WW3 Tutorial 5.1: Unstructured WW3

1. Purpose

In this tutorial exercise, we will construct a hindcast simulation using WW3's unstructured grid mode. Topics covered are the generation of unstructured meshes and the setting up of the unstructured model run, including the specification of offshore wave boundary conditions for coastal applications.

2. Input files

At the start of this tutorial exercise, you will find the following files in the directory `day_5/tutorial_unstructured/`:

`ww3_grid.inp`	(ww3 grid preprocessor input file)
`ww3_bound.inp`	(ww3 boundary condition preprocessor input file)
`ww3_shel.inp`	(ww3 main model input file)
`ww3_ounf.inp`	(ww3 field output postprocessor input file, NetCDF format)
`hawaii_v5.msh`	(unstructured computational mesh)
`mww3.?????????.spec`	(Various boundary condition input files)
`plot_gmsh.m`	(Matlab script for visualizing the unstructured mesh)
`plot_ounf_unstr.m`	(Matlab script for visualizing unstructured output, from NetCDF)

2.1 Unstructured mesh definition

The main difference in this model configuration relative to the regular grid option is the presence of the unstructured mesh file `*.msh`. Since this mesh does not have a regular structure as in the case of a regular grid, the grid points (called *nodes*) and their connections (called *elements*) need to be explicitly defined. The unstructured meshes used by WW3 comprise of *triangular* elements, but in general unstructured grids can also feature quadrilateral, pentagonal, hexagonal, etc. elements.

The construction of a good unstructured mesh is a time-consuming process, and we will only discuss the general principles here. There are a number of grid generation programs available to build meshes, either relatively simple open source systems such as Gmsh[1], Triangle[2], BatTri[3], or more sophisticated proprietary software such as SMS[4]. WW3 has been designed to work with the mesh format produced using the Gmsh program, which is distributed under the terms of the GNU General Public License (GPL). See also the enclosed `gmsh_manual.pdf`. Figure 1 presents an unstructured mesh for the Hawaiian Islands. You can produce this image by running the Matlab function `plot_gmsh.m` (for instructions, type `help plot_gmsh`). This grid accommodates the accurate inclusion of the islands in this otherwise open ocean domain by reducing the grid element sizes in the region of the islands Use Matlab's zoom tool to explore the increasing resolution along the island shorelines.

---

[1] http://geuz.org/gmsh
[2] http://www-2.cs.cmu.edu/~quake/triangle.html
[3] http://www-nml.dartmouth.edu/Publications/internal_reports/NML-03-15
[4] http://www.aquaveo.com/sms

