Running the model, single grid version

Roberto Padilla

The WAVEWATCH III Team + friends
Marine Modeling and Analysis Branch
NOAA / NWS / NCEP / EMC

NCEP.list.WAVEWATCH@NOAA.gov
NCEP.list.waves@NOAA.gov

Atmospheric and Oceanic Science

Covered in this lecture:

- Running the model.
  - WAVEWATCH III codes.
  - Running the essentials (*ww3_shel*).
- Processing model input.
- Process model output.
- Nesting (*ww3_shel*).
- *ww3_shel* final remarks.

Running the model

The model consists of a set of programs :

- Grid preprocessor (*ww3_grid*).
- Initial conditions program (*ww3_strt*).
- Boundary data from external sources (*ww3_bound*).
- Model input preprocessor (*ww3_prep*).
- The actual wave model (*ww3_shel*, *ww3_multi*).
- Output post-processors:
  - Numerical (*ww3_outp*, *ww3_outf*, *ww3_trck, ww3_grib, nc_outf nc_outp*).
  - GrADS (*gx_outf, gx_outp*).
- Regridding (*ww3_gint, ww3_gspl*)

Introduced in version 4.xx
Introduced in version 3.14
Introduced in version 2.22

Running the model

Basics :

- All programs read their instructions from a a file with the extension ***.inp***, for instance ***ww3_grid*** reads from ***ww3_grid.inp***.
- The first character of the first line of the ***.inp*** file identifies the comment character.
  - Lines starting with this character are skipped while reading.
  - In all examples used by us the comment character is ‘$’ .
- All programs write output to standard output.
- All raw output files of the wave model are identified by the extension ***.ww3***. (exception: running ***ww3_multi***).
- This presentation will focus on running the conventional single-grid wave model ***ww3_shel***.
  - ***ww3_multi*** will be discussed on day 2.

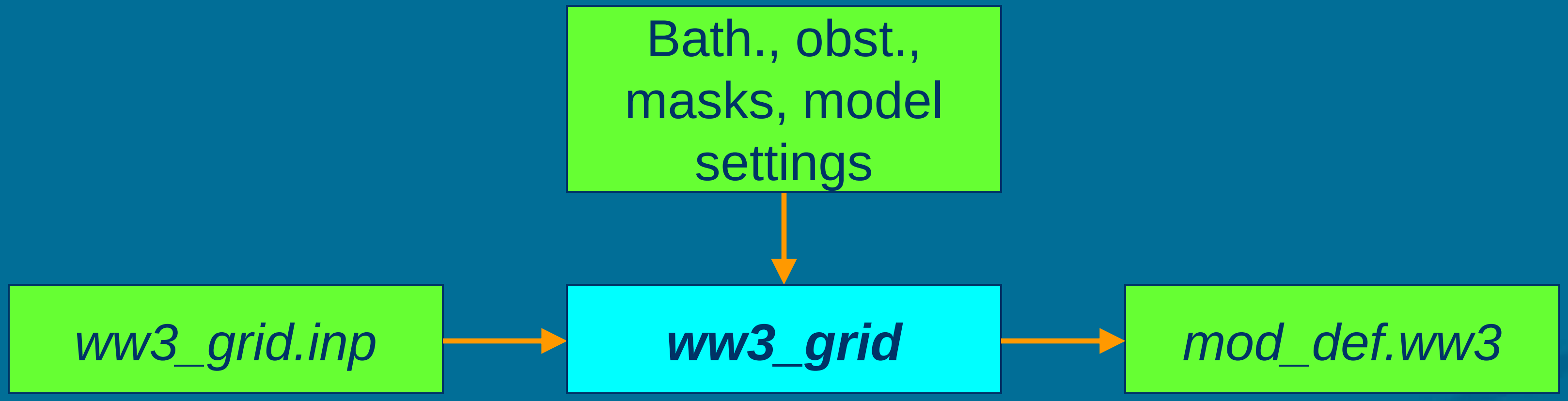
Cont’ed :

- Before anything else can be done, an application is defined using ***ww3_grid*** (run once). resulting in a model definition file ***mod_def.ww3*** (use often) This file contains:
  - Bathymetry, obstruction grids, masks.

    select at compile
  - Parameter settings for all physics and numerical approaches (including flags and dry-run option).
  - Data fields for above approaches as needed.
  - Time steps for grid.
  - Input boundary points for grids.
  - Output boundary points for nested grids if off-line nesting is performed (***ww3_multi*** does in-line nesting).
    - Points are defined, but output is controlled by ***ww3_shel***.

