XBeach grid tutorial

Joost den Bieman

June 3, 2013

Contents

1 Introduction **2**

2 XBeach Grid Definitions **3**
2.1 Model grid 3
2.1.1 Model coordinates 4
2.1.2 World coordinates 4
2.2 Wave energy discretization 5
2.3 Vectorial quantities 5
2.4 Overview 5

3 Grid creation tutorial **7**
3.1 Visualising model input 7
3.2 Running the models 8
3.3 Analysing model results 8
3.4 Model setup 9
3.4.1 Grid creation 10

Chapter 1

Introduction

This tutorial describes two different methods for creating a rectilinear XBeach model grid. Chapter 2 describes the different methods and associated XBeach keywords and parameters. Chapter 3 contains examples for each method.

The examples featured in this tutorial make use of the XBeach toolbox for Matlab. This toolbox is part of the OpenEarth [van Koningsveld et al., 2010] toolbox, which can be found here: `http://publicwiki.deltares.nl/display/OET/Join+OpenEarth` (register for an account, then checkout using subverion).

Chapter 2

XBeach Grid Definitions

2.1 Model grid

The XBeach model [Roelvink et al., 2009] makes use of a recti- or curvilinear computational grid. In this tutorial, the focus will be on rectilinear grids. A rectilinear grid can be defined in two ways:

1. As a grid relative to a known origin
2. As a grid in a meter-based coordinate system (e.g. UTM)

Both methods are explained in this chapter. For both methods, the x and y coordinates of the grid points are specified in seperate files, using the keywords *xfile* and *yfile* respectively. These keywords contain the names of ASCII files with a $m * n$ matrix (see equation 2.1) with x- or y-values, corresponding to the bed level elevation information specified in the *depfile*. The size of both dimensions of the matrix is 1 element larger than specified in *nx* and *ny* ($nx = n - 1, ny = m - 1$), because the model needs an extra row and column of 'dummy' grid points. In the *xfile* and *yfile*, the first element of the matrix (x_{11}) contains the coordinates of the grid origin (this does not have to be 0,0). Furthermore, the leftmost column ($x_{11} \ldots x_{m1}$) always has represents the coordinates of the offshore boundary (thus $x_{11} \ldots x_{1n}$ and $x_{m1} \ldots x_{mn}$ are the two lateral boundaries, and $x_{1n} \ldots x_{mn}$ is the landward boundary).

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & \ddots & & \\ \vdots & & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix} \tag{2.1}$$

The model assumes Neumann conditions on the lateral boundaries (no gradient in velocity and water level). These assumptions are more robust when there is no (alongshore) gradient in the bed level on the cells close to those lateral boundaries. The *xb_grid_finalise.m* function takes care of this by adding a few grid cells to each lateral boundary with the same bed level as the former boundary cell.

2.1.1 Model coordinates

A sketch of the definitions used in XBeach is given in figure 2.1.

