Keywords

Place keywords here

Summary

Place summary here

References

Place references here

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Version | Date | Author | Initials | Review | Initials | Approval | Initials |
|  | jan. 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 6

2.4 Shallow water equations 6

2.5 Nonhydrostatic pressure correction 6

2.6 Groundwater flow 6

2.7 Bedload transport 6

2.8 Suspended load transport 6

2.9 Bottom updating 6

2.9.1 Due to sediment fluxes 6

2.9.2 Avalanching 6

2.9.3 Bed composition 6

3 Numerical implementation 7

3.1 Grid types 7

3.1.1 1D 7

3.1.2 Rectilinear 7

3.1.3 Curvilinear 7

3.2 Wave action balance 7

3.2.1 Stationary solver 7

3.2.2 Nonstationary solver 7

3.3 Shallow water equations 7

3.4 Nonhydrostatic pressure correction 7

3.5 Advection-diffusion equation for sediment 7

3.6 Bottom updating schemes 7

3.7 Avalanching 7

3.8 Bed composition 7

4 Boundary conditions 8

4.1 Waves 8

4.1.1 Time series 8

4.1.2 Spectra 8

4.1.3 Lateral boundary conditions 8

4.2 Shallow water equations 8

4.2.1 Absorbing-generating 8

4.2.2 River and point discharge 8

4.2.3 Ship motion 8

4.2.4 Lateral boundaries 8

4.2.5 Tide and surge 8

4.3 Sediment transport 8

5 Input description 9

5.1 General 9

5.2 Physical processes 10

5.3 Grid and bathymetry 11

5.4 Waves input 14

5.4.1 Stationary wave boundary conditions 16

5.4.2 Spectral wave boundary conditions 17

5.4.3 Boundary conditions for non-hydrostatic model 24

5.4.4 Special types of wave boundary conditions 24

5.4.5 Temporally and/or spatially varying wave boundary conditions 24

5.4.6 Notes on the generation of wave boundary conditions 25

5.5 Tide and surge input 26

5.5.1 Tide boundary conditions 26

5.5.2 Flow boundary conditions 26

5.6 Water level (dam break) 28

5.7 Wind input 29

5.8 Sediment input 30

5.9 Vegetation input 31

5.10 Discharge input 31

5.11 Drifters input 32

5.12 Ship input 32

5.13 Output selection 32

5.14 Time parameters 34

5.15 Model coefficients 35

5.15.1 Wave numerics 35

5.15.2 Wave dissipation 35

5.15.3 Rollers 36

5.15.4 Wave-current interaction 36

5.15.5 Flow 37

5.15.6 Flow numerics 38

5.15.7 Sediment transport 38

5.15.8 Sediment transport numerics 42

5.15.9 Quasi-3D sediment transport 42

5.15.10 Morphology 42

5.15.11 Bed update 43

5.15.12 Groundwater flow 44

5.15.13 Non-hydrostatic correction 45

5.15.14 Physical constants 47

5.15.15 Coriolis force 47

5.15.16 MPI 47

5.15.17 Output projection 47

6 Bibliography 49

7 Tutorial 50

7.1 1-D profile model 50

7.2 2-D area model 50

7.3 Langsgetij + riveroutflow 50

# 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

Kees - bezig

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 (*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 (*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

#### Breaking

Kees - bezig

There are in four different wave breaking formulations 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 of break=1
4. Non-stationary waves: adaptation of break=1 (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=1*. In *α* is applied as wave dissipation coefficient, *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 a 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=3.*



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=4*.



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=4*.



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

Kees

#### Vegetation

Arnold

### Roller energy balance

Dano

## Shallow water equations

Kees

## Nonhydrostatic pressure correction

Robert

## Groundwater flow

Kees/Robert

## Bedload transport

Kees + Lodewijk

## Suspended load transport

Kees + Lodewijk

## Bottom updating

### Due to sediment fluxes

Kees

### Avalanching

Kees + Pieter

### Bed composition

Bas

# Numerical implementation

Dano behalve 3.4,3.8

## Grid types

### 1D

### Rectilinear

### Curvilinear

## Wave action balance

### Stationary solver

### Nonstationary solver

## Shallow water equations

## Nonhydrostatic pressure correction

Robert

## Advection-diffusion equation for sediment

## Bottom updating schemes

## Avalanching

## Bed composition

Bas

# Boundary conditions

## Waves

### Time series

Kees, Ap review

### Spectra

Kees, Ap review

### Lateral boundary conditions

Dano

## Shallow water equations

### Absorbing-generating

Ap met appendix

### River and point discharge

Bas

### Ship motion

Dano

### Lateral boundaries

Kees

### Tide and surge

Kees

## Sediment transport

Dano

# Input description

Bas - params en attribute files

## 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 5.15.11 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 5.15.1 Wave numerics) and wave-current interaction (5.15.4 Wave-current interaction), wave dissipation model (see 5.15.2 Wave dissipation) and wave roller model (5.15.3 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 5.4.5 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 |