Running ww3_grid :

- ***ww3_grid*** is typically run once, and ***mod_def.ww3*** is then stored for use with all other programs.



Bath., obst., masks, model settings

ww3_grid.inp

ww3_grid

mod_def.ww3

- Run code once, use file often ….

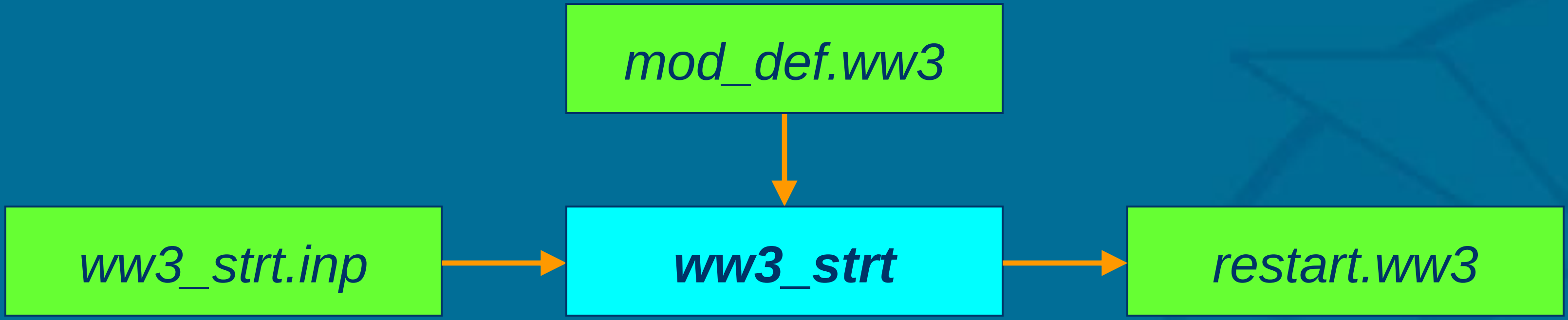
file, data

program

data flow

Creating initial conditions :

- Initial conditions are generated using ***ww3_strt.*** Options:
  - Gauss / cosine (idealized swell).
  - JONSWAP, Gauss in space.
  - JONSWAP, fetch limited based on local wind and grid spacing.
  - Single user-defined spectrum.
  - All at rest (new in 3.14).



Cont’ ed :

- Preprocessor ***ww3_strt*** and/or file *restart.ww3* not mandatory. If the file is not found, the main program will assume:
  - Initializing with local wind if no linear growth option is selected.
  - Otherwise start from rest.

- Note that the initial condition program can be used to set steady-state boundary conditions by defining points for input boundary data but by not providing a file with such data.
  - Dynamically varying externally defined boundary data can be used using ***ww3_bound*** (new in 4.08)

Running the model

With the ***mod_def.ww3*** and ***restart.ww3*** files we have enough to run a basic wave model.

