WAVEWATCHIII Online Tutorial

Release 6.07

Franklin Ayala

Jun 03, 2022

CONTENTS:

WAVEWATCH III (WWIII) is a community wave modeling framework that includes the latest scientific advancements in the field of wind-wave modeling and dynamics. The core of the framework consists of the WAVEWATCH III third-generation wave model (WAVE-height, WATer depth and Current Hindcasting), developed at NOAA/NCEP.

WWIII solves the random phase spectral action density balance equation for wavenumber-direction spectra. The implicit assumption of this equation is that properties of medium (water depth and current) as well as the wave field itself vary on time and space scales that are much larger than the variation scales of a single wave. The model includes options for shallow-water (surf zone) applications, as well as wetting and drying of grid points. The wave model is a sophisticated modeling system with numerous developments that have been added in recent years (wave – hurricane interaction physics, new wave growth and dissipation physics packages, wave – mud, wave – vegetation and wave – bottom interaction physics to name a few) that make this model attractive for this project. Propagation of a wave spectrum can be solved using regular (rectilinear or curvilinear) and unstructured (triangular) grids.

WWIII is also used to model atmosphere-ocean interactions with the OWA (Ocean-Wave-Atmosphere) modelling system. Due to a diferent capabilites of coupling with atmosphere models (i.e WRF) and ocean models (i.e CROCO, NEMO,etc), this model is widely used by the community climate modelling.

This tutorial were created with the WWIII v6.07 and is designed to take you through this amazing wave model, step by step. Simply follow the tutorial flow (read de table of contents)

Please, look out all signs that showed along the tutorial:

Note: This is note text. Use a note for information you want the user to pay particular attention to.

Hint: hint/tip/important are similar

We recommend that you work through the tutorial examples before you try to run WRF ARW on your own.

Warning: Warning/caution/attention are similar

This is warning text. Use a warning for information the user must understand to avoid negative consequences.

Danger: danger/error are similar

This is a new tip, wonderful!

Let's go to model and ride the wave with us!

CHAPTER

ONE

SYSTEM REQUERIMENTS

- Fortran compiler on linux
- C++ compiler on Linux
- netCDF library
- MPICH library

1.1 Fortran and C++ compilers

First and foremost, it is very important to have a Fortran and C++ compiler (gcc, cpp, etc.). We recommend gfortran and gcc. To test whether these exist on the system, type the following:

```
which gfortran
which cpp
which gcc
```

If you have these installed, you should be given a path for the location of each. If you dont have you can install them from your terminal:

```
sudo apt install build-essential
sudo apt-get install gfortran
```

or

```
sudo apt-get install gcc gfortran
```

These commands should install the latest versions of gfortran and gcc available for your system. To determine the versions of your libraries you have, type:

```
gcc --version
gfortran --version
cpp --version
```

1.2 Installing netCDF

It's possible to install netCDF directly from snap on the ubuntu terminal with `sudo apt-get install libnetcdf-dev libnetcdff-dev` (this would download the lastest version netCDF), however this opction could bring problems at the moment to coupling WW3 with WRF troughout OASIS. The main reason is the 3 models must be compiled with the same netCDF version, and the lastest version netCDF seems to be conflicting with older WRF versions (v3.7.1) used in this documentation.

It is recommended to install netCDF-Fortran and netCDF-C manually, to do this first it woul be installed 2 dependeces: zlib and hdf5. These libraries together with libpng and jasper libraries will be used for the WRF compiling.

1.2.1 Installing dependences

Before getting started, you must move into to the folder that contain WRF model directory and to type

```
mkdir -p Build_WRF/Libraries
```

The command `-p` will create parent directory first if it doesn't exist, i.e `Build_WRF` and then will create the folder `Libraries` where will be installed the libraries.

Add the following enviroment variables to your bash profile. In our case, the bash profile is `.zshrc`, this file is normally in `home` folder and you can easily edit it.

```
export DIR=/home/franklin/Documentos/WRF/Build_WRF/Libraries
export CC=gcc
export CXX=g++
export FC=gfortran
export FCFLAGS=-m64
export F77=gfortran
export FFLAGS=-m64
```

and type the following command to upload the change sin bash profile:

```
sourde ~/.zshrc
```

The firt library to be installed is zlib. zlib is a library used to SOMETHING. Go to the zlib oficial page and download the suorce files to the installation (file .tar) in `Libraries` directory . In our case, we choose the version 1.2.8. In your terminal type the following:

```
cd $DIR
tar xvf zlib-1.2.8.tar.gz
./configure --prefix=$DIR/grib2; make clean; make all install
cd ..
```

The folder in `$DIR/grib2` is the folder where will be installed and we explain it.