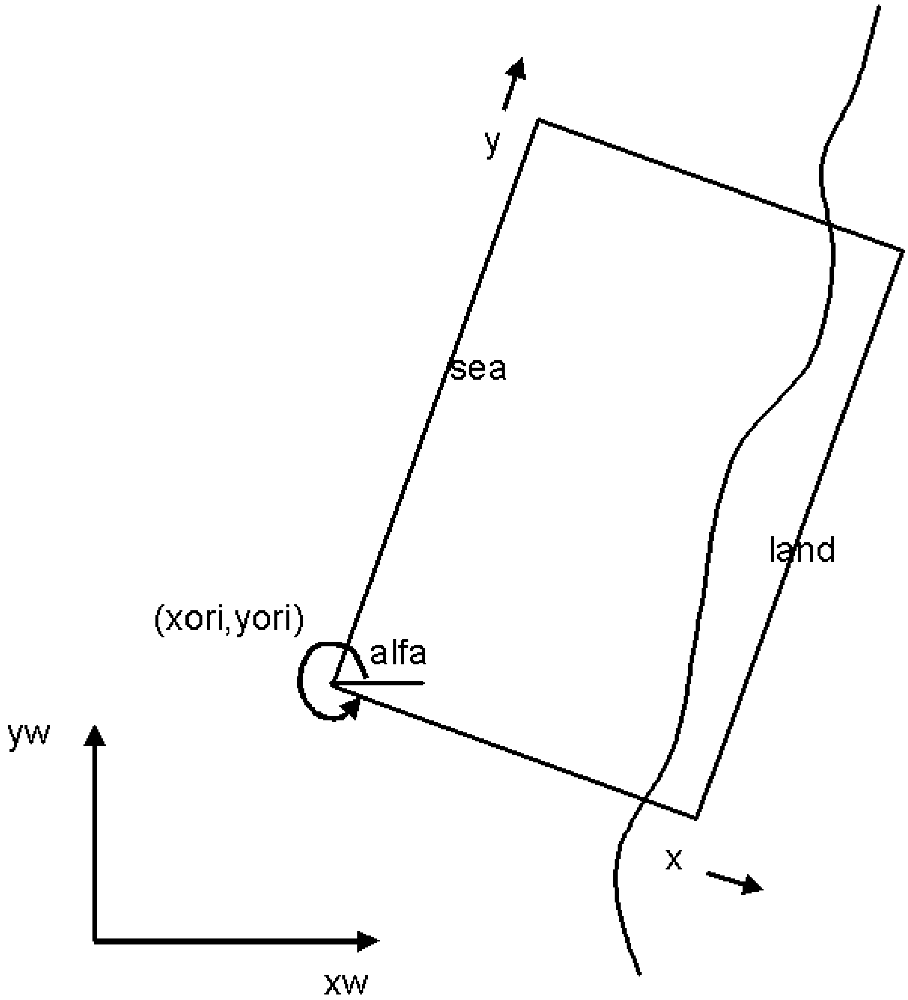

Figure 2.1: XBeach model grid definitions.

In the XBeach coordinate system, the origin is located on the offshore model boundary, the x-axis is positive in landward direction and the y-axis is pointed alongshore. The XBeach model coordinates are related to a coordinate system by specifying the coordinates of the origin (using keywords *xori* and *yori*) and the angle *alfa* between the East and the direction of the x-axis (defined counter-clockwise).

2.1.2 World coordinates

When using world coordinates, the *xfile* has to contain the x-world coordinates and the same holds for y direction. This means that these x- and y-axes do not necessarily meet the requirements specified in section 2.1.1 (x-axis pointed shoreward, y-axis pointed alongshore, etc.). This is no problem, as long as x_{11} contains the origin and $x_{11} \ldots x_{m1}$ define the offshore boundary (see equation 2.1). However, be aware that the output parameters related to a specific direction are actually related to the directions of the grid axes, so their actual orientation depends on how the grid is defined. For instance the wave forces *Fx* and *Fy* are related to the *xfile* and *yfile* respectively. This means that *Fx* isn't necessarily in shoreward direction and the same holds for *Fy*.

When using world coordinates, *xori*, *yori* and *alfa* do not have to be used (can all be set to 0), since no rotation or translation is needed.

2.2 Wave energy discretization

In the XBeach model, the wave energy is discretized in directional bins. These bins are defined by the upper and lower directional limits (*thetamin* and *thetamax* respectively), combined with the resolution of the bins, *dtheta*. All three are specified in degrees. By default, they are interpreted w.r.t. East, counterclockwise (East = $0°$, North = $90°$) and internally, these angles are rotated by *alfa*. However, they can also be specified adhering to the nautical convention (North = $0°$, East = $90°$) by setting *thetanaut* to 1, which ignores the *alfa* value.

When simulating either oblique waves or varying wave direction, the directional discretization becomes relevant. The *thetamin* and *thetamax* should be defined such that the simulated wave directions remain within these bounds (also accounting for spreading). Furthermore, the required directional resolution depends on the required amount of detail.

2.3 Vectorial quantities

XBeach knows several vectorial quantaties, examples of which are the velocities u and v and the wave forces *Fx* and *Fy*. These quantities are specified relative to the x- and y-directions. Be aware that, when using world coordinates, the x- and y-directions are often not aligned with the direction of the grid cells.

