WW3 Tutorial 1.1: Basic Setup and Compilation

Purpose

In this tutorial exercise, we will go trough the steps of installing and compiling version 4.08 of the WAVEWATCH III® code on your machine. Subsequently, the compiled code will be tested by means of a simple regression test case to ensure that the compilation has been successful. Variations of this compiled code will be used throughout the rest of the tutorial series.

System requirements

Please note that the WAVEWATCH III system has been developed for a UNIX/Linux environment. Basic proficiency in this type of operating system is therefore a prerequisite for this course. Also, ensure that the following programs have been installed on your system:

- Fortran 77 compiler
- Fortran 90 compiler
- NetCDF installation (version 3 or version >4.1.1)
- GrADS
- MATLAB >7.10.0

Exercise 1: Installation

In this exercise we shall cover how to install and run the WAVEWATCH III (WW3) code from the release version on a Linux workstation. This is the standard approach used for setting up WW3. Alternatively, if you have access to the development Subversion (SVN) server, you can set up the code from there. That will be covered in a later exercise session.

1.1 Downloading the distribution code

WW3 code can be obtained from two locations: (1) the password-protected public WW3 distribution site on NOAA/NCEP's website (http://polar.ncep.noaa.gov/waves/wavewatch/wavewatch.shtml), and (2) the WW3 development SVN server (this will be covered later in the week). You will need a username and password to get access to the former – see the WAVEWATCH III website for more information. For the purpose of this workshop, a beta version of WW3 version 4.8 has been provided on all the computers, as it would be made available in a public release.

(***Note:*** This is not the official public release. However, since significant changes have been introduced between the current development version of WW3 and the official public release (v3.14), it was deemed worthwhile to provide a beta version for the purposes of this workshop). Those who have never downloaded the public version before, and have not registered with us, should do so before the end of the workshop so that you can have a username and password and try to download and run the public release version.

1.2 Installation

This part of the exercise will follow the installation section of the manual (Section 5). Please refer to the latter for additional information. In your home directory you will find a `wwatch3` directory where the tarred code is kept. The code is distributed in this fashion for public release and we shall start from this point. In this directory you will find the following files

```
install_ww3_v4_svntar
wwatch3.4.08.model.tar
wwatch3.4.08.guide.tar
wwatch3.4.08.manual.tar
wwatch3.4.08.regtests.tar
```

(***Note:*** We are in the process of over hauling our packages so between now and the next public release of the code filenames and the number of files may change, but the general idea of using one installation script with a set of tarballs will be unchanged)

Step 1: The process starts by running this installation script:

```
$ install_ww3_v4_svntar
```

Installation will begin with asking if you want to use a local or global option. The choice is used to decide where to place an environment defining file (more on this below). Earlier versions would only place the environment file in your home directory (global option) but we are in the process of changing this to allow for a local environmental variable. However, at this moment not all our scripts have been migrated so please chose the global option. The script will then prompt you if the default environment is acceptable. This includes directory where temporary files will be stored, the main home directory for the code and the generic F77 compiler that will be used for compiling the pre-compiler auxiliary program **`w3adc`** (source code in `ww3/aux/w3adc.f`, see below). This Fortran 77 program is needed to create Fortran specific files from the WAVEWATCH III `*.ftn` file formats. You will have to change this because the compiler that is available is `gfortran` (***Note:*** This is important, because if the script cannot find this compiler it will not be able to generate the **`w3adc`** executable. Many an hour was lost in debugging just because an appropriate F77 compiler was not specified in the installation process!)

Step 2: After running the installation script, the following directories will have been created in `wwatch3/`:

```
aux/        bin/        inp/        manual/
ftn/        mod/        obj/        guide/
exe/        tmp/        work/       oldtests/
regtests/
```

See Section 5 of the manual for the purpose of each of these directories. To ensure that the installation occurred properly, check that the pre-compiler executable **`w3adc`** is present in the `bin/` directory.

Step 3: As part of the installation script, a **`.wwatch3.env`** file is generated in the user's home directory. Take a moment to check the content of this file. It should resemble the example below:

```
#
# Environment variables for wavewatch III
# ---------------------------------------
#

WWATCH3_LPR     printer
WWATCH3_F77     pgf77
WWATCH3_CC      cc
WWATCH3_DIR     $HOME/wwatch3/
WWATCH3_TMP     $HOME/wwatch3/ww3_tmp/
WWATCH3_SOURCE  yes
WWATCH3_LIST    yes
```

Note: This environment file is used in all subsequent compilation scripts. Users have to be extra careful when having multiple versions of WW3 as the installation script will use the existing `.wwatch3.env` file and that can destroy an existing installed WW3 code! We are currently working on a new installation process that will create the environment file in the directory where the code is stored, as opposed to the home directory.