The other library, hdf5 is used in SOMETHING. HDF5 is available from the HDF5 downloads site . Our version installed is 1.8.13 and the installing process is similar to the zlib

```
tar xvf hdf5-1.8.13.tar.gz
cd hdf5-1.8.13
./configure --prefix=$DIR/grib2 --with-zlib=$DIR/grib2; make clean; make all install
cd ..
```

the command –with-zlib link the hdf5 with zlib library.

If you go to the grib2 directory, and you type `ls` in your terminal, you could get something like this:

```
libhdf5.a     libhdf5_hl.la  libhdf5_hl.so.8      libhdf5.la        libhdf5.so    ␣
↪libhdf5.so.8.0.2  libz.so    libz.so.1.2.8
libhdf5_hl.a  libhdf5_hl.so  libhdf5_hl.so.8.0.2  libhdf5.settings  libhdf5.so.8  libz.a␣
↪           libz.so.1  pkgconfig
```

1.2.2 Building netCDF

netCDF library is used widely for the meteorological and oceanic data. In versions before 4.2, the Fortran netCDF library source (netCDF-Fortran) was bundled with the C library source (netCDF-C) in one distribution. With more recent versions, the Fortran netCDF library has been split off into an independent source distribution, intended to be built as a separate library, after the C library is built and installed.

We decide to install 4.1.3 version, but you can install a newer version if you want it.

Hint: We recommend a library where NetCDF-4/HDF-5 is enabled.

To install the version above mentioned, visit the netCDF downloads site and get your .tar file. Then, type:

```
tar xvf netcdf-4.1.3.tar.gz
cd netcdf-4.1.3
LDFLAGS=-L$DIR/grib2/lib CPPFLAGS=-I$DIR/grib2/include ./configure --prefix=$DIR/netcdf;␣
↪make clean; make all install
cd ..
```

You can test if your netCDF library were well built typing these command lines in your terminal:

```
wget https://www.unidata.ucar.edu/software/netcdf/examples/programs/simple_xy_wr.f90
gfortran simple_xy_wr.f90 -o test_nc.exe -I$NETCDF/include -L$NETCDF/lib -lnetcdff
./test_nc.exe
```

This should print:

```
0          12          24          36
SUCCESS writing example file simple_xy.nc!
```

There is a lot of available tools to check, pre-visualizate and manipulate data content in .nc files. `ncdump` is one of them and can be installed with the commands:

```
sudo apt install netcdf-bin
```

if you type:

```
ncdump -h simple_xy.nc
```

This print information about the .nc file:

```
netcdf simple_xy {
dimensions:
x = 6 ;
```

(continues on next page)

(continued from previous page)

```
y = 12 ;
variables:
int data(x, y) ;
}
```

Finally, its important to verify that the installed version has netCDF4, to do it, move into `bin` directory in `$NETCDF` and type:

```
./nc-config --has-nc4
```

If the terminal print: **yes**, then all is working perfectly!

1.3 Installing MPICH

MPICH library is necessary if you are planning to build WW3 in parallel and to couple WW3 with WRF. If your machine does not have more than 1 processor, or if you have no need to run WW3 with multiple processors, you can skip this section.

To install this library:

```
sudo apt install mpich
```

if was correctly installed, the you can type:

```
which mpirun
```

you should be given a path for the location of each.

./ungrib.exe: error while loading shared libraries: libpng12.so.0: cannot open shared object file: No such file or directory

When the coupling

```
cd /your_software/../lib/ (the directory containing libz.so.1)
mv libz.so.1 libz.so.1.old
ln -s /lib/x86_64-linux-gnu/libz.so.1
```

The solution were found in this stackoverflow questions

CHAPTER

TWO

GETTING THE CODE

2.1 WAVEWATCH

WAVEWATCHIII v6.07 is hosted in Github. Github is a Git repository hosting service and Git is a free, open and well-known source distributed version control system. Because it is a open project, the modelling community can both clone the newest or any version WW3 and contribuite to the development of WW3 from the creation of new features to the bugs and errors handling. For more information you can visit (link)

It is recommended to review the availiable information of the WW3 repository. In a specific section you can view all releases WW3 and a brief description of each one.