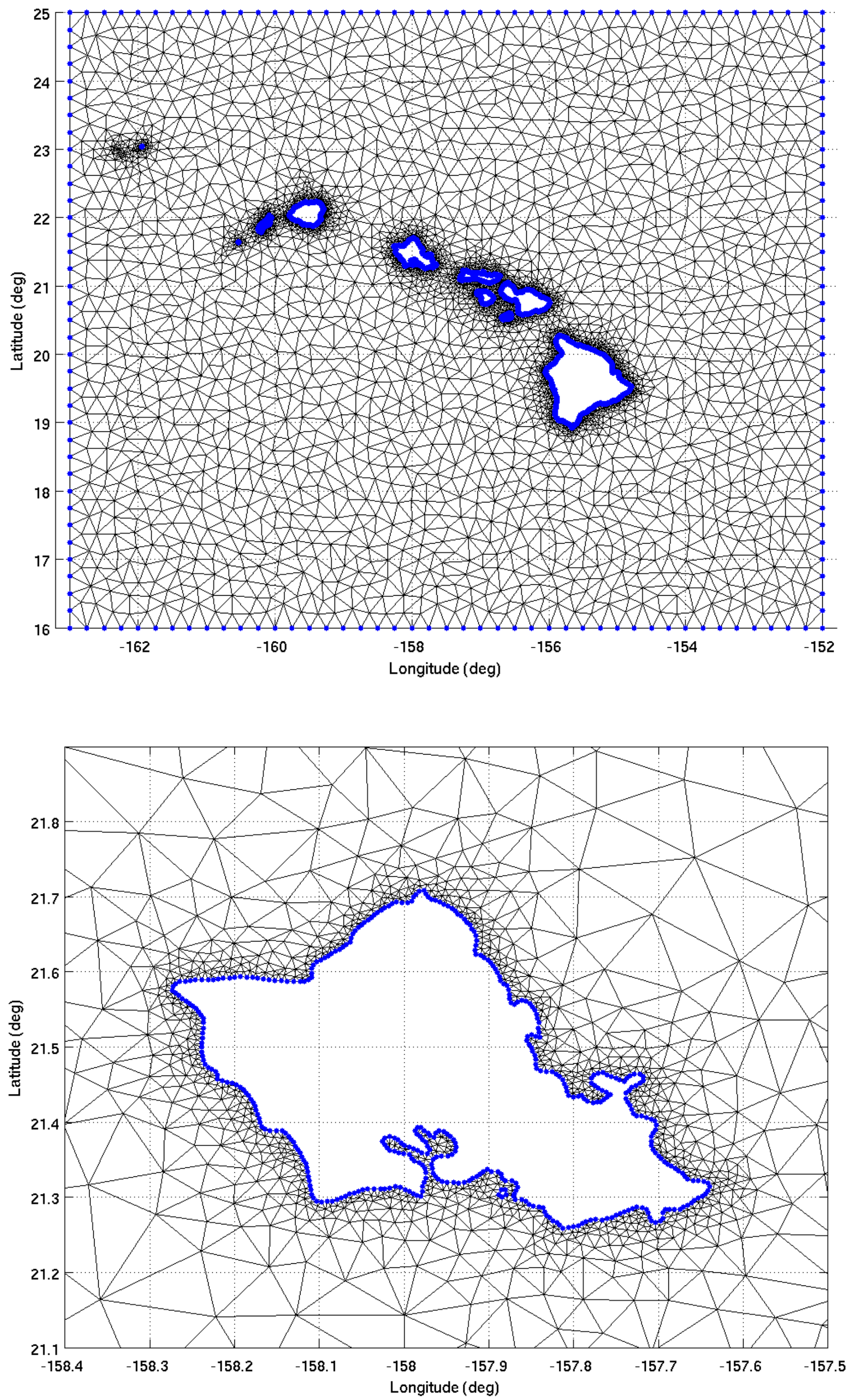

Fig 1: Unstrucutred grid for the Hawaiian Island (top), and a zoomed in view of Oahu (bottom). Blue dots indicate land and ocean boundary nodes.

The first step in the generation of a grid is to specify the land and open ocean boundaries (*polygons*), indicated by the blue nodes in blue in Figure 1. Data for these land boundaries are available at NOAA's National Geophysical Data Center (NGDC) (http://www.ngdc.noaa.gov/mgg/shorelines/shorelines.html). Next, it is specified for which polygon areas the mesh should be built (wet areas) and which polygon areas should be omitted (dry land, e.g. the islands in this case). With this information, the automatic mesh generation algorithm can construct a basic unstructured mesh constrained by the nodal points on these boundary polygons. One typically wants the mesh to resolve relevant spatial gradients in the solution (e.g. wave heights, periods, directions). Offshore elements where the solution changes gradually are thus typically large, whereas shallow water regions near the coast have increasingly smaller elements. The simplest way to create such a mesh - the method followed here - is to specify relatively large increments between nodes along the offshore polygon (here 0.25 arc-deg.) and relatively small increments along the coastal polygons (here ~20 arc-sec).

However, simply constraining mesh elements along the boundary polygons does not always provide adequate resolution in all regions of interest. Therefore, in most grid generation tools, the user can also specify a so-called *shape function*, which is a spatial field that influences the element size allocation in the automatic mesh generation process. Often the bathymetry is used as the basis for the shape function, since shallower areas naturally require higher resolution. However, the user can also construct an arbitrary shape function which focusses resolution on a study region of interest.

