# Introduction

History: HISWA, STWAVE, SWAN, XBeach stationary

Need for simple physics, flexible grids. Forward marching not sufficient. Single-dir.

Wind growth important but not yet included.

# Model description

## Wave energy balance

### Basic equations

### Discretization and solution scheme

The wave energy balance as solved in XBeach reads:



where *ee* is the spectral energy density, *Cg* the group velocity, the wave direction and *dd* the wave dissipation density.

We can write this in simpler form:



Here, *s* is the distance along each wave direction. This can be discretized as follows:



where *k* is the grid number, the direction bin number, *n* the timestep number, *D* the integrated dissipation, *E*  the integrated wave energy. The subscript *prev* refers to the point upwind of grid point *k,* as illustrated in fig XX. Group velocity and energy density in point *prev* are computed as follows:



Computation of the tables *w*  and *prev* for each mesh node and directional bin is carried out once, and can be done for any grid type as long as it is clear how the nodes are connected. In the case of SnapWave standalone a Delft3D-FM unstructured network is used (UG format).



We can write the system of equations per grid point as:



Here, the coefficients are given by:



This is a tridiagonal system with the dimension of ntheta that can be efficiently solved for each point using a standard algorithm. The solution for each point relies on having (ideally converged) estimates of the wave energy density *ee* in the upwind points for each wave direction. Obviously, this works best when the points are solved after ordering by the main wave direction. Secondary effects of refraction are covered by ‘sweeping’ in all 4 directions. Since the wave dissipation is a very nonlinear function of the wave height and water depth the whole system needs to iteratively come to a converged solution.

## Implementation in SnapWave standalone model

### Input file SnapWave.inp

The SnapWave input file is keyword-oriented, where the order of the keywords is irrelevant and only lines with an ‘=’ are interpreted, any other lines can be comments.

In the table below an example input file is shown.

**Time parameters**

Here, *tref* is the reference date and time, relative to which all times are defined. The start and stop time of the simulation are defined by *tstart* and *tstop.* The *timestep* in seconds defines the time interval between the (stationary) wave fields computed.

**Network parameters**

The unstructured grid file name is defined by the *gridfile* keyword; no parentheses are required for the file name. Keyword *sferic* determines whether the coordinates are in a metric (0) or geographical system (decimal degrees lon/lat). *dtheta* and *ntheta* determine the directional resolution; for now, this has to remain unchanged. For each wave computation the directional grid is centered around the mean wave direction at the offshore boundary.

**Boundary information**

Boundary conditions can be provided uniformly or at a number of points near the offshore boundary. The location of the boundary condition points is given in *bndfile.* The grid points where boundary conditions are imposed are 1) at the outer edges of the grid, which may be straight , curving or staircase-like, and 2) within the polygon defined by *encfile.* In other words, the polygon in *encfile* must enclose the network boundaries where the offshore conditions can be imposed. At each valid boundary point the wave conditions are determined by interpolation between the two nearest boundary condition points.

The actual time series of boundary conditions are given in *bhsfile* for Hm0 (m), *btpfile* for Tp (s), *bwdfile* for mean wave direction (deg. N), *bdsfile* for directional spreading (deg.) and *bzsfile* for mean water level (m). For each of these files, the format is a column of time in s, followed by columns of the parameter at all boundary condition points.

**Model settings**

The most relevant model settings are *gamma*, the breaker parameter in the Baldock formulation, *hmin*, the minimum water depth, and *restart*, which determines if each consecutive wave run continues on its last solution or not.

**Output information**

The output locations are defined in *obsfile,* as two x, y or lon, lat columns. For these locations, the Hm0, Tp, wave direction and directional spreading are written to the netcdf *his\_file.* Complete maps of these parameters are written to the netcdf *map\_file.*  For both, if the filename is left empty then no his or map file is created.

|  |
| --- |
| Time parameters tref = 20181007 000000 tstart = 20181007 000000 tstop = 20181007 020000 timestep = 600.0  Network parameters gridfile = ugdcsm-fm\_100mgrid\_netherlands\_era5bnd\_net.nc sferic = 1 dtheta = 10.0 ntheta = 18.0  Boundary information bndfile = dcsm\_nl\_pnts.txt encfile = bndenc\_nl.txt bhsfile = dcsm\_hs.txt btpfile = dcsm\_tp.txt bwdfile = dcsm\_wd.txt bdsfile = dcsm\_ds.txt bzsfile = dcsm\_zs.txt  Model settings gamma = 0.75 hmin = 0.1 restart = 1  Output information obsfile = obspoints.txt his\_file = snapwave\_his.nc map\_file = snapwave\_map.nc uuupwindfile = snapwave.upw |

### Example DCSM cutout for the Netherlands

The example shown in Figure 1 concerns a network that was cut out of the Dutch Continental Shelf Model (DCSM) model, in a fine resolution down to 100m. The boundaries do not have to follow regular straight or curved lines but can be arbitrarily cut out, roughly along the lines connecting boundary support points. An enclosing polygon has to be provide as indicated, to limit the boundary points to those on the outer edge of the mesh that are at sea,



Figure 1 Cutout of DCSM-fine grid for the Netherlands (mesh and bathymetry); boundary enclosure (blue polygon); ERA5 output points used as boundary support points (red circles).

**Validation**

Linear sho aling

Circular island

LSTF

Poort?

Haringvliet mouth

Buck Island

River Outflow

**Discussion**

Performance

Flexibility

**Conclusions**