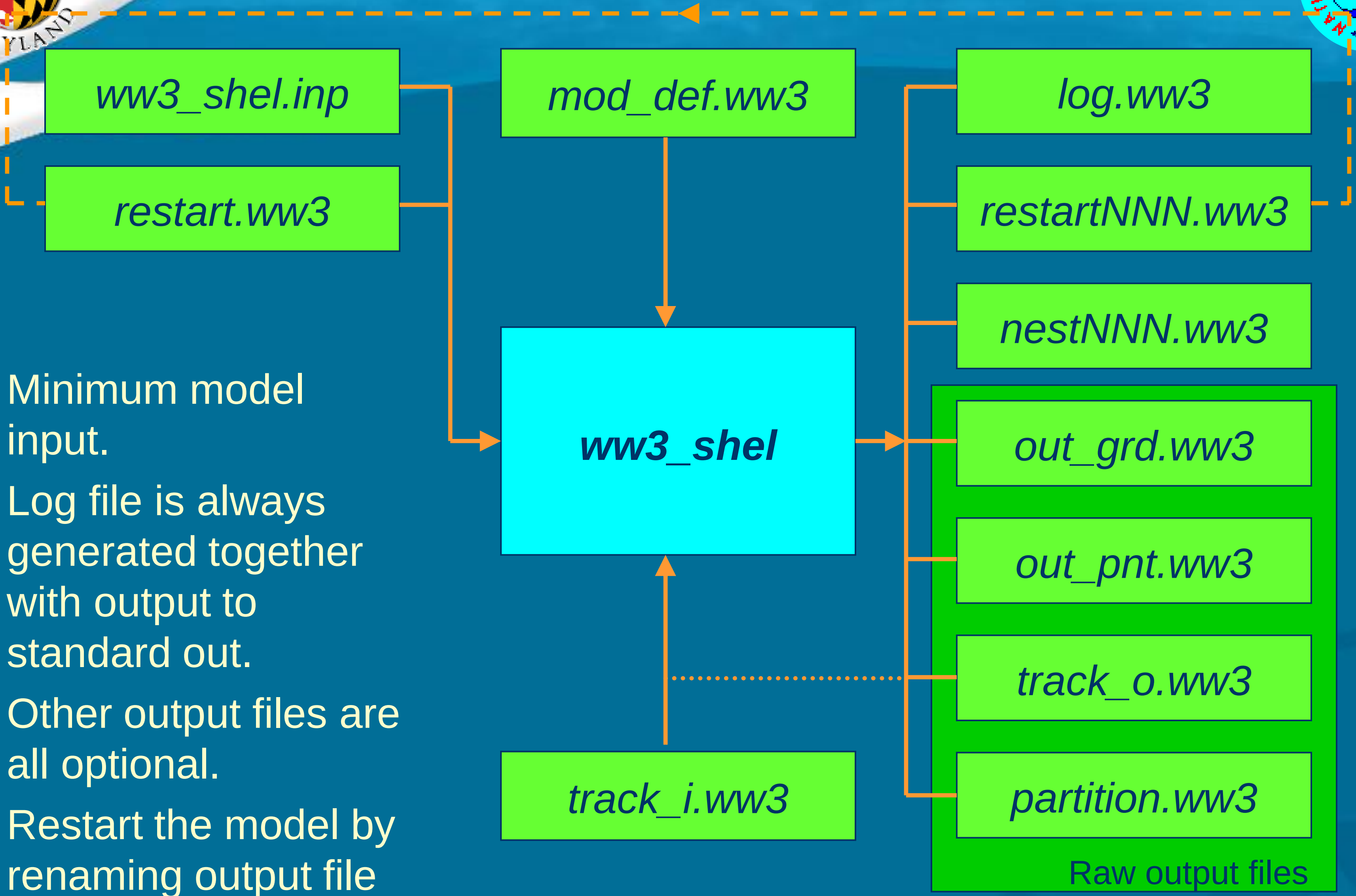
The input file for the basic wave model (***ww3_shel.inp***) contains :

- Definition of inputs to use (level, current, wind, ice; homogeneous or variable).
  - Homogeneous input data makes it possible to run model without separate input data files.
- Start and end time.
- Output requests for six types of standard output.
- Homogeneous input field data if requested.

ww3_shel.inp cont'ed:

- Output needs to be selected in ***ww3_shel.inp***.
  - Gridded output fields.
  - Point output containing full spectral data.
  - Nesting data for higher resolution grids.
    - Only output times set in ***ww3_shel.inp***.
    - Up to nine different files.
  - Restart files.
    - Up to nine different files.
  - Output of spectra along pre-defined tracks.
  - Partitioned spectral data (new in 3.14).

Running the model



- Minimum model input.
- Log file is always generated together with output to standard out.
- Other output files are all optional.
- Restart the model by renaming output file to input file.