To clone the WW3 repository implies to do a copy of project from Github. To clone type the following command in your terminal from the directory where you want download WW3 source code. It must be installed git (on linux/ubuntu is installed by default)

```
git clone https://github.com/NOAA-EMC/WW3
```

The previous command line will download a folder called `WW3` in the directory where it was cloned. This folder contains this structure:

- `manual`: the manual directory, here are the .tex files that build the manual WW3.
- `model`: model source code, here the files are built after the compilation (FORTRAN modules, etc.), template input files to execute the model, etc.
  - `bin`: this folder contains the neccesary executable programs to compile WW3.
  - `aux`: Raw auxiliary programs fo the compilation WW3.
  - `ftn`: All FORTRAN files (source code).
  - `inp`: input files.
  - `nml`: namelist input files.
  - `mod`: module files (created only after the compilation).
  - `obj`: object files (created only after the compilation).
  - `exe`: executable files (created only after the compilation) .
- `regtests`: WW3 test cases.
- `guide`: a guide to WW3 in .tex files.
- `smc_docs`: documentacion and files to SMC grids.

Additionally, from your top-level- directory (`path/to/WW3/model/`) you can execute `ww3_from_ftp.sh` located in `bin` folder:

```
./model/bin/ww3_from_ftp.sh
```

Hint: To execute a file on Linux/Ubuntu you have the following options: - type `.` and then the `path/to/executable/file/` - type `sh path/to/executable/file/` - type `bash path/to/executable/file/`

With this instructions you can get the binary and large files used for WW3 test cases in `regtest` directory. With this ub the top-level directory you could get two folder more:

- `cases`:
- `data_regtests`: data necessary for the execution some regtests

Also, in this directory there is a hidden folder `.git`, which is created by default when a Github repository is cloned on your local machine, a licence file, .gitignore file (to exclude same files) and README.md file to desribe instructions.

2.2 Gridgen

Gridgen is a MATLAB program to create files that contain bathimethric information to be used in WW3. This program is also hosted in Github, you can visit the repository (LINK) and obtain it in your local computer with (TALK MORE ABOUT GRIDGEN):

```
git clone https://github.com/NOAA-EMC/gridgen
```

This command line will download a folder called `gridgen` in the directory where it was cloned. This folder contains this structure:

- `bin`:
- `examples`:
- `manual`:
- `reference_data`:

Later, you must cd to `reference_data` and type:

```
wget ftp://polar.ncep.noaa.gov/tempor/ww3ftp/gridgen_addit.tgz
```

This will unpack the netcdf batymetry files:

- etopo1.nc
- etopo2.nc

and matlab binary files:

- coastal_bound_coarse.mat
- coastal_bound_high.mat
- coastal_bound_low.mat
- coastal_bound_full.mat
- coastal_bound_inter.mat
- optional_coastal_polygons.mat

For more information, it's recommended to visit the oficial web repository.

2.3 OASIS-MCT

Warning: If you are interested in coupling WW3 with other climate models (ice, ocean or atmosphere model) this step is necessary. If you are not, please skip it.

The OASIS is a software coupler that allows synchronizing exchanges of information between numerical codes and it's specially used to represent different components of the climate system. Current OASIS developers are CERFACS (Toulouse, France) and Centre National de la Recherche Scientifique - CNRS (Paris, France)

Due to this coupler must be interfaced, Model Coupling Toolkit (MCT) from the Argonne National Laboratory is used. This offers a fully parallel implementation of coupling field regridding and exchange. OASIS-MCT is a coupling library is linked with to the component models with the main function of interpolation and exchanging the coupling fields between these components.

If you want download any OASIS-MCT version, please visit the OASIS download site (LINK), after filling out the registration form, you can obtain the source code with:

WAY 1

or

WAY 2

This folder contains this structure:

- `doc`: OASIS-MCT documentation in .tex files.
- `examples`: tutorials and basic examples for beginners in OASIS-MCT complation.
- `lib`: FORTRAN files used to the OASIS-MCT compilation.
- `util`: editable Make files to adjust the local conditions to the OASIS-MCT compilation.

more information is available in LINK.