Once the unstructured mesh has been built, it is necessary to check the mesh quality. Typically, abrupt changes in element size, small angles and more than 6 connections per node (wagon wheels) should be avoided. Many mesh generators feature facilities to visualize these mesh quality aspects. The final step is to interpolate the bathymetry data onto the nodes of the unstructured mesh.

Now take a minute to study the structure of the mesh file `hawaii_v5.msh` (see Chapter 9 of the Gmsh manual). The file comprises three parts, namely the `Meshformat`, `Nodes` and `Elements` sections, indicated by the tags '`$[`*Section*`]`' and '`End[`*Section*`]`'. The `Meshformat` section contains information on the version number, file type and data size (three numbers, not relevant here):

```
$MeshFormat
2 0 8
$EndMeshFormat
```

The `Nodes` section starts with the total number of nodes (grid points), and then lists the details of each one, namely its index, longitude (x), latitude (y) and elevation:

```
$Nodes
     10366
       1   -163.00000000   16.00000000     5564.00000000
       2   -162.75000000   16.00000000     5577.00000000
       3   -162.50000000   16.00000000     5518.00000000
       4   -162.25000000   16.00000000     5706.00000000
     ...
$EndNodes
```

Finally, the `Elements` section provides information on the connectivity of these nodes, describing the mesh elements and boundary locations. The first number in this section is the total number of elements, followed by a listing of all the elements and their characteristics. These include: element index, element type (`2` = 3-node triangle; `15` = 1-node point, used to describe boundary points), total number of tags describing this element, the list of these tags (e.g. used to specify boundary conditions along a given grid edge), and the list of nodal indices forming the corner points of the element (=3 for triangular elements; =1 for boundary points). The boundary points (type 15 elements) have been indicated with blue dots in Figure 1 above.

```
$Elements
     20765
       1        15    2     0     0           1
       2        15    2     0     0           2
       3        15    2     0     0           3
       4        15    2     0     0           4
       ...
       2599     2     3     0     1     0     6886  6457  3782
       2600     2     3     0     2     0      160     1     2
       2601     2     3     0     3     0     1563  3784  1562
       2602     2     3     0     4     0      160  9211   159
       ...
$EndElements
```

3 Running the model

The following sections detail the setup of the remaining input files, the execution of the unstructured model and the displaying of the output.

3.1 Setting the switch file and compilation

Running WW3 in unstructured mode requires compile-level changes in the source code. Specifically, the switches '`UG`' and '`PR3`' should be included in the `switch` file. Also, enable writing of unstructured mesh output in NetCDF format, include the switch '`NC3`' (for v3) or '`NC4`' (for v4), see Day 1 tutorial. The following `switch` file settings are used in the current exercise (see bold):

```
F90 NOGRB NC3 LRB4 MPI DIST NOPA PR3 FLX2 LN1 ST2 STAB2 NL1
BT1 DB1 MLIM TR0 BS0 XX0 WNX1 WNT1 CRX1 CRT1 O0 O1 O2 O3 O4 O5
O6 O7 O11 O14
```

Now recompile the relevant WW3 subprograms by typing:

```
$ w3_make
```

The updated executables will be placed in the directory `model/exe/`. The model is now ready to process unstructured mesh files. In the next sections, we will step through the input requirements for each subprogram.

3.2 Grid preprocessor

As usual, the first WW3 subprogram to be executed is `ww3_grid`. For the unstructured mode, certain options regarding spatial propagation, the computational mesh as so forth need to be specified in the input file `ww3_grid.inp`. First, in the namelists section, include the following line to indicate unstructured grid mode and the choice of spatial propagation:

```
$ Start of namelist input section -------------------------- $
&UG  UGOBCAUTO = T  EXPFSN = T   /
```

The second keyword above specifies the spatial propagation that will be applied to the unstructured grid. The user can choose between explicit schemes (with CFL constraint, similar to the default WW3) or an implicit scheme, which is unconditionally stable (similar to that used in the model SWAN). The latter option is a practical solution in the case of very small coastal mesh elements, which would otherwise severely limit the CFL time step for spatial propagation. The options are as follows:

