords determine the fraction of the variable bed layer thickness at which the layer is split or merged respectively with the surrounding bottom layers. The variable layer is chosen using the *nd\_var* keyword.

Pre-defined bed updates can be used with the keywords *nsetbathy* that determines the number of updates and *setbathyfile* that references a file that determines what beds are used for each update. The format of the *setbathyfile* file is as follows:

bathy.txt

???

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| frac\_dz\* | Relative thickness to split time step for bed updating | 0.7 | 0.5 - 0.98 | - |  |
| merge\* | Merge threshold for variable sediment layer (ratio to nominal thickness) | 0.01 | 0.005 - 0.1 | - |  |
| nd\_var\* | Index of layer with variable thickness | 2 | 2 - par%nd | - |  |
| nsetbathy\* | Number of prescribed bed updates | 1 | 1 - 1000 | - |  |
| setbathyfile\* | Name of prescribed bed update file |  |  | <file> |  |
| split\* | Split threshold for variable sediment layer (ratio to nominal thickness) | 1.01 | 1.005 - 1.1 | - |  |

### Groundwater flow

The parameters listed in the table below involve the process of groundwater flow. The vertical permeability coefficient in the vertical can be set differently than that of the horizontal using the keywords *kz* and *kz* respectively. The initial bed level of the aquifer is read from an external file referenced by the *aquiferbotfile* keyword and the initial groundwater head can be set to either a uniform value using the *gw0* keyword or to spatially varying values using an external file referenced by the *gw0file* keyword. Both files have the same format as the bathymetry file explained in 4.3 (Grid and bathymetry).

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| aquiferbot\* | Level of uniform aquifer bottom | -10.0 | -100.0 - 100.0 | m |  |
| aquiferbotfile\* | Name of the aquifer bottom file |  |  | <file> |  |
| dwetlayer\* | Thickness of the top soil layer interacting more freely with the surface water | 0.1 | 0.01 - 1.0 | m |  |
| gw0\* | Level initial groundwater level | 0.0 | -5.0 - 5.0 | m |  |
| gw0file\* | Name of initial groundwater level file |  |  | <file> |  |
| gwReturb\* | Reynolds number for start of turbulent flow in case of gwscheme = turbulent | 100.0 | 1.0 - 600.0 | - |  |
| gwfastsolve\* | Reduce full 2D non-hydrostatic solution to quasi-explicit in longshore direction | 0 | 0 - 1 | - |  |
| gwheadmodel\* | Model to use for vertical groundwater head | parabolic | parabolic, exponential |  |  |
| gwhorinfil\* | Switch to include horizontal infiltration from surface water to groundwater | 0 | 0 - 1 | - |  |
| gwnonh\* | Switch to turn on or off non-hydrostatic pressure for groundwater | 0 | 0 - 1 | - |  |
| gwscheme\* | Scheme for momentum equation | laminar | laminar, turbulent |  |  |
| kx\* | Darcy-flow permeability coefficient in x-direction | 0.0001 | 1e-05 - 0.1 | ms^-1 |  |
| ky\* | Darcy-flow permeability coefficient in y-direction | 0.0001 | 1e-05 - 0.1 | ms^-1 |  |
| kz\* | Darcy-flow permeability coefficient in z-direction | 0.0001 | 1e-05 - 0.1 | ms^-1 |  |

### Non-hydrostatic correction

The parameters listed in the table below involve the settings for the non-hydrostatic option (keyword: nonh=1). These are all considered advanced options and it is thus recommended not to change these