Model output

- The ***ww3_shel*** model generates two formatted output files.
  - std out output tracks progress “on screen”.
  - ***log.ww3*** is generated by the actual wave model routines. This file is provided to assure consistent info on the model when the routine is included in an integrated model.

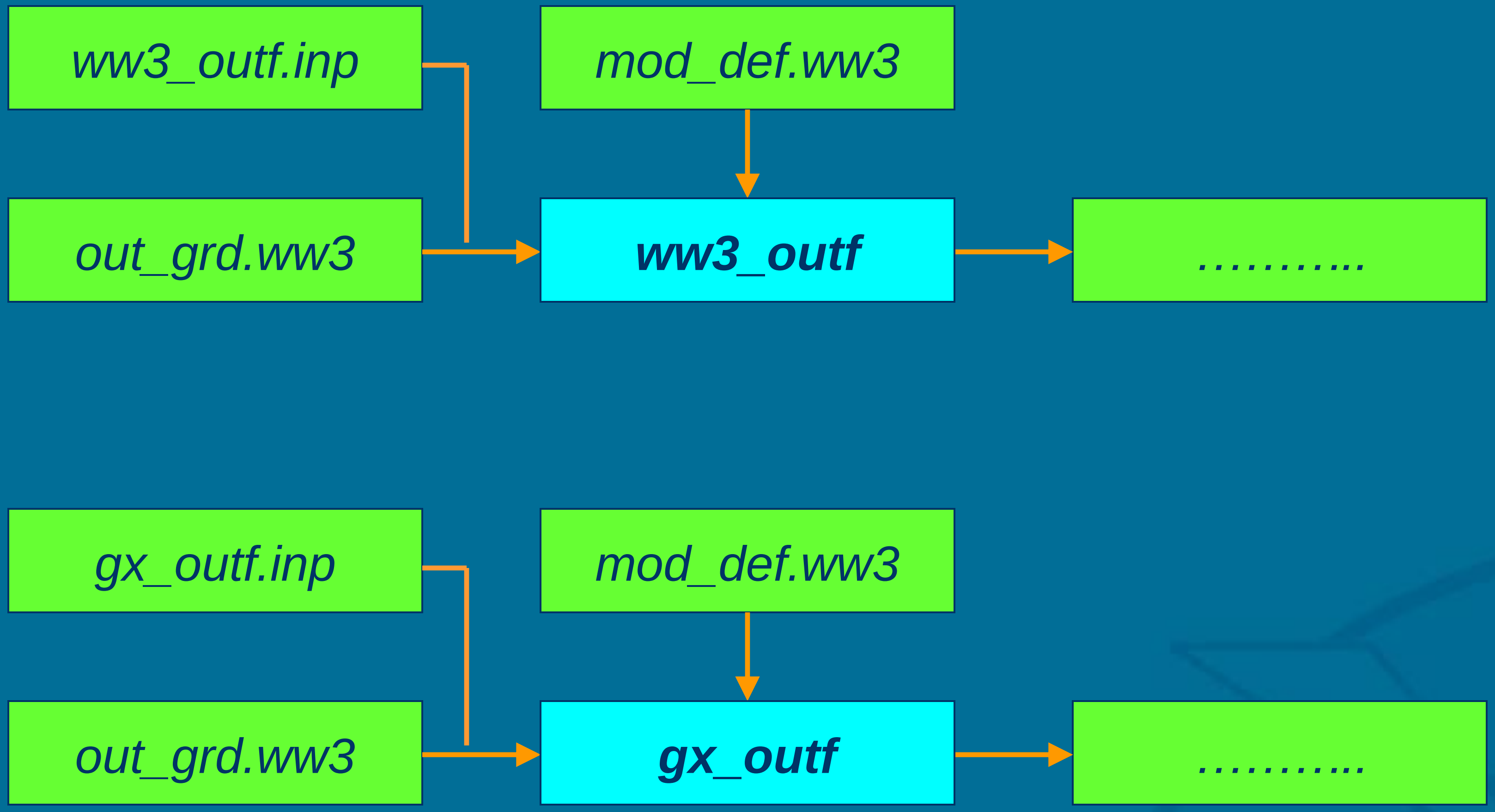
- The raw data files and the output restart file are only generated upon request (***track_i.ww3*** only needed for request). They are unformatted and partially packed.
  - ***restartN.ww3*** and ***nestN.ww3*** are used by subsequent model runs as is (after renaming).
  - All other raw output files require postprocessing.