`EXPFSN`	N Explicit (Fluctuation Splitting)
`EXPFSPSI`	PSI Explicit (Fluctuation Splitting)
`EXPFSFCT`	Flux Corrected Transport Explicit
`IMPFSNIMP`	N Implicit (Fluctuation Splitting)

Next, the computational grid is defined. In one line specify the computational grid type (keyword '`UNST`' for unstructured, as opposed to '`RECT`' for regular and '`CURV`' for curvilinear), the flag for the grid coordinate convention (`T` = spherical lon/lat; `F` = Cartesian x/y) and the flag for a periodic grid (`T` = periodic; `F` = non-periodic). Subsequently, the grid definition is provided. The required parameters are fewer that in the case of a regular grid (only the last of the usual four lines), since the grid definition is provided in the mesh file `*.msh` itself. Therefore, provide only information on the (i) the limiting bottom depth (m), (ii) minimum water depth (m), (iii) unit number of file with bottom depths, (iv) scale factor for bottom depths (mult.), (v) IDLA, (vi) IDFM, (vii) format for formatted read, (viii) FROM, and (ix) the name of the mesh file.

```
$ Define grid -------------------------------------------- $
'UNST' T F
    4.0 0.30  20  -1. 4 1 '(20f10.2)'  'NAME' 'hawaii_v5.msh'
```

The next step is to specify at which nodes boundary conditions are to be applied (the actual specification of the boundary conditions is discussed in Section 3.3 below). Application of boundary conditions is useful, since an unstructured mesh domain will often be a nest over a limited-area coastal domain, with boundary conditions provided by a larger domain WW3 run, or from buoy observations (if hindcasting). Each line contains discrete grid counters (IX,IY) of the active point and a connect flag. If this flag is true, and the present and previous point are on a grid line or diagonal, all intermediate points are also defined as boundary points. In unstructured mode, only the single node index of the boundary nodes should be provided, and is specified with the grid counter IX (the IY counter is set to 1 and ignored). The boundary node indices can be found in the `*.msh` file discussed in Section 2.1. above, and typically corresponds with the open ocean polygon specified a the start of the mesh generation process.

```
$ Input boundary points ------------------------------------- $
   1     1   F
   160   1   T
```

This completes the adjustments to the input file. Now run the grid preprocessor by typing:

```
$ ww3_grid > ww3_grid.out
```

The usual binary `mod_def.ww3` model definition file will be created as output.

3.3 Boundary condition preprocessor

Since unstructured model runs will typically focus on coastal regions, offshore wave boundary need to be applied. Boundary conditions can either be generated by a preceding, larger domain WW3 run, which is packed into a nesting file `nest.ww3`, or it can be compiled from arbitrary wave spectra (either model or observations), also packed in the `nest.ww3` file format. Here we will discuss the procedure for producing the latter form of boundary condition input using the preprocessor `ww3_bound`. The user instructions are given in the file `ww3_bound.inp`, see example below. In the first line, the option to read from ('`READ`') or write to ('`WRITE`') the boundary data file `nest.ww3` is given. Here we will use the latter option. The boundary condition data may be spatially varying (as well as temporally), but will typically not be specified explicitly at every boundary point. In the next line, we can therefore chose between assigning either the value of the nearest specified boundary value (option `1`), or to interpolate between specified values (option `2`). In the next line, the file format is given (`A` = ASCII or `N` = NetCDF).

```
$ boundary option: READ or WRITE
WRITE
$ Interpolation method. 1: nearest, 2: linear interpolation
2
$ Type of spectra files: 'A' for ASCII and 'N' for NetCDF
$ (this option is not yet implemented)
A
$ list of spectra files. These ASCII files use the WAVEWATCH III
$ format examples of such files can be found at (for example):
$   ftp://polar.ncep.noaa.gov/pub/waves/develop/glw.latest_run/
$   (the *.spec.gz files)
$   http://tinyurl.com/iowagaftp/HINDCAST/GLOBAL/2009_ECMWF/SPEC
$
$ In the case of NetCDF files ... that is coming ...
$
mww3.W1520N160.spec
mww3.W1520N170.spec
...
mww3.W1630N250.spec
'STOPSTRING
```

