XBeach Course

31 October 2016
Delft, The Netherlands

1 GETTING STARTED

The course materials contain the XBeach executable, an export of the Matlab toolbox, model examples and some Deltares software (RGFGRID, Quickin and Quickplot) that will help us to analyze and modify model (results)

2 Using the XBeach toolbox (Matlab)

In the following exercises we will make use of the Matlab routines that are part of the XBeach toolbox. The tools can be used to set up and run an XBeach model, as well as to analyze and visualize simulation results. The toolbox is open source (like XBeach) and can be downloaded freely. The only requirement is to have a recent version of Matlab installed on your pc (preferably version 2009a or more recent).

There are two ways to download the toolbox. The toolbox is included in the Open Earth repository; see the website for more information (www.openearth.eu). The toolbox can also be downloaded (as stand-alone version) from the XBeach website (www.xbeach.org or via http://oss.deltares.nl)

Note that all Matlab routines have a help function, which describes the input and output of a specific routine. The help function works like for any other Matlab function (type in the Matlab command window: “help *functionname.m*”). Examples of how to use the functions and some tips and tricks can be found on the XBeach website (www.xbeach.org).

3 HANDS-ON EXERCISE: Storm impact at Boscombe Beach (U.K.)

In this exercise we will set up a storm impact model from scratch with use of the XBeach toolbox. The case concerns Boscombe beach, located at the south coast of the U.K, and we will use (measured) data of the local bathymetry, hydrodynamic conditions and some other characteristics to set up a storm impact model.

We have already downloaded the toolbox for you, so you only need to add the toolbox paths to your current Matlab session. This can be done by running ‘oetsettings.m’, which can be found in the ‘Toolbox’ directory. After doing this, you are ready to start constructing your first XBeach model from scratch!

You can work on the following assignments:
1. Open a new empty script in the Matlab editor. We will use this script to write down all steps in setting up the model. Don’t forget to save it once in a while!
2. We will start by constructing a grid and bathymetry. In the simulation directory you find a measured bathymetry (‘bathy.dep’). Load the bathymetry file in your Matlab workspace

(with the 'load' command). Plot the bathymetry to check what the bathymetry looks like (use both the 'pcolor' and 'surf' command). What is the maximum dune height? And what is the offshore water depth?

3. The bathymetry is projected at an equidistant grid with dx = 5m and dy = 20m. Now construct two arrays (for cross-shore and alongshore) with the gridpoint locations. (e.g. x = [0:1:nx-1]*dx, and similar for y). Tip: the amount of grid cells (nx and ny) can be found by looking at the amount of points in the depth file.
4. Now plot the bathymetry again, but including the grid information (e.g. surf(x,y,depth)). What's the size of the model domain (cross-shore and alongshore length)?
5. To reduce the computational effort we would like to have a lower grid resolution at deeper water. However, we do want to keep the minimum grid size (5m) near the coast to solve the morphodynamics accurately enough. To do this we can make a variable grid spacing in cross-shore direction. The tool 'xb_grid_xgrid.m' can be used for this, and will make a new cross-shore grid spacing based on the Courant condition, and the mean wave period at the offshore boundary. Check the help function to see what the tool requires as input and what the output of the tool is.
6. Assume a mean period of 8 seconds and construct a new x-array with your grid point locations in cross-shore direction. (Tip: you can only use this tool for a 1D profile, so you need to select a characteristic cross-shore profile from your bathymetric data).
7. Plot your original cross-shore profile (the one you selected to be representative) and plot the new cross-shore profile. Check if the profile did not change, and check the change in grid size spacing (use markers in your plot). What is the grid size at the offshore boundary?
8. We can also vary the grid spacing in alongshore direction. This is often useful when you want to focus on a local hotspot, but to keep the computational effort still manageable. Another reason why we would like to vary the grid spacing in alongshore direction is that the number of grid cells covering shadow zones due to oblique waves can be reduced by this approach. Check the help file of the tool 'xb_grid_ygrid.m', and try to understand how the tool works.
9. We would like to put some more resolution (dy = 10m) in the middle of the alongshore domain (40%). The maximum alongshore grid size can be kept default (dy = 20m). Construct your new y-array with the tool 'xb_grid_ygrid.m'.
10. Interpolate the bathymetry on your new grid using the interp2-function.
11. Now construct your grid in matrix form (rather than two arrays for x and y), use the meshgrid function for this.
12. Now plot your new bathymetry with the surf command and compare with the plot you made before of the original bathymetry. Also look at the amount of grid points (in your Matlab workspace), did the number of grid cells increase/decrease?
13. The offshore water depth is rather small, and it will possibly affect the boundary conditions (waves and flow) too much. Therefore we want to extend our model domain until a larger depth. We also would like to extend the model a bit in lateral direction to obtain an alongshore uniform coastline at the boundary (because of the Neumann boundaries). The tool 'xb_grid_finalise2.m' can be used for extending the model domain. The routine reads in the grid information (xgrid and ygrid) and the bathymetry, and will finalize the grid/bathymetry based on some user defined settings. Check the help function of this tool and see what the possibilities are for finalizing our model domain.
14. We will use the 'lateral_extend' and the 'seaward_extend' options. Specify an offshore water depth of 15m, with a slope of 1/50.
15. Plot your new model domain with the surf command and check whether you are happy with the result. If not, change some settings in the finalize-tool.
16. Next we will work on the hydrodynamic boundary conditions. In the simulation directory you can find a file which contains the tide boundary condition (tide.txt), and a file for the wave boundary condition (waves.txt). These files are already in the right layout for XBeach. Open