Model output

- The file ***out_grd.ww3*** contains fields of mean wave parameters such as the significant wave height, and is processed using ***ww3_outf***. ***ww3_outf*** produces :
  - File inventories of ***out_grd.ww3***.
  - Print plots of fields.
  - Statistics for given ranges of the grid
  - Data transfer files.

- The file ***out_grd.ww3*** can be converted into GrADS data files using ***gx_outf***, or NetCDF using ***ww3_ounf***. Both operate similar to ***ww3_outf***.

Model output


ww3_outf.inp

mod_def.ww3

out_grd.ww3

ww3_outf

………...

gx_outf.inp

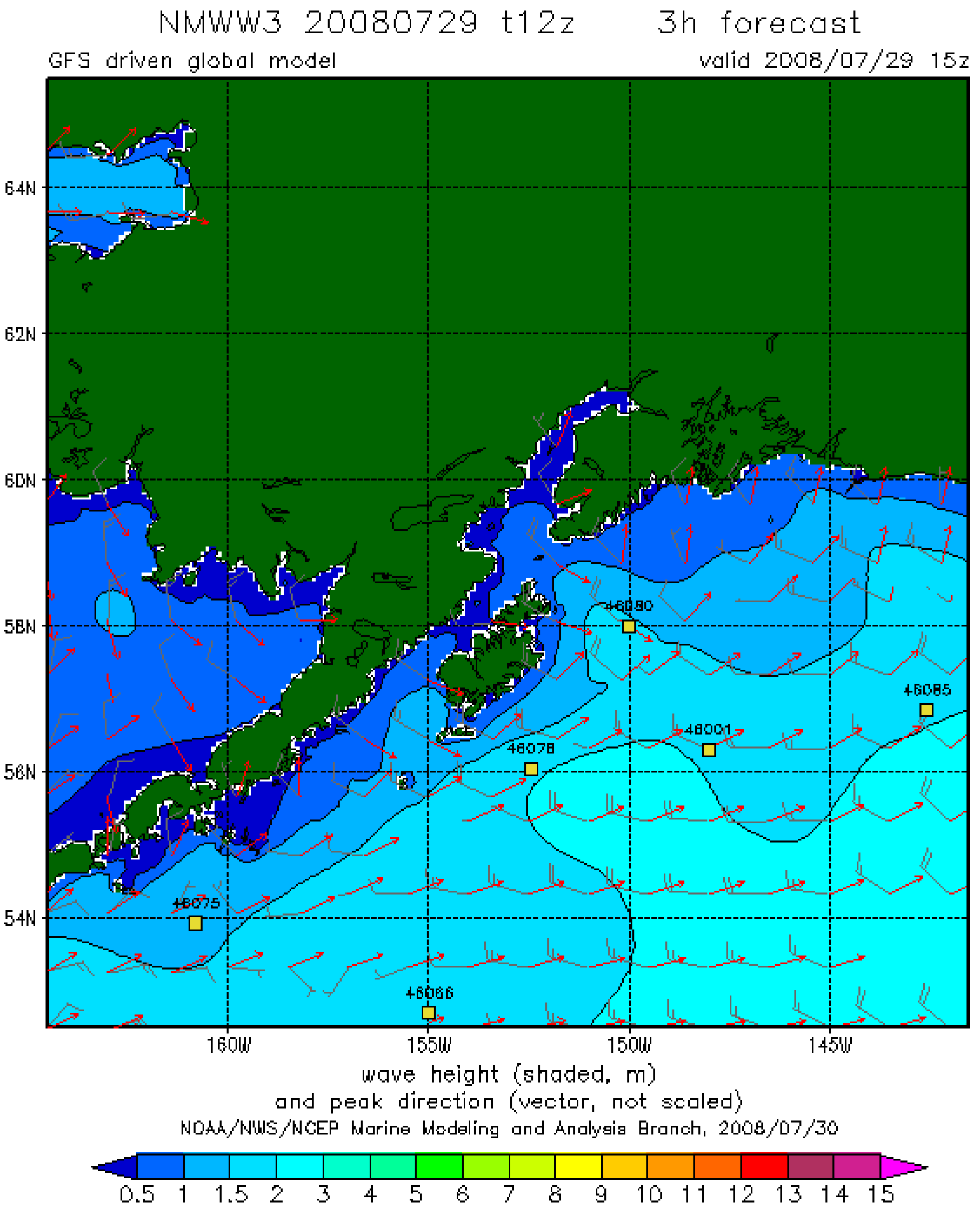
mod_def.ww3

out_grd.ww3

gx_outf

………...

Model output