Step 4: The scripts, executables, compiler and switch settings needed to compile WW3 codes are stored in the `ww3/bin/` directory, whereas the executable WW3 codes are stored in `ww3/exe/` directory. If you want to be able to compile and execute WW3 codes from anywhere, you need to add these two directories in their path environment, e.g. in the **`.cshrc`** profile file. A default path name has been set up in everyone's `.cshrc` file in your home directory. Take this moment to edit the file to provide the appropriate path to your directory (each user's path will be unique) ***Tip:*** We usually create generic paths `$HOME/wwatch3/bin` and `$HOME/wwatch3/exe`, and link `wwatch3` to the actual directory where the code is stored. Remember to run

```
source .cshrc
```

after editing the file so that the paths are appropriately setup. To check that these paths have been set up properly, run

```
which w3adc,
```

which should display the path to this file.

Step 5: To enable reading and writing NetCDF files by WW3, information on the NetCDF installation needs to be provided. Versions 3 and 4 are currently supported. Information on the NetCDF version, the library path and include path must be specified in the **`.cshrc`** profile file in the user's home directory. This has already been done but do take a moment to check them out

Exercise 2: Compilation

Unlike the auxiliary programs used in the compilation process (e.g. **w3adc** discussed above), the actual WAVEWATCH III code is written in Fortran 90. Options for various Fortran 90 compilers are included in the WAVEWATCH III distribution package, including Intel, Portland and gfortran. For each compiler choice, a unique `comp` and `link` script should be used. These are found in the directory `ww3/bin/`, and are labeled **comp.Intel**, **comp.Portland**, etc. and **link.Intel**, **link.Portland**, etc. respectively.

Step 1: Choose the relevant compiler and copy (or link) the corresponding **comp.XXX** and **link.XXX** scripts to simply **comp** and **link**. Of particular importance here are the lines where the compiler ("`comp=`") and the compiler options ("`opt=`") are specified. Check these for completeness. We have created specific scripts for this workshop called `comp.UMD` and `link.UMD` (***Note:*** If the compiler you have available does not have corresponding **comp** and **link** scripts included in the distribution package, then you will have to create your own, based on the compile options of your particular compiler. You can use any one of the available **comp** and **link** scripts as templates. Once you are successful, we ask that you share your scripts so that they can be distributed with subsequent releases.)

Step 2: The next step is to set up a **switch** file (default file available in `ww3/bin/`) which determines the model options (physics packages, numerical schemes, serial/parallel architecture etc.). See the manual section on various model switches (Section 5.4). Specify the following list of model options as one line at the top of the **switch** file and save it (for NetCDF v4, replace "`NC3`" with "`NC4`"):

```
F90 NOGRB NC3 LRB4 SHRD 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
```

Step 3: The compilation of WW3 is done using the script **w3_make**. There are two ways to do this: You could run the command from the directory `ww3/bin/`, in which case you do not have to make any changes. Alternatively, you could run it from your current work directory. This is useful, since you may need to compile various versions of WW3 to run different model configurations within the same project. In this case, you have to make sure that the PATH environment variable includes the directory `ww3/bin/` (see above). You also need to create links from your work directory to the files **comp, link** and **switch** in the `ww3/bin/` directory. Here we will demonstrate the latter option. Type the following sequence of commands:

```
cd ~/day_1/tutorial_compilation
ln -sf ~/wwatch3/bin/comp .
ln -sf ~/wwatch3/bin/link link
ln -sf ~/wwatch3/bin/switch .
```

Now run the compilation script in the directory `day_1/tutorial_compilation`:

```
$ w3_make
```

This script will output a screen listing of WW3 program components as they are being compiled. The process ends with an "*** `end of compilation` ***" statement.