the files and check the input. What is wave height? And what is the maximum water level difference throughout the simulation?

17. The params.txt file is the file that contains all information (settings, boundary conditions etc.) for an XBeach simulation. Using the tool 'xb_generate_settings' we can create a params.txt file very quickly. The tool works by filling in keywords and corresponding value (e.g. pars_struc = xb_generate_settings ('dx',0.5,'dy',20 etc.), and will create a structure with all information. In this case we want to specify the following keywords:
    - origin of the grid (xori=412500 and yori=90700)
    - wave grid (thetamin = -90, thetamax = 90, dtheta = 20)
    - wave boundary condition option (instat = 'jons_table', bcfile = ????)
    - water level boundary conditions (zs0file = ????, tideloc = 1)
    - morphological acceleration (morfac = 5)
    - output settings (outputformat ='netcdf', nglobal = {'H','zs','zb','u','v'}, tintg = 100)
    - time management (tstart = 0, tstop = ????, morstart = 100)

    Do you know what all the specified keywords stand for? (Maybe you can take a look in the XBeach manual?)
18. All we need to do now is merge the grid and bathymetry information with the settings information. We can do this with 'xs_join', but we will first need to make a structure for the grid and bathymetry data: bathy_struc = xb_grid_add (x,y,z).
19. Now merge the two structures: xb_struc = xs_join(bathy_struc, pars_struc).
20. Finally save your model with xb_write_input(filename,xb_struc). Note that the filename should always be params.txt (but it can contain a full link, e.g. d:/xbeach/boscombe/run1/params.txt).
21. Run the model, and check some output using Quickplot (bed level changes, wave propagation etc.). Is there a lot of erosion/accretion occurring?
22. If not much is happening, think of a way to get some more impact and update your model.

Bonus assignment: Using boundary conditions from an operational model (COSMOS)

Up till now we have had ready-to-use boundary conditions. In this part of the exercise we will use data from an operational model (COSMOS) to define our boundary conditions. You can add the code for the following exercise to your existing code, or start a new script.

23. Make a new simulation directory and copy your Boscombe storm impact model to the new directory.
24. Read in the current model in Matlab using the xb_read_input tool. Make sure you save the model in an XBeach structure: xb_si = xb_read_input('params.txt').
25. We will now download some netcdf files from the internet. Copy the following code into your Matlab editor:

```
% Url where the data can be found
  url =
  'http://opendap.deltares.nl/thredds/dodsC/opendap/deltares/cosmos/f
  orecasts/csm/';

% Netcdf files
nc_wl = 'boscombe.wl.2012.nc';
nc_hs = 'boscombe.hs.2012.nc';
nc_tp = 'boscombe.tp.2012.nc';
nc_dir = 'boscombe.wavdir.2012.nc';

% Read the time
time = nc_varget([url nc_wl],'time');
time = datenum(1900,1,1)+time/3600/24;
```

```
% Read variables
wl   = nc_varget([url nc_wl],'wl');
hs   = nc_varget([url nc_hs],'hs');
tp   = nc_varget([url nc_tp],'tp');
dir  = nc_varget([url nc_dir],'wavdir');
```