CHAPTER

THREE

INITIAL COMMANDS

3.1 Ejecutando el setup

Estando en la carpeta WW3, se debe scribir la siguiente línea en la terminal.

```
cd ruta/a/WW3
./model/bin/w3_setup model
```

Este comando ejecutará las siguientes acciones:

Preguntará por los programas previos que se necesitan para WAVEWATHCIII, como sigue:

- Compilador de Fortran
- Compilador de C++
- Printer
- Ruta scratch
- Ruta temporal

Después de las preguntas correrá la instalación de programas auxiliares, ejecutando los codigos en FORTRAN usando el compilador elegido. Después de la compilación de estos programas auxiliares se crean algunos archivos en el carpeta .model/aux o .model/bin

- `w3adc.f`: WAVEWATCH III FORTRAN preprocessor.
- `w3prnt.f`: Print files (source codes) including page and line numbers.
- `w3list.f`: Generate a generic source code listing.
- `w3split.f`: Creará el archivo de configuración del entorno `wwatch_env` a partir del setup. Este script crea/edita el archivo de configuración del entorno de WAVEWATCH III

`wwatch3.env` será creado en la carpeta `bin` y alojará las configuraciones que tendrá el modelo para cuando se compile. Este archivo puede ser re-editado volviendo a ejecutar el programa `w3_setup` o editandolo manualmente.

Adicionalmente en el directorio `model` deben aparecer los siguientes directorios:

- `exe`: WAVEWATCH III executables.
- `mod`: Module files.
- `obj`: Object files.

Note: Esta confguración no reemplaza las configuraciones que cada usuario debe realizar para ejecutar sus casos específicos. Solo representa una configuración inicial

3.2 Obteniendo los regtests

Estando en la carpeta WW3, se debe escribir la siguiente línea en la terminal.

```
./model/bin/ww3 from ftp.sh
```

Este comando se utiliza para obtener archivos de gran tamaño que no están almacenados en el repositorio de Git.

Al ejecutarla aparecen dos carpetas en el directorio superior:

- `data_regtests`: contiene toda la información de los regtests
- `cases`:

Un regtest en WAVEWATCH es un caso base creado para que los usarios del modelo se familiaricen

3.3 Errores comunes

- Escribir mal las rutas
- No tener algunas librerías necesarias instaladas

Todos siguen la misma línea:

```
w3_setup /home/user/WW3/model -c <comp> -s <switch>
```

CHAPTER

FOUR

SETTING UP THE ENVIRONMENT

WAVEWATCH III permite que los resultados obtenidos sean exportados en formato NetCDF-4. Para esto se deben verificar las versiones de la librería NETCDF instalada, así:

VERIFICACION

Cuando se ha verificado lo anterior, se crean las siguientes variables de entorno:

Si tiene NetCDF-3:

- `WWATCH3_NETCDF= NC3`

Si tiene NetCDF-4:

- `WWATCH3_NETCDF= NC4`
- `NETCDF_CONFIG`: es la ruta a NetCDF-4 nf-config utility program, regularmente es la carpeta `bin` en el que está construído la librería NETCDF

Estas variables de entorno se puedene definir de dos formas:

1. Escribir en la terminal:

```
export WWATCH3_NETCDF=NC4
export NETCDF_CONFIG=ruta/a/la/carpeta/bin/de/netcdf
```

Esto crea de forma temporal las variables mencionadas anteriormente

1. Se debe buscar en el directorio `HOME` un archivo llamado `.bashrc`. Este archivo *Qué es?*. En él se deben escribir las siguientes líneas

```
export WWATCH3_NETCDF=NC4
export NETCDF_CONFIG=ruta/a/la/carpeta/bin/de/netcdf
```

y luego compilarlo así:

```
source ~/.bashrc
```

Note: Si usa otra tipo de terminal, más ergonómica o tuneable que *bash*, debe buscar su archivo homónimo, por ejemplo: si tiene oh-my-zsh el archivo es *.zshrc*

CHAPTER

FIVE

CONTAINS AND COMPILATION

5.1 Principales programas

Hablar de que es en la carpeta bin dond enos vamos a mover y es importante tener un buen manejo y conocimiento de todo lo que haya en ella, no necesariamente script a scipt pero sí algo recordado

En `bin` se almacenan todos los comandos ejecutables de WW3 para la compilación del mismo.