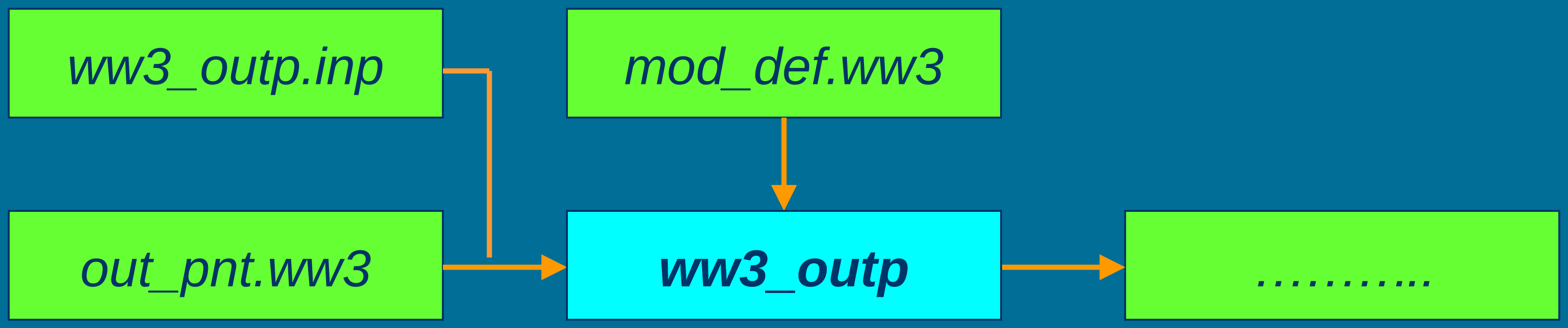
Example of GraDS output for present operational resolution for Cook Inlet in Alaska, generated after post-processing ***out_grd.ww3*** with ***gx_outf.***

Model output

- The file ***out_pnt.ww3*** contains full spectral data for selected output points, and is processed using ***ww3_outp***. ***ww3_outp*** produces :
  - File inventories of ***out_pnt.ww3***.
  - Spectral data as print-plots, tables of 1-D spectra and data transfer files.
  - Tables of mean environmental and wave parameters.
  - Source term data similar to that of spectra.

- The file out_pnt.ww3 can be converted into GrADS data files using ***gx_outp***, or NetCDF using ***ww3_ounp***.