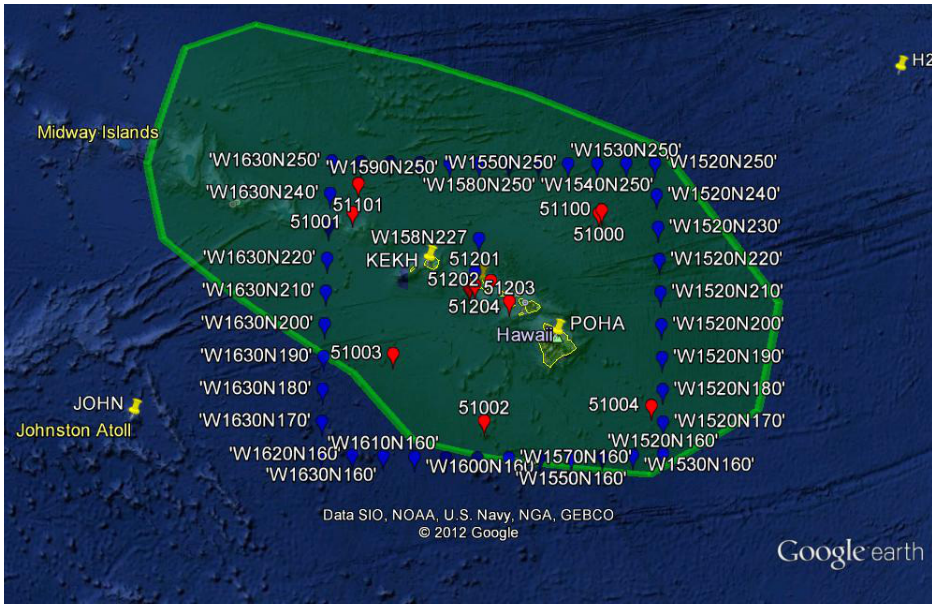

Fig 2: Boundary condition locations

Next, the file names of the two-dimensional spectral files containing the actual boundary data are listed. Any number of files may be specified, and then terminated with the keyword '`STOPSTRING`'. The format of these boundary condition files is the same as any two-dimensional WW3 spectral file (example below). Note in particular that the geographic location as well as the date at which each boundary condition is valid is listed inside each of these files (bold values). Here we apply boundary data shown in Figure 2. For the boundary condition to be valid, this geographical location should coincide with one of the boundary locations in the `*.msh` file (Section 2.1) and listed in the `ww3_grid.inp` file (Section 3.2). The boundary conditions should at least be specified at the start and end points of each boundary segment or side (present example), but intermediate points can be given also. All active boundary points in between are assigned values based on the interpolation option `1` or `2` chosen above.

```
'WAVEWATCH III SPECTRA'     32    24     1 'MULTIGRID GLOBAL05+OTHERS     '
 0.373E-01 0.410E-01 0.451E-01 0.496E-01 0.546E-01 0.601E-01 0.661E-01 0.727E-01
 0.800E-01 0.880E-01 0.967E-01 0.106E+00 0.117E+00 0.129E+00 0.142E+00 0.156E+00
 0.171E+00 0.189E+00 0.207E+00 0.228E+00 0.251E+00 0.276E+00 0.304E+00 0.334E+00
 0.367E+00 0.404E+00 0.445E+00 0.489E+00 0.538E+00 0.592E+00 0.651E+00 0.716E+00
  0.157E+01  0.131E+01  0.105E+01  0.785E+00  0.524E+00  0.262E+00  0.000E+00
  0.602E+01  0.576E+01  0.550E+01  0.524E+01  0.497E+01  0.471E+01  0.445E+01
  0.419E+01  0.393E+01  0.367E+01  0.340E+01  0.314E+01  0.288E+01  0.262E+01
  0.236E+01  0.209E+01  0.183E+01
20080225 000000
'W1520N160 '  16.00-152.00     5458.7    5.23 109.8    0.00 270.0
  0.846E-07  0.371E-06  0.141E-05  0.743E-05  0.326E-03  0.590E-02  0.266E-02
  0.285E-03  0.188E-03  0.328E-04  0.131E-04  0.203E-05  0.345E-06  0.180E-06
  0.566E-03  0.305E-03  0.932E-04  0.224E-04  0.360E-05  0.342E-06  0.192E-07
  ...
```

The boundary condition preprocessor is run by typing

```
$ ww3_bound > ww3_bound.out
```

which produces the boundary condition file `nest.ww3`.