26. If you run this code it will get the data from the so-called OpenDAP server and will give you timeseries of water levels (wl), wave heights (hs), peak period (tp) and wave direction (dir). If you have no internet available, then load the egmonddata.mat –file into your Matlab.
27. Modify the wave direction by substracting 100 degrees from the Egmond wave direction.
28. Now plot all four variables in one plot and check what the hydrodynamic conditions look like. What was the maximum wave height?
29. Also check the time vector, what is the time step (dt) in the timeseries?
30. Select a period in the timeseries which you want to model in your storm impact model (for example the first hour, or a few hours with high wave heights etc.).
31. Now we will generate a tidal water level signal. Use the tool xb_generate_tide for this. You will have to provide an array with time and one with water level elevation: for example tide = xb_generate_tide (‘time’, [time],’front’,[water level]). Select the period you find interesting and make sure you specify the time in seconds (!).
32. Next we will specify the wave boundary conditions. Use the tool xb_generate_waves and define the wave input as follows. For every specified wave condition you need to specify a duration, an easy way to do this automatically is shown here:
    waves = xb_generate_waves(‘Hm0’,[wave height: hs],’Tp’,[peak period: tp], ‘mainang’,[wave angle: dir_boscombe], ‘duration’,[repmat (dt, nt,1)]). The duration input will now be a vector with a length equal to the number of timesteps with value dt (in seconds!).
33. Now make sure the right stop time is included in the params.txt: pars_si_Cosmos = xb_generate_settings(‘tstop’,[simulation time in seconds]).
34. Now join all your xb structures to a new storm impact Cosmos model structure:
    xb_si_cosmos = xs_join (xb_si,waves,tide,pars_si_Cosmos)
35. Save the model with xb_write_input and run it.
36. Check your results with Quickplot.

Bonus assignment: BeachWizard Model

We would now like to make a new model for the same case, but now a Beach Wizard (BW) model instead of a storm impact model. More information about the BW system can be found at the bottom of this exercise. The main differences between a storm impact and a BW model is that the BW model is run with a stationary wave condition (while in a storm impact model wave groups are simulated), without morphology, and usually for a longer simulation period. You can add the code for the following exercise to your existing code, or start a new script.

37. Make a new simulation directory and copy your Boscombe model (except for the output files) to the new directory.
38. Read in the model in Matlab using the xb_read_input tool. Make sure you save the model in an XBeach structure: xb_si = xb_read_input(‘params.txt’).
39. Now change some parameters to convert the model to a Beach Wizard model using the xb_generate_settings tool. This works the same as we did for the storm impact model: pars_BW = xb_generate_settings(‘instat’,’stat_table’,….). Make sure you also specify the interval in which the stationary wave module is called (‘wavint’ = 6000), and change the other relevant parameters.
40. Now merge the new settings into the existing model: xb_BW = xs_join(xb_si , pars_BW).

41. Save your new model using the xb_write_input tool, and run it.
42. Check your results using Quickplot, do you notice different behaviour? If yes, what?

BEACH WIZARD

Coastal managers need up-to-date and accurate information about the state of their coast, including the state of the nearshore bathymetry, the area that is below the tidal level but is changing everyday. Beach Wizard can supply this information using a “smart” translation of video images to bathymetries. Beach Wizard is developed by Deltares in close cooperation with the Naval Research Laboratories, US Geological Survey and Oregon State University, the originator of the ARGUS video system.

Beach Wizard makes use of the unique relation between wave breaking intensity and local depth. It compares (ARGUS) video images of wave breaking intensity that are taken with shore-based camera’s with a computer model output of wave breaking. If there is a difference in location of the wave breaking area, or the intensity of wave breaking, Beach Wizard gradually adjusts the depth in the computer model. It does so for many consecutive (hourly) image, so that the computer and video images of the wave breaking will coincide. If video and model match, then due to the relation between breaking and local depth, the modeled and true bathymetries will also match as has been verified by comparing with field measurements. Using the accurate prediction of the bathymetry, the model can predict local currents, such as rip currents.