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| Topt\* | Absolute period to optimize coefficient | 10.0 | 1.0 - 20.0 | s |  |
| breakviscfac\* | Factor to increase viscosity during breaking | 1.5 | 1.0 - 3.0 | - |  |
| breakvisclen\* | Ratio between local depth and length scale in extra breaking viscosity | 1.0 | 0.75 - 3.0 | - |  |
| dispc\* | Coefficient in front of the vertical pressure gradient | 1.0 | 0.1 - 2.0 | ? |  |
| kdmin\* | Minimum value of kd (pi/dx > min(kd)) | 0.0 | 0.0 - 0.05 | - |  |
| maxbrsteep\* | Maximum wave steepness criterium | 0.6 | 0.3 - 0.8 | - |  |
| nhbreaker\* | Non-hydrostatic breaker model | 2 | 0 - 3 | - |  |
| reformsteep\* | Wave steepness criterium to reform after breaking | 0.25d0\*par%maxbrsteep | 0.0 - 0.95d0\*par%maxbrsteep | - |  |
| secbrsteep\* | Secondary maximum wave steepness criterium | 0.5d0\*par%maxbrsteep | 0.0 - 0.95d0\*par%maxbrsteep | - |  |
| solver\* | Solver used to solve the linear system | tridiag | sip, tridiag |  |  |
| solver\_acc\* | Accuracy with respect to the right-hand side usedin the following termination criterion: ||b-Ax || < acc\*||b|| | 0.005 | 1e-05 - 0.1 | - |  |
| solver\_maxit\* | Maximum number of iterations in the linear sip solver | 30 | 1 - 1000 | - |  |
| solver\_urelax\* | Underrelaxation parameter | 0.92 | 0.5 - 0.99 | - |  |

### Physical constants

The parameters listed in the table below involve physical constants used by XBeach. The gravitational acceleration and density of water are universally used coefficient. The depthscale is a factor in order to set different cut-off values like eps and hswitch. A value of the depthscale lower than one means the cut-off values will increase.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| depthscale\* | Depthscale of (lab)test simulated, affects eps, hmin, hswitch and dzmax | 1.0 | 1.0 - 200.0 | - |  |
| g | Gravitational acceleration | 9.81 | 9.7 - 9.9 | ms^-2 |  |
| rho | Density of water | 1025.0 | 1000.0 - 1040.0 | kgm^-3 |  |

### Coriolis force

The parameters listed in the table below involve the settings for incorporating the effect of Coriolis on the shallow water equations. The keywords are universally used coefficients.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| lat\* | Latitude at model location for computing Coriolis | 0.0 | -90.0 - 90.0 | deg |  |
| wearth\* | Angular velocity of earth calculated as: 1/rotation\_time (in hours) | 1.d0/24.d0 | 0.0 - 1.0 | hour^-1 |  |

### MPI

The parameters listed in the table below involve the settings for parallelization of XBeach. A full description of the parallelization of XBeach can be found in REF. When running XBeach in parallel mode, the model domain is subdivided in submodels and each submodel is then computed on a separate core. This will increasethe computational speed of the model. The submodels only exchange information over their boundaries when necessary. The MPI parameters determine how the model domain is subdivided. The keyword *mpiboundary* can be set to *auto, x, y* or *man*. In *auto* mode the model domain is subdivided such that the internal boundary is smallest. In *x* or *y* mode the model domain is subdivided in submodels extending to either the full alongshore or the full cross-shore extent of the model domain. In *man* mode the model domain is manually subdivided using the values specified with the *mmpi* and *nmpi* keywords. The number of submodels is not determined by XBeach itself, but by the MPI wrapper (e.g. MPICH2 or OpenMPI). It is important to note that information about slopes isn’t exchanged over the bounadries. Therefore the avalanching algorithm will not function over MPI boundaries.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| mmpi\* | Number of domains in cross-shore direction when manually specifying mpi domains | 2 | 1 - 100 | - |  |
| mpiboundary\* | Fix mpi boundaries along y-lines, x-lines, use manual defined domains or find shortest boundary automatically | auto | auto, x, y, man |  |  |
| nmpi\* | Number of domains in alongshore direction when manually specifying mpi domains | 4 | 1 - 100 | - |  |

### Output projection

The parameters listed in the table below involve the projection of the model output. These settings do not influence the model results in anyway. The *rotate* keyword can be used to rotate the model output with an angle specified by the keyword *alfa*. The *projection* string can hold a string specifying the coordinate reference system used and is stored in the netCDF output file as meta data.