3.4 Field preprocessor

In this example we will consider a swell event with pure propagation, and not impose any wind field. Hence the program `ww3_prep` will not be used.

3.5 Main model

There are no specific changes to the main model subprogram ww3_shel to enable the unstructured mode. Take a look at the contents of the input file `ww3_shel.inp`. Notice the inclusion of the field output request for a number of wave parameters:

```
$ Type 1 : Fields of mean wave parameters
$          Standard line and line with flags to activate output fields
$          as defined in section 2.4 of the manual. The second line is
$          not supplied if no output is requested.
$
   20080225 000000 10800  20080227 000000
  F F T F F  F T F F F  F F F F F  F F F F F  F F F F F  F F F F F
  F F F F F  F F T T F  F F T T F  T T F F F  T F F
```

To run the main model subprogram, type:

```
$ mpirun -np 4 ww3_shel > ww3_shel.out
```

Here we use MPI, in order to speed up the rather lengthy simulation (see also the `switch` file above). As usual, the binary output files `out_grd.ww3` and `out_pnt.ww3` will be created.

3.6 NetCDF field output postprocessor

It is practical to pack the output of an unstructured mesh run into a flexible NetCDF format file. This is done using the NetCDF field postprocessor subprogram `ww3_ounf`. In the input file `ww3_ounf.inp`, set the output fields required (same as in `ww3_shel.inp` above).

```
$ Output request flags identifying fields as in ww3_shel input and
$ section 2.4 of the manual.
$
$----------------------------------------------------------------
$ D C W D W  I H W T T  T C F D S  D P P P P  P P W P D  F C C U C
$ P U N T L  C S L 0    M G P I P  P H T L T  S W S N T  C F F S H
$ T R D   V  E     2    1 E   R R    S P P H  I S F R D    X D T A
$----------------------------------------------------------------
  F F T F F  F T F F F  F F F F F  F F F F F  F F F F F  F F F F F
  F F F F F  F F T T F  F F T T F  T T F F F  T F F
```

Next, specify the version of NetCDF installed on your system (v3 or v4, see above). In the following lines, list which wave partitions to compute from the output field (0=wind sea, 1=first swell, 2=second swell, etc.), and whether these should be written to the same output file (option `T`) or not (`F`).

```
$ Output type 4 [3,4]               (netCDF version)
$ Output type 0 1 2 [0,1,2,3,4,5]   (swell partition)
$ variables [T] or not [F] in the same file
  3
  0 1 2
  F
```

Next, details are provided of the output file prefix, the format for the output date and the range over which the output is written. For unstructured meshes, nodes are identified by a single index (see file structure above). This is specified in the IX range, with the values of the IY range set to `1 1` and ignored by the program. The final number is the format indicator IDFM (here 3 = free format).

```
$ File prefix
$ number of characters in date
$ IX, IY range
ww3.
6
 1 10366 1 1 3
```

Now run the postprocessor by typing:

```
$ ww3_ounf > ww3_ounf.out
```

This produces the output file `ww3.200802_PP.nc` for each output parameter PP, for example ww3.200802_hs.nc. This file can be visualized with the Matlab function `plot_ounf_unstr.m` (for instructions, type `help plot_ounf_unstr`), producing the plot shown in Figure 3. Zoom in to study the detail along the coastlines of the islands.

4. Conclusion

In this tutorial exercise, we have set up and run WW3 in unstructured mode. This mode of operation is the method of choice to resolve the vast range of spatial scales found in nearshore regions.