2.4 Overview

An overview of all above mentioned grid related properties can be found in figure 2.2.

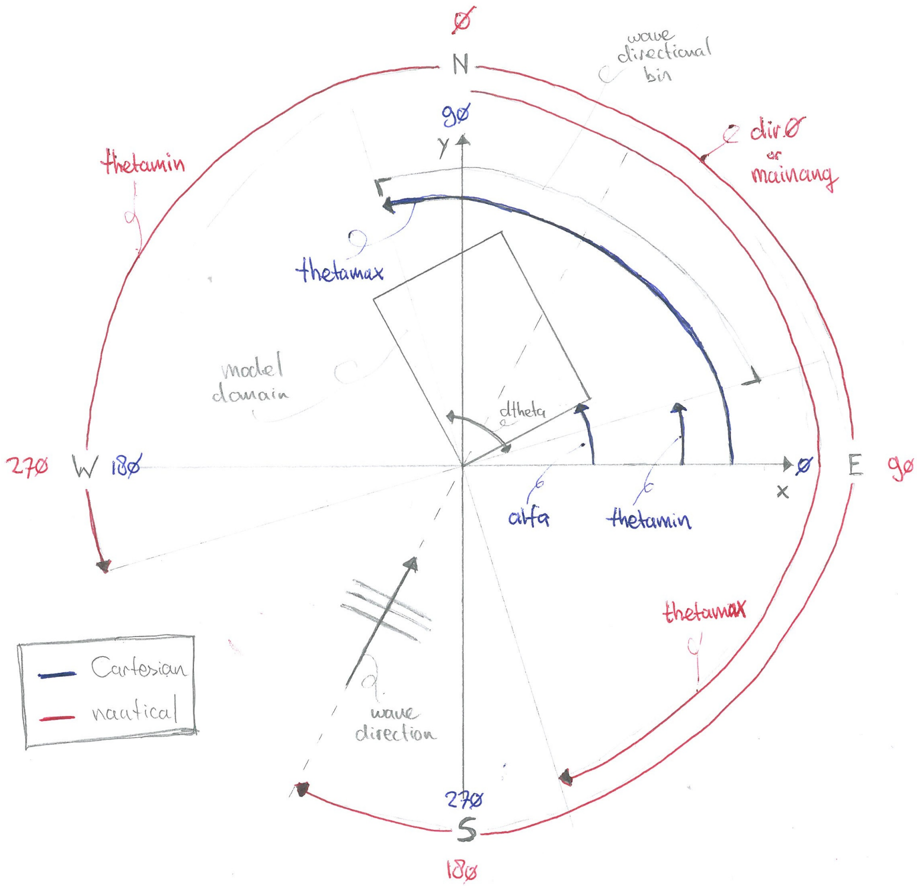

Figure 2.2: XBeach grid orientations, by B.M. Hoonhout.

Chapter 3

Grid creation tutorial

The examples in this tutorial show simple model of part of the barrier island Terschelling in the Netherlands, see figure 3.1. The two models are identical, except that one grid is defined in XBeach model coordinates and the other in the Dutch RD coordinate system. Note that the bathymetric data set does not extend too far inland, so the missing values have been replaced with a dummy value of 1.5 m.

Figure 3.1: Terschelling, from `www.bing.com/maps/`.

The files needed for this tutorial can be downloaded from: `dummy.url`

The XBeach binary can be downloaded from: `http://oss.deltares.nl/web/xbeach/source-code-and-exe`. After downloading, it should be extracted in the *xbeach_binary* directory (within the tutorial directory). The scripts in this tutorial make use of relative paths, so make sure the scripts are run with the tutorial directory as working directory. Additionally, the scripts make use of the XBeach tools in the OpenEarth toolbox, so *oetsettings.m* needs to be run before running any of the scripts in the tutorial.