| keyword | description | default | range | units | remark |
| --- | --- | --- | --- | --- | --- |
| projection\* | Projection string | ' ' |  | - |  |
| rotate | Rotate output as postprocessing with given angle | 1 | 0 - 1 | - |  |
|  |  |  |  |  |  |

## Numerical implementation

### Grid types

#### 1D

#### Rectilinear

#### Curvilinear

### Grid set-up

The new implementation utilizes a curvilinear, staggered grid where depths, water levels, wave action and sediment concentrations are given in the cell centers (denoted by subscript z) and velocities and sediment fluxes at the cell interfaces (denoted by subscript u or v). In Figure 1 the z, u, v and c (corner) points with the same numbering are shown. The grid directions are named s and n; grid distances are denoted by and , with subscripts referring to the point where they are defined. A finite-volume approach is utilized where mass, momentum and wave action are strictly conserved. In the middle panel of Figure 1, the control volume for the mass balance is shown with the corresponding grid distances around the *u-* and *v-*points. The right panel explains the numbering of the fluxes *Q* and the volume *V*.



Figure 1 Location of staggered grid points (left panel); definition of grid distances (middle) and terms in volume balance (right). This schematization is presented on a curvilinear grid.

### Shallow water equations

#### Mass balance equation

The mass balance reads as follows:



This is discretized according to:



In this formatulion *Az* is the area of the cell around the cell centre, *zs* is the surface elevation, *uu* is the u-velocity in the u-point, *hu* the water depth in the u-point and *vv* the v-velocity in the v-point. The indices *i,j* refer to the grid number in u resp. v direction; the index *n* refers to the time step.

#### Momentum balance equation

Second, we will outline the derivation of the u-momentum balance. The control volume is given in Figure 2. It is centered around the u-point. We now consider the rate of change of the momentum in the local u-direction as follows:



In @ V is the cell volume, u the velocity in local grid direction, Q the fluxes, the density, g acceleration of gravity,  the bed shear stress, wind shear stress and wave force in u-direction. We consider that the outgoing fluxes carry the velocity inside the cell, *u* and that *uin* is determined at each inflow boundary by interpolation, reconstructing the component in the same direction as *u*.

The volume balance for the same volume reads:



By multiplying the volume balance by *u*, subtracting it from the momentum balance and dividing the result by *V* we arrive at the following equation:



In @ *A*  is the cell area and *hum* is the average depth of the cell around the *u*-point. The procedure for the second term (the others are straightforward) now boils down to integrating (only) the incoming fluxes over the interfaces and multiplying them with the difference between *u* in the cell and the component of velocity in the same direction at the upwind cell.



**Figure 2 Control volume u-momentum balance and definition of fluxes**

In equations and and the procedure for computing the u-momentum balance is outlined. The discharges in the u-points are computed by multiplying the velocity in the u- or v-point by the water depth at that point. These discharges are then interpolated to the borders of the control volume around the u-point. The difference in grid orientation between the incoming cell and the u-point is computed and used to compute the component of the incoming velocity in the local u-direction, from the left and right side of the control volume.



The same is done for the top and bottom of the control volume, based on the discharges in v-direction:



Finally, the advective term in the momentum balance is given in equation .



#### Time integration scheme

The time integration of the mass and momentum balance equations is combined in an explicit leap-frog scheme, as depicted in Figure 3. The velocities (in the '-' points) are updated using the momentum balance, the water levels are updated using the mass balance. The water level gradients influence the momentum balance and the velocities and derived discharges affect the mass balance. Because of the leap-frog scheme these influences are always computed at the half time step level, which makes the scheme second order accurate.



**Figure 3 Leap-frog time integration scheme**

Using this straightforward finite volume approach, complicated transformations of the equations are avoided and the solution scheme remains transparent. It is also completely compatible with the original rectilinear implementation and is even slightly more efficient.

### Wave action balance

#### Nonstationary solver

The time-varying wave action balance solved in XBeach is as follows:



Where *E* is the wave energy or wave action, Cg is the group velocity, the refraction speed in theta-space and *Sink* refers to effects of wave breaking and bottom friction.