More information:

Fabrice Ardhuin (Fabrice.Ardhuin@ifremer.fr)
André van der Westhuysen (Andre.VanderWesthuysen@noaa.gov)

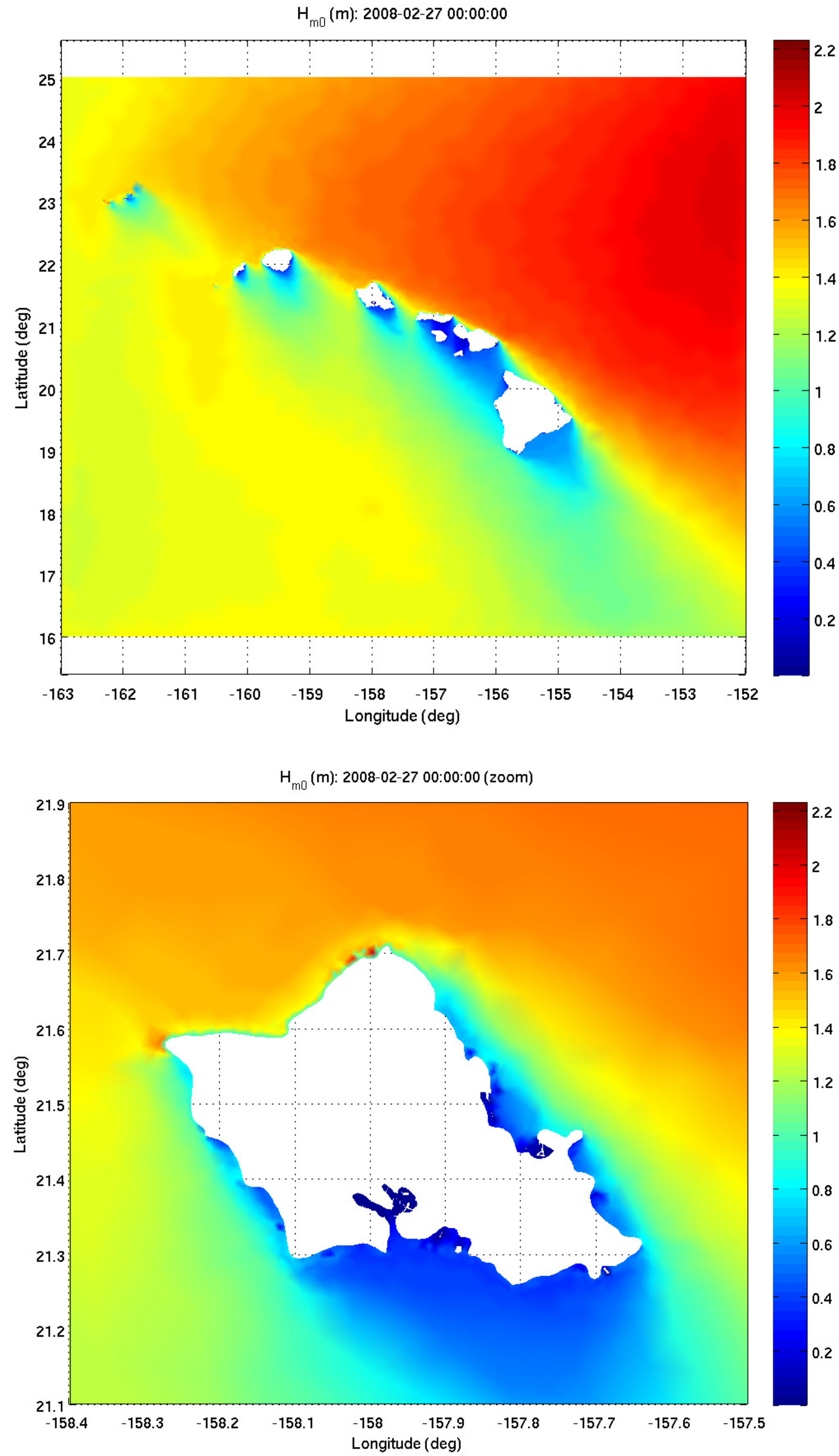

Fig 3: Wave height field produced on the Hawaiian Island unstructured grid (top), with detail over Oahu (bottom).