Step 4: If the compilation completed successfully, then a number of executable programs will have been placed in the directory `wwatch3/exe/`. These are the collection of subprograms comprising WW3:

`ww3_grid`	(Model grid preprocessor)
`ww3_strt`	(Initial conditions preprocessor)
`ww3_prep`	(Input field preparation – wind, current, water level)
`ww3_shel`	(Main WW3 program, for single grid, or one-way nesting operation)
`ww3_multi`	(Main WW3 program, for two-way nesting operation)
`ww3_outf`	(Field output postprocessor)
`ww3_outp`	(Point output postprocessor)
`ww3_trck`	(Wave output along a predefined track)
`ww3_grib`	(Field output in WMO's GRIB2 format)
`ww3_gint`	(Re-gridding output to older formats, for backward compatibility)
`gx_outf`	(Field output postprocessor, in GrADS format)
`gx_outp`	(Point output postprocessor, in GrADS format)
`ww3_ounf`	(Field output postprocessor, in NetCDF format)
`ww3_ounp`	(Point output postprocessor, in NetCDF format)

If these files are not all present, or if error messages appear on the screen, compilation has not been successful. The most common error encountered is that the **`comp`** and **`link`** scripts have been set up incorrectly. Either a non-existent compiler has been specified (line "`comp=`"), or incorrect compile options were provided (line "`opt=`"). Check these, and rerun **`w3_make`**.

Exercise 3: Testing the compilation.

Once the installation and compilation have been completed successfully, it is good practice to test the installation by running one or more regression tests. For this purpose, we will run a regression test in the directory `day_1/tutorial_compilation/regtests_tutorial/ww3_tp2.2/`. The regression tests each contain a documentation file `info`, and a directory containing the model inputs, called `input/` (see lectures Day 3). The present case is a simple test of wave propagation along the equator. Inside the `inputs/` directory, the following files are found:

```
ww3_grid.inp
ww3_strt.inp
ww3_shel.inp
ww3_outf.inp
```

These are the input files for a selection of the subprograms listed in Step 4 above. A typical execution of a WW3 simulation comprises of sequentially running a number of these model components. The number of subprograms run will depend on the complexity of the input and output requirements, but the subprograms `ww3_grid`, `ww3_strt`, `ww3_shel` and `ww3_outf` can be considered a minimum. To test whether the paths to these programs have been set properly, type e.g.

```
which ww3_grid.
```

Step 1: The simulation is started by running the grid preprocessing program `ww3_grid`, which uses the corresponding input file `ww3_grid.inp`. Take a moment to go through this input file. All lines starting with the "$" sign are comment lines and are ignored by the WW3 executables. To run the grid preprocessor type:

```
ww3_grid > ww3_grid.out
```

Here the screen output has been redirected to the log file `ww3_grid.out`. The main product of the grid preprocessor is the binary file `mod_def.ww3` which contains a description of the model domain and discretizations used by all the other subprograms.

Step 2: Next, the initial conditions preprocessor `ww3_strt` is run, which sets the state of the wave field at the first time step. It uses the input file `ww3_strt.inp`. In the present test case, the simulation is started with a swell field centered on lat/lon = (0 N, 90 E). To run the initial conditions preprocessor, type:

```
ww3_strt > ww3_strt.out
```

Again the screen output is redirected to the log file ww3_strt.out. The main output of this program is the binary initial conditions file `restart.ww3`, used by the main model subprogram `ww3_shel`.

Step 3: Now we are ready to run the main model subprogram `ww3_shel`. Its associated input file `ww3_shel.inp` contains information on the input fields to be used, the start and end times of the simulation, and output types. Take a minute to study the contents of this file. To run the main model subprogram, type:

```
ww3_shel > ww3_shel.out
```

The screen output is again redirected to the file `ww3_shel.out`. Depending on the outputs types requested, this subprogram produces the binary bulk output files `out_pnt.ww3` and `out_grd.ww3` used by the post-processing subprograms. In addition, a log file `log.ww3` is produced, which lists information on input and output sources used at each time step.

Step 4: In this final step, the field post-processing program `ww3_outf` is run to produce spatial fields of wave parameters, extracted from the binary file `out_grd.ww3` discussed above. The input file `ww3_outf.inp` contains instructions for the data extraction. To run the field post-processor, type:

```
ww3_outf > ww3_outf.out
```

This produces a series of daily field output files for the significant wave height, namely `ww3.68061200.hs`, `ww3.68061300.hs`, … `ww3.68061800.hs`. These files can be plotted using the Matlab script `plot_fields.m`. Before proceeding, set the directory variable in this script to your own local directory:

```
direc = '[your run directory]';
```

Running this script produces the output shown in Figure 1.

If all the products described above have been produced successfully, the basic components of the installation and compilation can be considered validated. The same procedure can be followed for the rest of the regression tests in the folder `wwatch3/regtests/`, in order to ensure that all the features of the model are performing as intended.