Again, the advection terms are the only ones affected by the curvilinear scheme so we will discuss their treatment in detail. The control volume is the same as for the mass balance. In equation the procedure to compute the wave energy fluxes across the cell boundaries is outlined. All variables should also have an index *itheta* referring to the directional grid, but for brevity these are omitted here.

The component of the group velocity normal to the cell boundary, at the cell boundary, is interpolated from the two adjacent cell center points. Depending on the direction of this component, the wave energy at the cell boundary is computed using linear extrapolation based on the two upwind points, taking into account their grid distances. This second order upwind discretization preserves the propagation of wave groups with little numerical diffusion.



The other three fluxes are computed in a similar way; for brevity we will not present all formulations.

The time integration is explicit and the same as in the original implementation. The advection in u- and v-direction is computed simply by adding the four fluxes and dividing by the cell area. This procedure guarantees conservation of wave energy.



The procedure for the roller energy balance is identical to that for the wave energy balance and will not be repeated here.

#### Stationary solver

In the stationary solver the wave energy and roller energy balances are solved line by line, from the seaward boundary landward. For each line the automatic timestep is computed and the quasi-time-dependent balance according to equation is solved until convergence or the maximum number of iterations is reached, after which the solver moves to the next line.

The iteration is controlled by the parameters *maxiter*  and *maxerror.*

### Shallow water equations

### Groundwater flow

#### Hydrostatic

Not in manual yet, to-do…

#### Non-hydrostatic

In order to solve the equations in xx, the spatial and temporal domain of the groundwater system is split into the same spatial grid and time steps as the XBeach surface water model it is coupled to. At each time step in the numerical model, the depth average groundwater head is calculated in the centre of the groundwater cells, and the fluxes (specific discharge, submarine exchange, infiltration and exfiltration) are calculated on the cell interfaces

At the start of the time step, every cell is evaluated whether the groundwater and surface water are connected:



In *ε* is a numerical smoothing constant used to deal with numerical round off errors near the bed, and *i* and *j* represent cross-shore and longshore coordinates in the numerical solution grid, respectively. Infiltration is calculated in cells where the groundwater and surface water are not connected and there exists surface water. As shown in the infiltration rate is a function of the thickness of the wetting front, which is zero at the start of infiltration, and increases as a function of the infiltration rate. The equations for the infiltration rate and the thickness of the wetting front are approximated by first-order schemes, in which the wetting front is updated using a backward-Euler scheme, which ensures numerical stability:



In the superscript *n* corresponds to the time step number and *Δt* is the size of the time step. The infiltration rate in the coupled relationship can be solved through substitution:



At the end of infiltration, i.e. when the groundwater and surface water become connected or there is no surface water left, the wetting front thickness is reset to zero. If the infiltration rate exceeds the Reynolds number for the start of turbulence, the local hydraulic conductivity is updated using the local Reynolds number:



Xbeach iterates until a minimum threshold difference between iterations is found for and . Infiltration in one time step is limited to the amount of surface water available in the cell and to the amount of water required to raise the groundwater level to the level of the bed:



If during infiltration the groundwater level reaches the bed level, the fraction of the time step required to do so is estimated (x) and the remaining fraction is used in the submarine exchange.



Exfiltration is calculated in cells where the groundwater and surface water are not connected and the groundwater level exceeds the bed level:



After infiltration and exfiltration have been calculated, the groundwater level and surface water level are updated:



All updated cells are subsequently re-evaluated on whether the surface water and groundwater are connected or unconnected

The horizontal specific discharge on each cell interface can be found through an approximation of the groundwater head gradient:



In the superscripts *x* and *y* refer to the components of the variable in the crossshore and longshore direction, respectively, the subscripts *u* and *v* refer to variables approximated at the horizontal cell interfaces in the cross-shore and longshore direction, respectively, and the subscript *H* refers to variables approximated at the cell centers. The hydraulic conductivity may be different at each cell interface and is therefore computed at every interface where every K is calculated separately. The cell height at the centre of the groundwater cells (*ΔzH,i,j*) is calculated from the groundwater level and the bottom of the aquifer in the centre of the cell, whereas the cell heights at the horizontal cell interfaces are calculated using an upwind procedure:



As described in Section 3.3.6, the head applied on the top boundary of the groundwater domain (*Hbc*) depends on whether the groundwater and surface water are connected or unconnected:



The vertical submarine exchange at the top of the numerical groundwater cell, is found with



In the superscript *z* refers to the vertical component of the variable, the subscript *w* refers to a numerical approximation at the vertical cell interfaces.