Model output



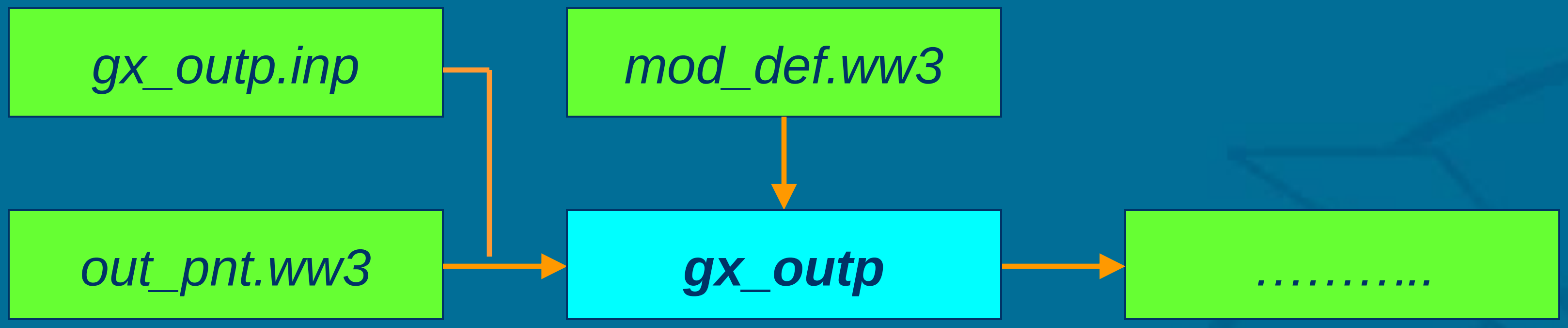
ww3_outp.inp

mod_def.ww3

out_pnt.ww3

ww3_outp

…………



gx_outp.inp

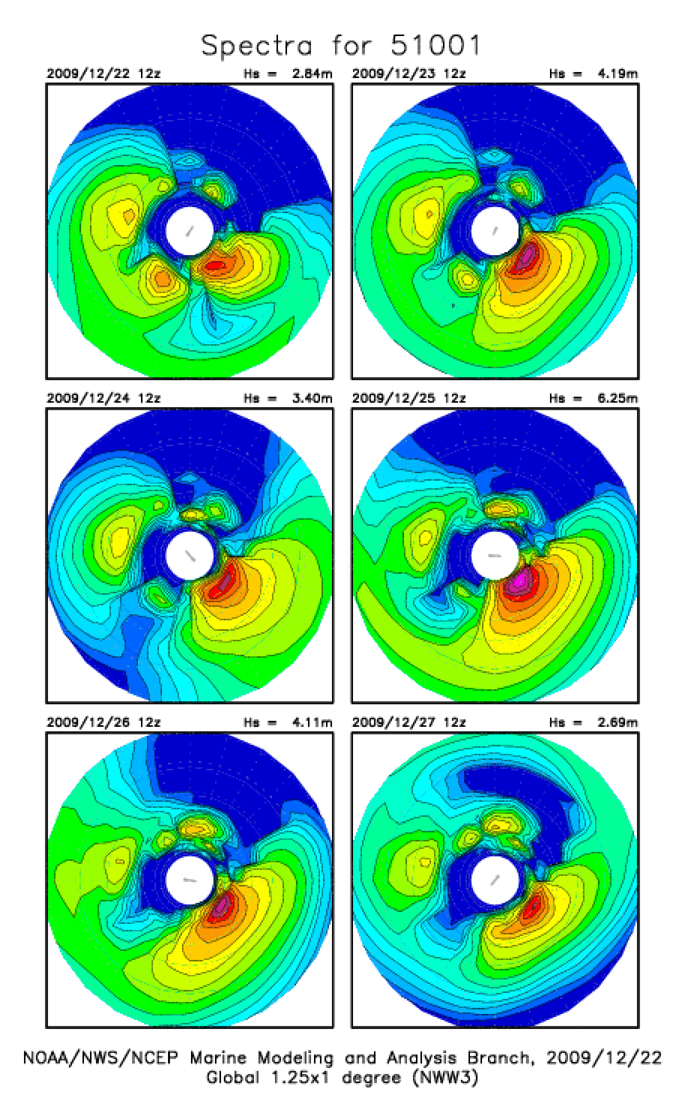
mod_def.ww3

out_pnt.ww3

gx_outp

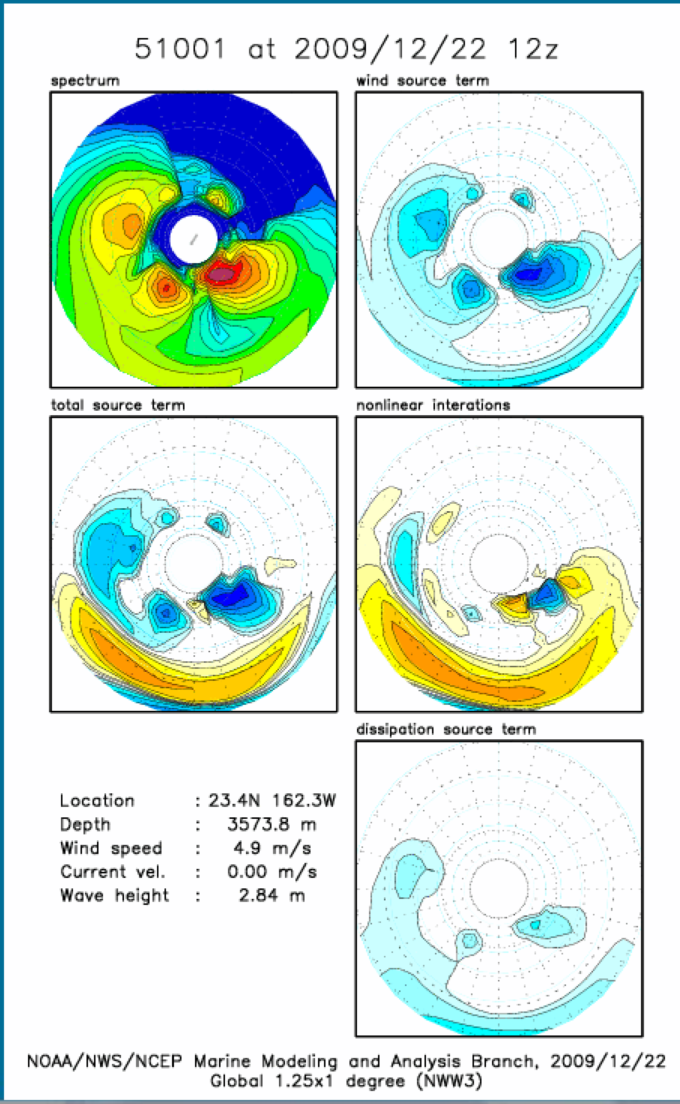
…………

Model output



NOAA/NWS/NCEP Marine Modeling and Analysis Branch, 2009/12/22
Global 1.25x1 degree (NWW3)