3.1 Visualising model input

After downloading, the input files for both the model coordinates and world coordinates simulations are already present. The difference between the models becomes evident when the initial bathymetry is plotted, see figure 3.2. The *visualize_input.m* scripts plots and saves this figure in the tutorial folder.

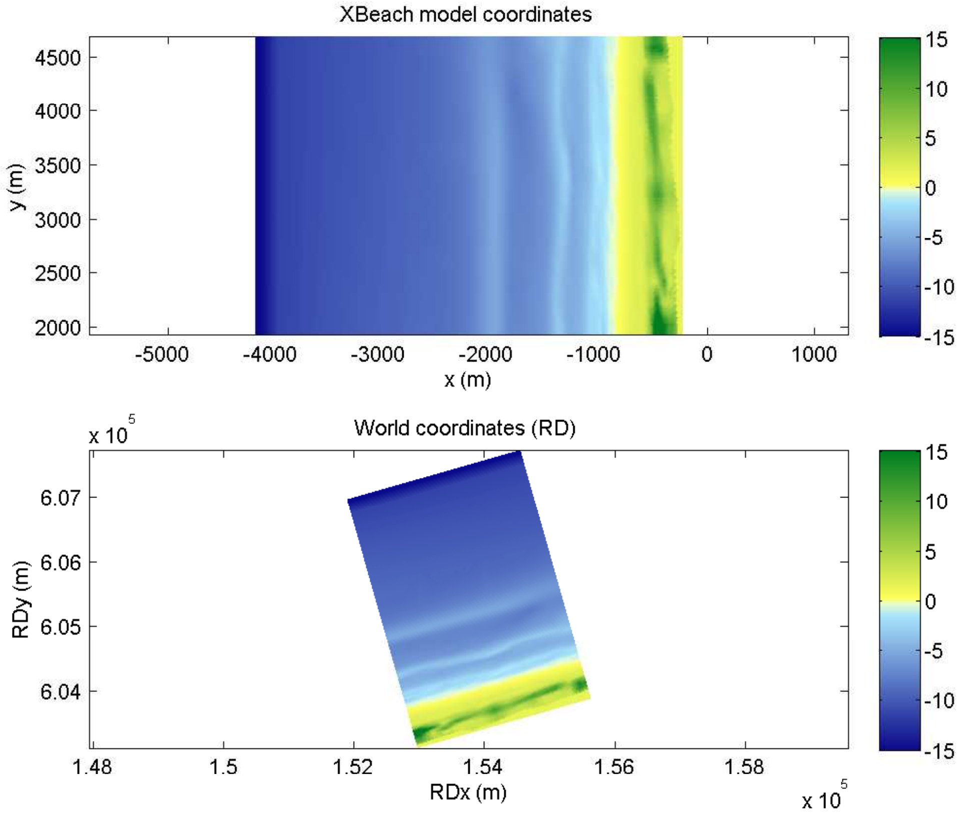

Figure 3.2: XBeach input grids and bathymetries.

Alternatively, *xb_visualize_modelsetup.m* is a generic script to visualize the grid of a 2D XBeach model, including the directional wave grid and incoming wave directions.

3.2 Running the models

The simulations can be started by running the *run_models.m* script. Alternatively, the *run_model.bat* file can be used to run each model without needing Matlab. Both methods depend on the XBeach binary being in the *xbeach_binary* directory.

3.3 Analysing model results

Both models should give the same results, despite the different ways of specifying the grid (note that this is only true when *random* is set to 0). To verify this, run the *analyse_results.m* script, which compares the wave height and water level for the same grid line in both models (see figure 3.3). As can be seen from the figures, the hydrodynamic results are the same.