Continuity in the groundwater cell is found following



All variables in contain an unknown value for the groundwater pressure head, described in terms of a known head at the surface of the groundwater (*Hbc*) and the unknown curvature of the vertical groundwater head function (*β*). Since water is incompressible, the groundwater pressure must be solved for all cells simultaneously using matrix algebra:



In A is a matrix containing coefficients for the horizontal and vertical specific discharge, x is a vector containing the unknown groundwater head curvature, and b contains the known forcing terms. For a one dimensional cross-shore case, A is reduced to a tridiagonal matrix. The vector of known forcing consists of the numerical gradients in the contribution of the head applied on the top boundary of the groundwater domain to the horizontal specific discharge.

In the one dimensional case, the solution to the tridiagonal matrix A can be computed using the efficient Thomas algorithm (Thomas 1949). In the two dimensional case, matrix A contains two additional diagonals that are not placed along the main diagonal, and vector b contains additional forcing terms from the alongshore contribution. The solution to the two dimensional case requires a more complex and less computationally efficient matrix solver. In this case the Strongly Implicit Procedure (Stone 1968) is used in a manner similar to Zijlema et al. (2011).

The horizontal and vertical groundwater fluxes are calculated using the solution of *x* plus and . Since some local velocities may exceed the critical Reynolds number for the start of turbulence (*Recrit*), the turbulent hydraulic conductivity (*K*) is updated using the local Reynolds number. The solution to and the update of the turbulent hydraulic conductivity are iterated until a minimum threshold difference between iterations is found.

The iterated solution for the specific vertical discharge is used to update the groundwater level and surface water level:



If the groundwater and surface water are connected, and the submarine exchange from the surface water to the groundwater estimated in is greater than the amount of surface water available in the cell, continuity is enforced by lowering the groundwater level to compensate for the lack of permeating water:



### Sediment transport

The advection-diffusion equation for suspended sediment is the basis for the sediment transport computations in XBeach. The partial differential equation to solve is:



Here *c* is the depth-averaged concentration, *ceq* the equilibrium concentration, *Ts* a typical timescale proportional to water depth divided by fall velocity. As is often done to increase robustness, we treat the erosion term explicitly but take an implicit scheme for the sedimentation term:



This can be rewritten as:



The sediment transport gradient is discretized in a similar way as the mass balance:



The sediment transports in the u- points contain an advective term, a diffusive term and a bed slope term:



Here *urep,s* is a representative velocity for suspended transport, which contains contributions due to return flow, wave skewness and wave asymmetry; *Dc* is a horizontal diffusion coefficient and *fslope* a coefficient. In discretized form the expression for the suspended transport in the u-point is:



The concentrations in the u-points are computed with a -method, where  means a fully upwind approximation, and  a central scheme. In practice, we mostly use the upwind approximation for its robustness.



The erosion and deposition terms, which may also be used in the bed updating, are finally computed from:



The evaluation of the bedload transport takes place in the same way as in the previous versions of XBeach, except for the fact that the directions are taken in local grid direction, and will not be repeated here.

### Bottom updating schemes

Two alternative formulations are available for the bed updating: one where the bottom changes are computed based on the gradients of suspended and bed load transport, equation , and one where the changes due to suspended transport are accounted for through the erosion and deposition terms, equation .





In both cases *MF* is the morphological factor used to accelerate morphological changes. In the first case, the sediment in the bottom is conserved in all cases, but changes in the amount of sediment in the water are not considered; one can also say that the sediment in suspension is added to the bottom sediment. In the second case, the storage of sediment in the water is accounted for, but will be distorted in cases of high *MF*. Since under most circumstances the real effect of the storage in the water phase is small we prefer the first formulation which guarantees mass conservation in the bottom.