- install ww3 tar Script to install WAVEWATCH III from tar files.
- `w3_setup`: Script for creating/editing the WAVEWATCH III environment setup file. The default setup file is bin/.wwatch3.env. switch and compiler are given in arguments. (see options, **link a un capítulo dedicado a setup**)
- `w3_clean`: Script to clean up WAVEWATCH III directories by removing files generated during compilation or test runs. 3 levels of clean*up. (see options). Cuando hago el clean es como si cambiara el molde pero no la panadería o el entorno de trabajo.
- `w3_make`: Script to compile and link components of WAVEWATCH III using a makefile. A list of programs can be given in arguments. Es para hacer los moldes del pan (o casos que quiero trabajar) **link a un capítulo dedicado a compilación**
- `w3_automake`: Script to automatically compile separately sequential and distributed programs in MPI, OMP or HYB depending on the switch file content. A list of programs can be given in arguments.
- `w3_new`: Script to touch all correct source code files to account for changes in compiler switches in combination with the makefile.
- `ww3_gspl.sh`: Script to automate use of ww3 gspl program (seeSection 4.4.11). ¿Crear multi…? para qué es …?
- arc wwatch3 tar Program to archive versions of WAVEWATCH III in the directory arc.
- `ww3_from_ftp.sh`: Script to download all the binary files not provided by git

Hablar de los más improtantes para el proyecto

5.2 Compilación

5.2.1 Creación del setup

Después de los dos comandos anteriores, WAVEWATCHIII está listo para ser compilado bajo la configuración que requiera el usuario

Para cualquier compilación se escribe el siguiente comando desde el directorio `bin`

```
w3_setup /home/user/WW3/model -c <comp> -s <switch>
```

En <comp> se escribe el nombre del compilador y en <switch> el nombre del switch

Compiladores

Switch

EJEMPLO DE UN SWITCH

Primer grupo

- F90: Swith que selecciona el código dependiente de la máquina. La otra opción disponible es: DUM

Segundo grupo

Aquí se define el modelo de kardware y el protocolo de transferencia de mensajes hay una pareja de opciones:

- SHRD shared memory model o DIST distribuites memory model (hardware)
- SHRD shared memory model o MPI message passing interface (messaging)

Notese que si se selecciona MPI, no puede seleccionarse SHRD en el hardware

Tercer grupo: Esquemas de propagación

- PRO, PR1,PR2,PR3,PRX: se elige uno de aquí
- PRO, PR1,UNO,UQ: se elige uno de aquí

Cuarto grupo: Selection of flux computation

- FLX0: Rutinas no usadas, flujo de computación incluida en los términos fuente
- FLX1: velocidad de fricción según una ecuación
- FLX2: vel fricción dede Tolman y CHalikov
- FLX3: Idem
- FLX4: vel fricción según otra ecuación
- FLXX: Definida por usuario

** Quinto grupo**: Selection of linear input * ln0: No linear input. * seed: Spectral seeding of Eq. (3.70). * ln1: Cavaleri and Malanotte*Rizzoli with filter. * lnx: Experimental (user supplied)

Sexto grupo: Selection of input and dissipation, stabn switches are optional and additional to corresponding stn switch:

- st0 No input and dissipation used.
- st1 WAM3 source term package.
- st2 Tolman and Chalikov (1996) source term package. See also the optional stab2 switch.

- stab0 No stability correction. Compatible with any source term (st) package. Including this switch has no effect.
- stab2 Enable stability correction (2.95) * (2.98). Compatible with st2 only.
- st3 WAM4 and variants source term package.
- stab3 Enable stability correction from Abdalla and Bidlot (2002).Compatible with st3 and st4 only.
- st4 Ardhuin et al. (2010) source term package.
- st6 BYDRZ source term package.
- stx Experimental (user supplied).