Conclusion

In this tutorial exercise, we have installed, compiled and tested the WW3 code. The WW3 subprogram executables created during this tutorial now reside in the directory `wwatch3/exe/`, variations of which will be used in the rest of the tutorial series. Some of the tutorials that follow will require different model settings, for which the `switch` file will have to be altered, and the compilation process repeated.

More information:
Arun Chawla (Arun.Chawla@noaa.gov)
André van der Westhuysen (Andre.VanderWesthuysen@noaa.gov)

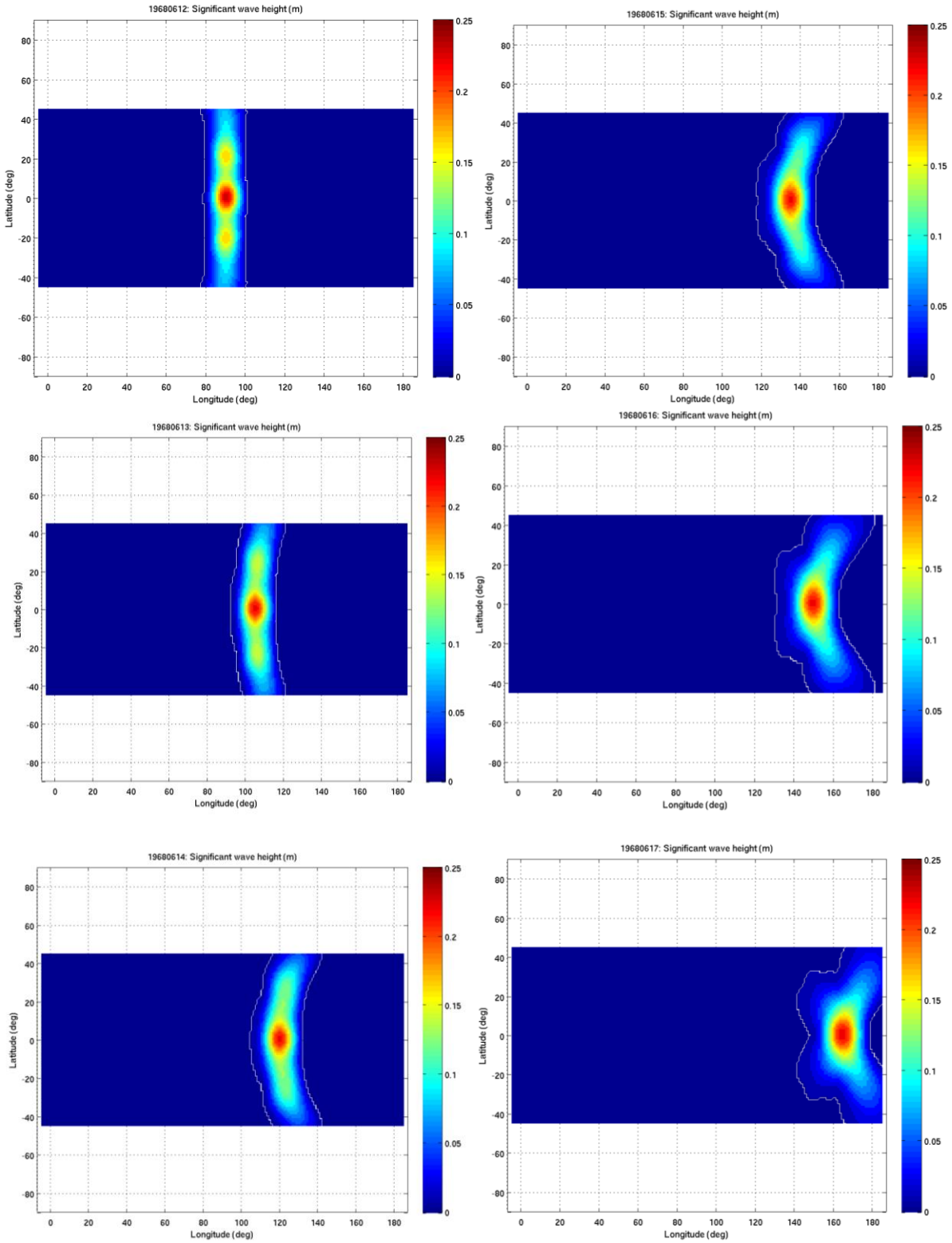

Fig 1: Field output from regression test tp2.2, showing wave propagation along the equator