#### Avalanching

#### Bed composition

If the effect of different sediment fractions, sorting and armouring are of importance, a bed composition constituting multiple sediment fractions can be defined. Each sediment fraction is characterized by a median grain size (D50) and possible a D15 and D90 as well. When using multiple sediment fractions, multiple bed layers are needed as well to describe the vertical distribution of the sediment fractions in the bed.

By specifying multiple bed layers, XBeach can keep track of the different sediment fractions both in the horizontal and in the vertical. Coarse sediments may be deposited on top of fine sediment after which erosion of the coarse sediment is needed to expose the fine sediment again, effectively armouring the bed. Three types of bed layers are distinguished: 1) the top layer 2) the variable or “breathing” layer and 3) the bottom layers. The top layer is the only layer that interacts with the water column and can be eroded, but preserves it thickness. The bottom layers are layers of constant thickness that move with the top layer. A single variable or “breathing” layer is defined that adapts its thickness to the erosion and sedimentation of the bed. At least one of each type of bed layer is needed, which makes that at least three bed layers are required. For example: if a grid cell is eroded, particular fractions of sediment are removed from the top layer, but the top layer preserves its thickness and thus it takes the same volume of sediment, likely of different composition than the eroded sediment, from the layer below. If this layer is a bottom layer, the thickness is preserved and again the same volume of sediment is taken form a lower bed layer. This continues until the variable or “breathing” layer is reached. This layer adapts its thickness to the amount of erosion. If the thickness of the layer becomes too small, the variable layer is merged with an adjacent bottom layer and a new bottom layer is defined underneath the existing ones to ensure a constant number of bed layers. Reversely, if a grid cell is accreting, the thickness of the variable layer will be increased and with sufficient increase the variable layer will be split in two effectively creating a new bottom layer. The lowest existing bottom layer is then discarded to ensure a constant number of bed layers.

Each grid cell in XBeach holds its own sediment distribution and the sediment transport formulations used differentiate between fractions. Therefore the distribution of sediment may change over time and processes like armouring and sorting can be simulated. Due to the shifting of sediment between bed layers numerical mixing of sediment occurs. Choosing bed layer thicknesses that are in balance with the expected erosion and deposition during the simulation should keep the numerical mixing to a minimum. A bed layer thickness that is too large will result in relatively uniform behaviour, while a bed layer thickness that is too small will result in a lot of shifting and thus numerical mixing.

### Non-hydrostatic

#### Global continuity equation

As was outlined in the previous chapter the global continuity equation, which describes the relation between the free surface and the depth averaged discharge, is given by



A simple semi-discretisation of using central differences for the space derivative and using the Hansen scheme for the coupling between velocity and free surface results in



With , and the water depth is defined by a first order accurate upwind interpolation



The resulting scheme is only first order accurate by virtue of the upwind interpolations and mass conservative. When first order computations are considered accurate enough  is set to . For higher order accuracy the first order prediction is corrected using a limited version of the McCormack scheme. The corrector step reads



With  and  is given for positive flow as



Here  denotes the minmod limiter. Similar expression can be constructed for negative flow. The expression for  and  are obtained in a similar manner. Note that the total flux at the cell boundaries thus reads



The predictor-corrector set is second order accurate in regions where the solution is smooth, and reduces locally to first order accuracy near discontinuities. Furthermore, the method remains mass conservative. Note that other flux limiters can be used instead of the minmod limiter. However, as the minmod limiter performed adequately, this has not been investigated. ( For an overview of flux limiters see Hirsch, 2007)

#### Local continuity equation

The depth averaged local continuity equation is given by



This equation is discretized using central differences



Missing grid variables are approximated with upwind interpolation. Because there is no separate time evolution equation for the pressure the local continuity equation will be used to setup a discrete set of poison type equations in which the pressures are the only unknown quantities.