Séptimo grupo: Selection of nonlinear interactions: * nl0 No nonlinear interactions used. * nl1 Discrete interaction approximation (DIA). * nl2 Exact interaction approximation (WRT). * nl3 Generalized Multiple DIA (GMD). * nl4 Two*scale approximation (TSA). * nlx Experimental (user supplied)

Octavo grupo: Selection of bottom friction: * bt0 No bottom friction used. * bt1 JONSWAP bottom friction formulation. * bt4 SHOWEX bottom friction formulation. * bt8 Dalrymple and Liu formulation (fluid mud seafloor). * bt9 Ng formulation (fluid mud seafloor). * btx Experimental (user supplied)

Noveno grupo: Selection of term for damping by sea ice: * ic0 No damping by sea ice. * ic1 Simple formulation. * ic2 Liu et al. formulation. * ic3 Wang and Shen formulation. * ic4 Frequency*dependent damping by sea ice

Décimo grupo: Selection of term for scattering by sea ice: * is0 No scattering by sea ice. * is1 Diffusive scattering by sea ice (simple). * is2 Floe*size dependent scattering and dissipation

Onceavo grupo: Selection of term for reflection: * ref0 No reflection. * ref1 Enables reflection of shorelines and icebergs

Décimo grupo: Selection depth*induced breaking of : * db0 No depth*induced breaking used. * db1 Battjes*Janssen. * dbx Experimental (user supplied)

Onceavo: Selection of triad interactions: * tr0 No triad interactions used. * tr1 Lumped Triad Interaction (LTA) method. * trx Experimental (user supplied)

Selection of bottom scattering: * bs0 No bottom scattering used. * bs1 Magne and Ardhuin. * bsx Experimental (user supplied).

Selection of supplemental source term: * xx0 No supplemental source term used. * xxx Experimental (user supplied).

Selection of method of wind interpolation (time): * wnt0 No interpolation. * wnt1 Linear interpolation. * wnt2 Approximately quadratic interpolation.

Selection of method of wind interpolation (space): * wnx0 Vector interpolation. * wnx1 Approximately linear speed interpolation. * wnx2 Approximately quadratic speed interpolation

Selection of method of current interpolation (time): * crt0 No interpolation * crt1 Linear interpolation. * crt2 Approximately quadratic interpolation.

Selection of method of current interpolation (space): * crx0 Vector interpolation * crx1 Approximate linear speed interpolation. * crx2 Approximate quadratic speed interpolation.

Switch for user supplied GRIB package. * nogrb No package included. * ncep1 NCEP GRIB1 package for IBM SP. * ncep2 NCEP GRIB2 package for IBM SP

Opcional switches * o0: imprime el namelist el el preprocesador de la malla (ww3_grid) * o1: imprime los puntos límite en el preprocesador de la malla (ww3_grid) * o2: imprime el mapa de estado de los mapas (ww3_grid) * o3: imprime el ciclo de cada paso de la compilación en el preprocesador del campo (ww3_prnc) * o4: Print plot of normalized one*dimensional energy spectrum in initial conditions program (w3_strt) * o5: Id. two*dimensional energy spectrum. * o6: Id. spatial distribution of wave heights (not adapted for distributed memory). * o7: Echo input

data for homogeneous fields in generic shell. * o10: Identify main elements of multi*grid model extensions in standard output. * o11: Additional log output on management algorithm in log.mww3

Cuando se corre `w3_setup` se crearán 3 scripts/archivos en el directorio `bin`:

- `comp`: script compilador. Este script está basado en la plantilla comp.tmpl con la definición del compilador y sus opciones. <*comp*> es el compilador que se desea emplear, por lo tanto, es bueno que se verifique cuál es el más aconsejable para su sistema operativo.
- `link`: script enlazador. Este script está basado en la plantilla link.tmpl con la definición del compilador y sus opciones
- `switch`: es el archivo que contiene todos los switches que activan las configuraciones que queremos y que son reconocidos por el preprocesador w3adc. Copied from `switch <switch>`

todos los posible archivos comp, swith y link están en el directorio `bin`

Cuando la sentencia anterior se ejecuta algo como esto:

GIF

El archivo de entorno wwatch3.env o actualizado si ya existe. The auxiliary FORTRAN programs for code prepocessor will be compiled using by default the GNU FORTRAN compiler which can be different to the provided compiler for the WAVEWATCH III programs.