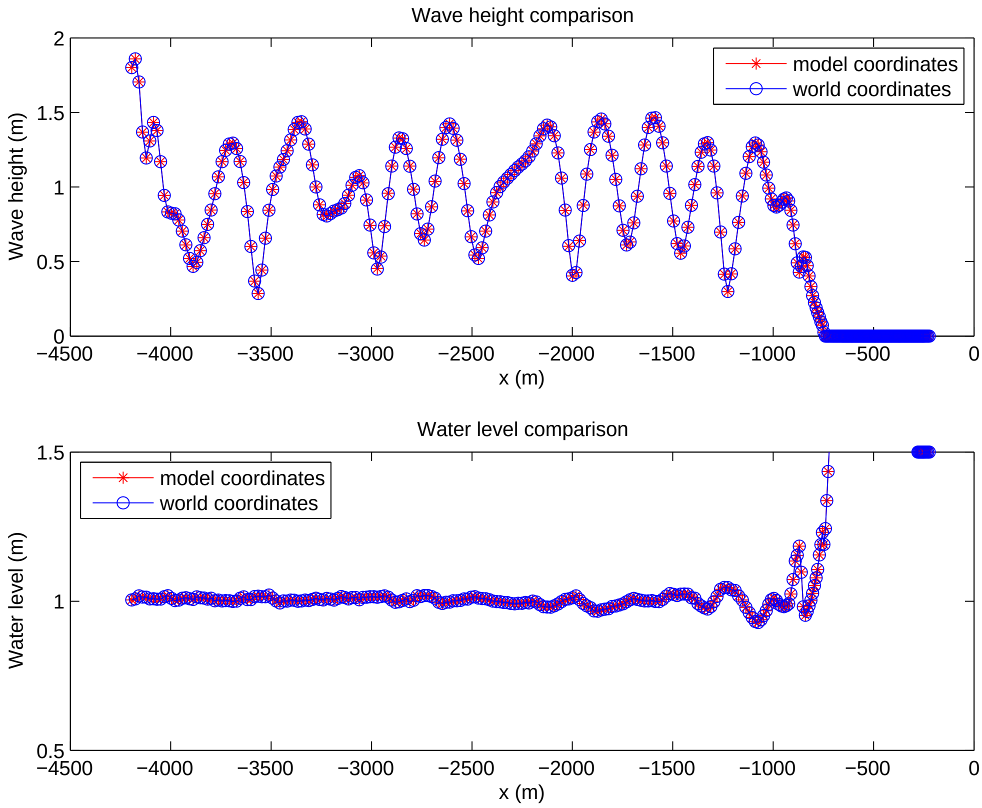

Figure 3.3: Comparison of wave height and water level in both models.

The *xb_view* script allows for visualization all output variables from a single model run and is an easy way to make a first assessment based on the simulation results.

3.4 Model setup

The script to create both models from the bathymetric data is included in the *generate_models.m* script, to allow for easy reproduction and adaptations. The script uses the *xb_generate_model* function to create the necessary XBeach input files and write them to a specified path. The model settings can be modified by using keyword-value pairs in the function arguments. The only difference between these two models is that the *world_coordinates* keyword is set to *false* when using model coordinates and *true* when using world coordinates. For a description of the general workings of the functions in the XBeach Matlab toolbox, see the help text included in the respective functions.

3.4.1 Grid creation

To create a model using other bathymetric or hydrodynamic data, load the data into Matlab and adapt the input of the *xb_generate_model* script to use that data instead. Be aware that the *crop* keyword should be set to *'select'*, so the extent of the model domain can be defined by selecting the opposite corner points.

Within the *xb_generate_model* script, an optimized model grid is created based on the bathymetry supplied. This is done by the *xb_grid_xgrid* and *xb_grid_ygrid* scripts. The parameters governing optimizing the grid are explained in the help text of both functions (accessible by typing *help xb_grid_xgrid*).

Bibliography

[Roelvink et al., 2009] Roelvink, J. A., Reniers, A. J. H. M., van Dongeren, A. R., van Thiel de Vries, J. S. M., McCall, R. T., and Lescinski, J. (2009). Modelling storm impacts on beaches, dunes and barrier islands. *Coastal Engineering*, 56(11-12):1133–1152.

[van Koningsveld et al., 2010] van Koningsveld, M., de Boer, G., Baart, F., Damsma, T., den Heijer, C., van Geer, P., and de Sonneville, B. (2010). OpenEarth - inter-company management of: data, models, tools & knowledge. In *Proceedings WODCON XIX Conference. Beijing, China.*