#### Horizontal Momentum

To obtain a conservative discretisation of the momentum equation the approach from Stelling and Duinmeijer (2003) is followed. However, to improve the accuracy of the method the combined space-time discretisation of the advection is done using a variant of the MacCormack (1969) is used. This scheme consists of a first order predictor step and a flux limited corrector step. The hydrostatic pressure is integrated using the midpoint rule and central differences, while the source terms and the turbulent stresses are integrated using an explicit Euler time integration. Formally the time integration is therefore first order accurate, but in regions where the turbulent stresses are negligible the scheme is of almost second order accuracy.

The depth averaged horizontal momentum equation for is given by



A first order accurate predictor step in time and space is then given as



Here Pr represents a discretisation of the dynamic pressure; T the effect of (turbulent) viscosity and S includes all other source terms. The discretisation of the (turbulent) viscous terms is given by central differences:



Here  and  are obtained from the surrounding points by simple linear interpolation.

Due to the incompressible flow assumption the dynamic pressure does not have a separate time evolution equation, but instead it satisfies an elliptical equation in space. As such its effect cannot be calculated explicitly using values at the previous time level. However to improve the accuracy of the predictor step the effect of the dynamic pressure is included explicitly. To do this first the unknown pressure is decomposed as:



where the difference in pressure is generally small. In the predictor step the effect of the pressure is included explicitly using. In the corrector step the full Poisson equation is then solved for . The pressure term in the predictor step is thus given as



Here represents the average pressure over the vertical which is approximated with, in which  is the pressure at the bottom. Furthermore  is given as.

Currently is formulated with the depth integrated momentum as the primitive variable, and not the depth averaged velocity. To reformulate in terms of we use the method by Stelling and Duinmeijer (2003). First note that  and  are approximated as  and . Now using  is equivalent to:





Substituting into the full expressions (including those for ) become:



Where we again use a first order upwind interpolation for and. This is exactly the approximation used by Stelling and Duinmeijer (2003) and is fully momentum conservative.

The predictor step is first order accurate in both space and time due to the use of upwind approximations for and Euler explicit time integration for the advective terms, and first order time integration for the source/viscous terms. This level of accuracy is acceptable near shore, where strong non-linearity (wave breaking, flooding and drying) will force the use of small steps in space and time anyway. However, in the region where waves only slowly change (e.g. shoaling/refraction on mild slopes), the first order approximations suffer from significant numerical damping. To improve the accuracy of the numerical model in these regions a corrector step is implemented after the predictor step.

The corrector step is given by:



Or, when formulated in terms of the depth averaged velocity



The values of  are obtained from slope limited expressions. For positive flow these read:



Where  again denotes the minmod limiter. Similar expressions can be constructed for, and.

The predictor-corrector set is second order accurate in regions where the solution is smooth, and reduces to first order accuracy near sharp gradients in the solutions to avoid unwanted oscillations. Furthermore, the method remains momentum conservative.

#### Vertical momentum equations

The vertical momentum equation is discretized in a similar manner to the horizontal momentum equations using the McCormack scheme. In terms of the depth averaged vertical velocity the predictor step is:



The pressures are defined on the cell faces and therefore do not have to be interpolated. Furthermore, we can exactly set the dynamic pressure at the free surface  to zero. The vertical velocities are defined on the cell faces and therefore the depth averaged velocity  needs to be expressed in terms of the bottom and surface velocities. Using a simple central approximation gives



At the bottom the kinematic boundary condition is used for the vertical velocity:



Horizontal interpolation of  and  is done using first order upwind similar to . The turbulent stresses are again approximated using a central scheme as



Thus combining, and explicit expressions for  and  are obtained.

The predicted values are again corrected using a variant of the McCormack scheme and including the pressure difference implicitly gives the corrector step:



Where  and  are obtained using relations similar to . Note that similar to and again the kinematic boundary conditions is substituted for .

The discrete vertical momentum balance of and looks very different from the relations found in Zijlema and Stelling (2005), Zijlema and Stelling (2008) and Smit (2008). This is mainly due to the application of the McCormack scheme for the advection. The discretisation of the pressure term is numerically fully equivalent to either the Keller box scheme as used in Zijlema and Stelling (2005), Zijlema and Stelling (2008) or the Hermetian relation used in Smit (2008).