¿Qué creo esto comando?

se crea w3_split, w3_prnt, w3_list,w3_adc, switch, comp y link

- También crea un make_makefile.sh

GIF

Si desea cambiarse de configuración se recomienda ejecutar el script `w3_clean```desde lel directorio ``bin`. Para más información ver este [literal](#limpieza*de*las*configuraciones*realizadas):

5.2.2 Compilación a partir de las configuraciones hechas

Existen principalemente 2 formas de compilar

w3_make

Se usa w3_make para compotlar WAVEWATCH, si es usa sin parámetros se usan todos los programs básicos, sino los que acompañen la ejecucuoón de w3_make

w3_automake

Para una compilación selectiva

w3_automake

or for a few programs

w3_automake ww3_grid ww3_shel ww3_ounf

Se crean las siguiente carpetas:

Relativas a ejecuciones en serie * mod_SEQ: * obj_SEQ * work_SEQ * tmp_SEQ: se alamacenan los errores, warnings y demás que ocrrieron durante la compilación en serie

Relativas a ejecuciones en paralelo * mod_MPI: * obj_MPI * work_MPI * tmp_MPI: se alamacenan los errores, warnings y demás que ocrrieron durante la compilación en paralelo

Adicionalmente, los programas compilados serán todos almacenados en el directorio *exe* junto a una copia de los archivos switch, comp y link usados en el setup y sus correspondientes a SEQ y MPI

Limpieza de las configuraciones realizadas

Uso de *w3_clean *c* para tal comandos

Uso de *w3_clean *m* para tal comandos

Aprender a bajar datos de un ftp como el de ifremer

CHAPTER

SIX

TEST CASES

Lista de testcases

CHAPTER

SEVEN

PREPARE CONFIGURATION

preparar

CHAPTER

EIGHT

COMPILATION

Conociendo toda la configuración del Switch del mar Adriático

Switch Los parámetros que tiene el switch del mar adriático son:

F90 NOGRB NC4 SHRD PR3 UQ FLX0 LN1 ST4 NL1 BT1 DB1 TR0 BS0 IC0 IS0 REF0 XX0 WNT1 WNX1 CRT1 CRX1 O0 O1 O2 O3 O4 O5 O6 O7 O10 O11

reccordar que los programas en bash son ejecutados con un `.` antes de la ruta

CHAPTER

NINE

PREPROCESSING

Archivos de entrada

ww3_grid

inputs - ww3_grid.inp - archivo de batimetría (.bottom) - archivo de mallado de obstruccione (.obstr) - archivo mallada de máscara (.mask)

Los últimos 3 archivos son generados por la herramienta gridgen, por lo que es importante

output - Standard out (Salida formateada del programa) - mod_def.ww3 (Archivo de definición de modelo en formato WAVEWATCH III.) - mask.ww3 (Archivo de máscara de tierra-mar (interruptor o2a).)

Parámetros que se definen - Frecuencia inicial y factor de incremento de la frecuencia - número de frecuencias (Información espectral) - Información de paso de tiempo - Resolución

ww3_prep

inputs - ww3_strt.inp - mod_def.ww3

output - Standard out - restart.ww3

Parámetros que se definen Define el tipo de archivo inicial con el cual se ejecuta el modelo, elige el **tipo de espectro que se empleara en las condiciones iniciales**

- ww3_prep.inp
- mod_def.ww3
- Standard out
- level.ww3, current.ww3, wind.ww3, ice.ww3
- data0.ww3, data1.ww3,data2.ww3
- Forzadores del modelo
- Magnitudes de la malla
- Se define donde arranca y donde termina (LAT - LON)
- ww3_shel.inp
- mod_def.ww3, level.ww3,

  current.ww3, wind.ww3, ice.ww3, restar.ww3
- nest.ww3, data0.ww3, data1.ww3,

  data2.ww3
- track_i.ww3

-Standard out -log.ww3, test.ww3

restarn.ww3, nestn.ww3 -out_grd.ww3 -out_pnt.ww3

- Ejecución del modelo
- Tiempo de cálculo, cuando

empieza y cuando termina - Definir el tipo de archivo de salida del modelo

output del modelo

se recomienda el uso de ncview y ncdump para la manipulaicón de archivos en formato netCDF

Gridgen

CHAPTER

TEN

EXECUTE THE MODEL

Visualización de los resultados

ncdump ncview

CHAPTER

ELEVEN

AW SIMULATIONS IN BOCHICA

11.1 Initial folders

Initially, a folder havo to be created where the coupled simulation will be executed (WORKDIR). This folder have to be in the `runs` directory:

```
mkdir aw_HURRICANE_NAME
cd aw_HURRICANE_NAME
```

3 subfolders must be created/copied in this directory:

1. `input_files`: is a folder where the initial wrf files must be putted.

```
mkdir input_files
cp path/to/wrf/files/* input_files/
```

2. `exec_files`: is a folder where all executable files to prepare the coupled simulation are hosted. This folder is in the root directory of the user

```
cp -r ~/exec_files/ .
```

Note: In this directory there is a subfolder with the executable files for the coupled run from a local machine (`exec_from_local`). This folder must be copied to the local machine in order to execute them there.