In this way the Beach Wizard system can provide hourly information of the beach state, waves and currents with a very high resolution, and for a fraction of the cost of field measurements. It can provide information about the coastal state for coastal managers and about the waves and currents for the purpose of swimmer safety.

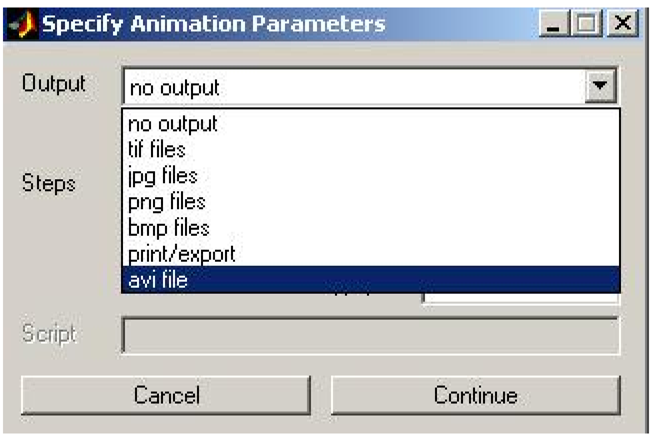
4 TUTORIALS

Quickplot Tutorial

Quickplot can be used to do a first inspection of XBeach output.

1. You can start quickplot via the Delft3D menu, (Utilities → Quickplot)
2. Click to open a file.
3. Choose Files Type 'NetCDF and GRIB Files' and open 'xboutput.nc'. Select 'Hrms wave height based on instantaneous energy' as the data-field to plot. Select the proper location (M corresponds to cross-shore locations and N to longshore locations) or station you want to investigate and press Quick View. If you want to make plots together and compare, you can choose a different colour by Colour, and then click Add to Plot. Use icons to zoom in/out. Drag or click on the toolbar to view different time steps.
4. To make animations of multiple variables instantaneously, variables need to be linked in time (space is another option). You can do this by pressing v and then select:

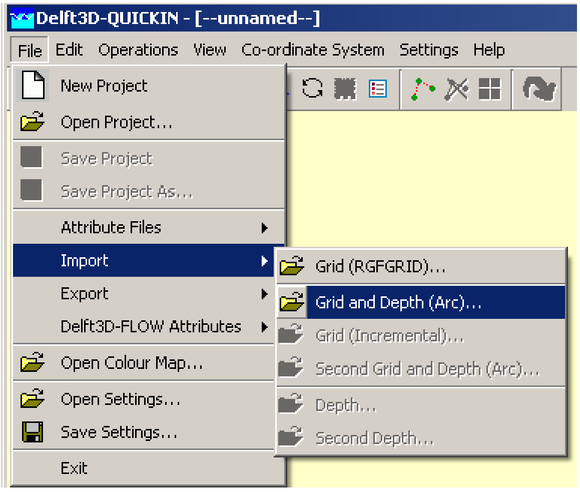
5. In the Quickplot figure click to make animations. Select 'avi' files in the output window to generate your own avi files.



RGFGRID / Quickin Tutorial

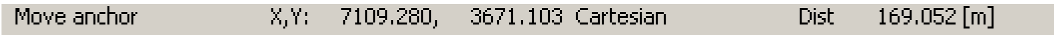
Quickin can be used to inspect computational grids or adapt bathymetries.

1. Importing a computational grid by selecting ‘File → Import → Grid’



2. View the Grid and domain properties by selecting ‘Operations →Actual and Maximum Data Dimensions’.

3. Select  on the toolbar and press A key on the keyboard, an anchor will appear, which acts as zero-distance point. The grid size can be measured in that way. The distance (in meter) is displayed in the status bar at the right of the co-ordinate.



4. To edit model bathymetry select  and draw a polygon. You close the polygon by clicking right mouse button. Next you can edit bathymetry in the polygon created. To do so i.e select ‘Operations → Delete → Depth’, this will delete all depth within the Polygon. Next you can select ‘Operations → Combine Depth and Uniform Value → Fill Missing Depth With Uniform Value. The new bathymetry can be exported to a new bathymetry file via ‘File → Export → Depth’.