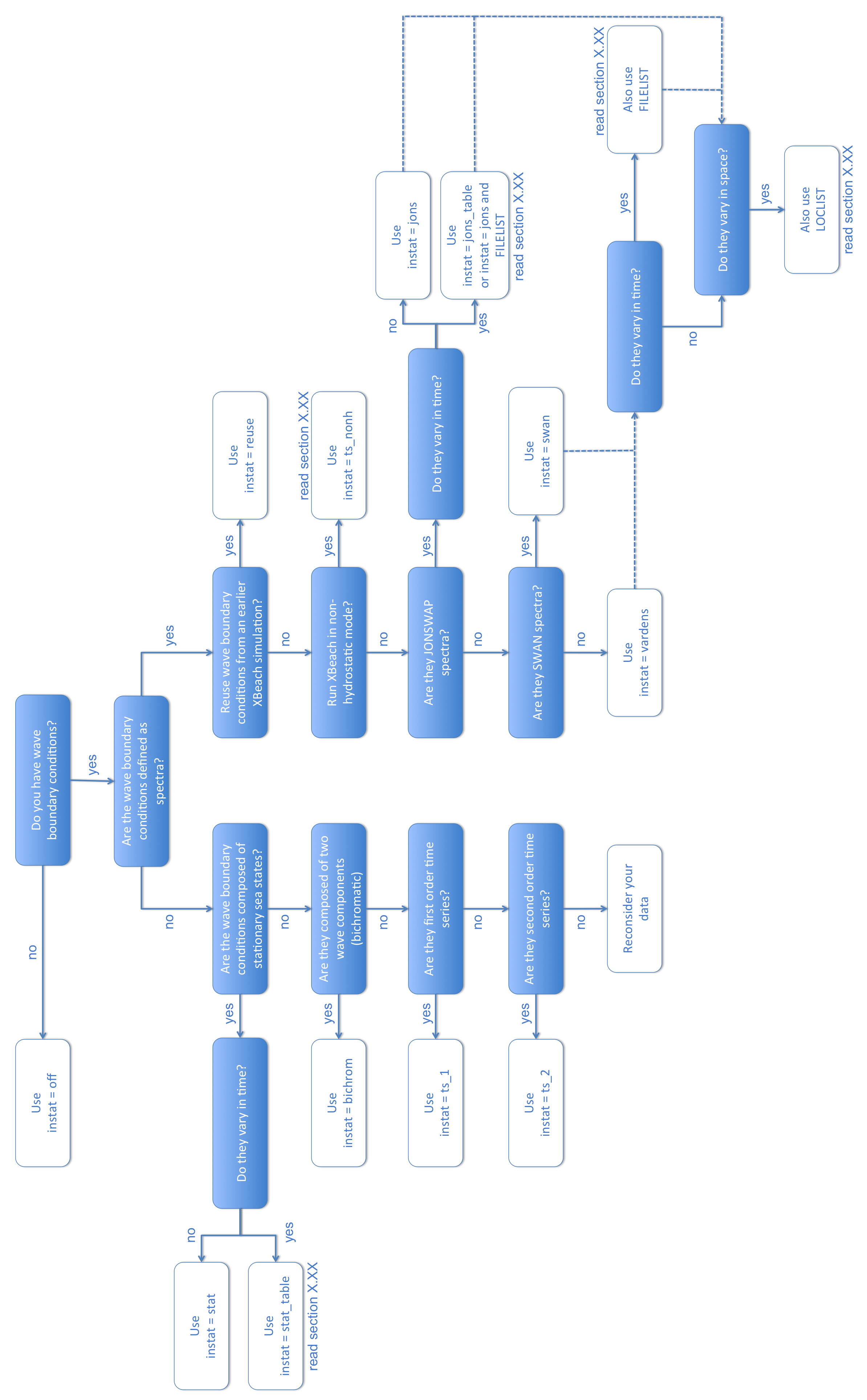


Figure XXX Decision tree for selecting the appropriate type of wave boundary conditions

### 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 5.4.2.1 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 5.4.1 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 5.4.5 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 5.4.5 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 5.4.5 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 5.15.5 Bed friction and viscosity) and flow numerics (5.15.6 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 5.4.1 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 5.5.2 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 5.3 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 5.15.7 Sediment transport and 5.15.8 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*.

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 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 cell 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. 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 5.3 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

Files: species, maps

| 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>

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

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>

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

Files: timeseries

| 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 5.13.2 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 5.13 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' |  | - |  |

## Model coefficients

The previous sections of this chapter described the main input parameters and files required by XBeach to start a simulation. 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 action balance that solves the wave propagation in the model. The keyword *scheme* can be used to set the numerical scheme. By default the higher-order Lax Wendroff 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* or *break=roelvink2*. 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, viz. the 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 the dimensional Chézy value *C*. Alternatively, spatially varying Chézy values for the bed friction can be provided through an external file referenced via the *bedfricfile* keyword. The file has the same format as the bathymetry file explained in 5.3 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.

| 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 behaviour. *hmin* prevents very strong return flows or high concentrations. *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, especially important near-shore. *facua* is an alias setting both parameters 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.

| 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.

| 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 from 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 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* enable the user to enable the morphological processes in XBeach only for a particular period during the (hydrodynamic) simulation. These options are useful 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 5.3 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. 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 groundwater flow. The vertical permeability coefficient in the vertical can be set differently to that in 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 5.3 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. These are all considered advanced options.

| 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.

| 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.

| 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 parallelisation of XBeach. A full description of the parallelisation 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, increasing the computational speed of the model. The submodels only exchange information over their boundaries when necessary. These 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).

| 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 | - |  |
|  |  |  |  |  |  |

# 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.

# Tutorial

Nog niet verdeeld. Later nog in te vullen.

## 1-D profile model

Delfland Deltagoot

## 2-D area model

Ocean bay park: getij+surge, baai, duin, nonerodible, overwash, collision,

## Langsgetij + riveroutflow

getijmodel + rivier + stationair.