---

3. An empty folder where will be the final files before to launch the simulation

```
mkdir cpl_files
```

11.2 Running executables

It must be edited `init.sh` file according to the folder initially created. This file is executed after all necessary modifications:

```
./exec_files/init.sh
```

In the local machine must be created a similar folder to `aw_HURRICANE_NAME` like this:

```
mkdir aw_HURRICANE_NAME
cd owa_HURRICANE_NAME
```

A folder with the initial batimethric files must be created in `owa_HURRICANE_NAME`, these files are copied for the original directory.

```
mkdir input_files_ww3
cp path/to/bath/files/* input_files_ww3/.
```

Inside this directory it is also copied the folder with the executable files aforementioned. The first file to edit is `convert_wrf_and_croco_input.sh` according the required settings. After this, `convert_wrf_and_croco_input.sh` must be runned in order to preprare the input files to the noncoupled simulation of ww3:

```
sshpass -f ~/pass_file scp -r ffayalac@168.176.123.121:exec_files/exec_from_local/ .
mv exec_from_local/* .
rm -rf exec_from_local
./convert_wrf_and_croco_input.sh
```

The non coupled simulation is generated in the cluster and it is used to obtain the ww3 initial file via `frc_ww3.sh` bash script. This initial file is later employed to generate the initial OASIS files. Please edit the script according to the requirements:

```
./exec_files/frc_ww3.sh
```

Subsequently, another file in the local machine is executed. With `./prep_oasis_files.sh` script are created the initial files for OASIS (`wav.nc` and `atm.nc`). Edit it and run it like this:

```
./prep_oasis_files.sh
```

In order to set-up the required files to launch the coupled run, it must be edited the ``aw_files.sh``script and later it is executed:

```
./exec_files/aw_files.sh
```

Finally, the coupled simulations is launched after to edit the `submit.pbs`

```
qsub exec_files/submit_aw.pbs
```

CHAPTER

TWELVE

OWA SIMULATIONS IN BOCHICA

12.1 Initial folders

Initially, a folder havo to be created where the coupled simulation will be executed (WORKDIR). This folder have to be in the `runs` directory:

3 subfolders must be created/copied in this directory:

1. `input_files`: is a folder where the initial croco and wrf files must be putted.

Note: In this directory there is a subfolder with the executable files for the coupled run from a local machine (`exec_from_local`). This folder must be copied to the local machine in order to execute them there.

3. An empty folder where will be the final files before to launch the simulation

---

12.2 Running executables

It must be edited `init.sh` file according to the folder initially created. This file is executed after all necessary modifications:

In the local machine must be created a similar folder to `owa_HURRICANE_NAME` like this:

A folder with the initial batimethric files must be created in `owa_HURRICANE_NAME`, these files are copied for the original directory.

Inside this directory it is also copied the folder with the executable files aforementioned. The first file to edit is `convert_wrf_and_croco_input.sh` according the required settings. After this, `convert_wrf_and_croco_input.sh` must be runned in order to preprare the input files to the noncoupled simulation of ww3:

The non coupled simulation is generated in the cluster and it is used to obtain the ww3 initial file via `frc_ww3.sh` bash script. This initial file is later employed to generate the initial OASIS files. Please edit the script according to the requirements:

Subsequently, another file in the local machine is executed. With `./prep_oasis_files.sh` script are created the initial files for OASIS (`wav.nc` and `atm.nc`). Edit it and run it like this:

In order to set-up the required files to launch the coupled run, it must be edited the ``owa_files.sh``script and later it is executed:

Finally, the coupled simulations is launched after to edit the `submit.pbs`

CHAPTER

THIRTEEN

DISCLAIMER

This tutorial is not intended to supplant the official WW3 documentation

CHAPTER

FOURTEEN

COMMENTS/HELP

If you have questions regarding WAVEWATCH III, or problems with this Online Tutorial, please let us know them in WWIII Forum (Link) or in the email something@gmail.com