Example of GraDS spectral and source term output made by processing ***out_pnt.ww3*** with ***gx_outp***, and using GrADS script provided with the wave model distribution.

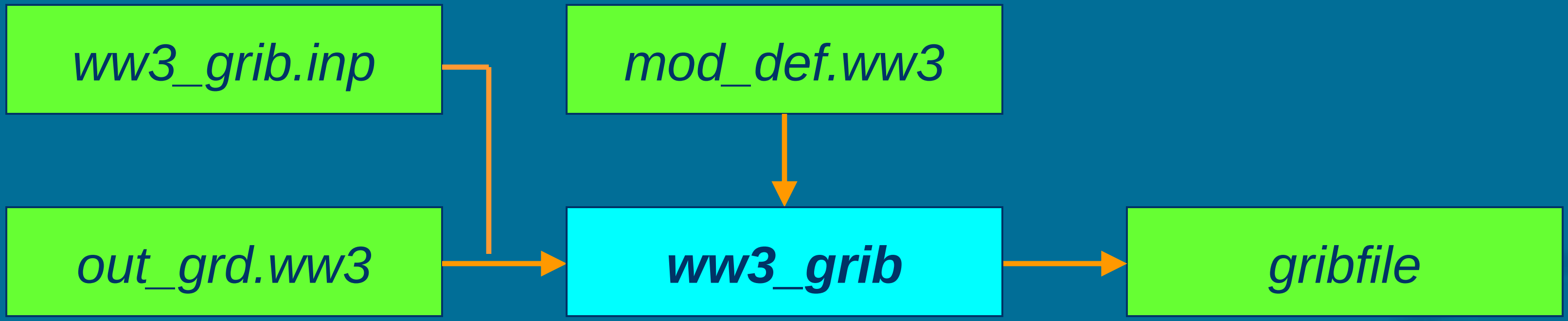


Location	:	23.4N 162.3W
Depth	:	3573.8 m
Wind speed	:	4.9 m/s
Current vel.	:	0.00 m/s
Wave height	:	2.84 m

NOAA/NWS/NCEP Marine Modeling and Analysis Branch, 2009/12/22
Global 1.25x1 degree (NWW3)

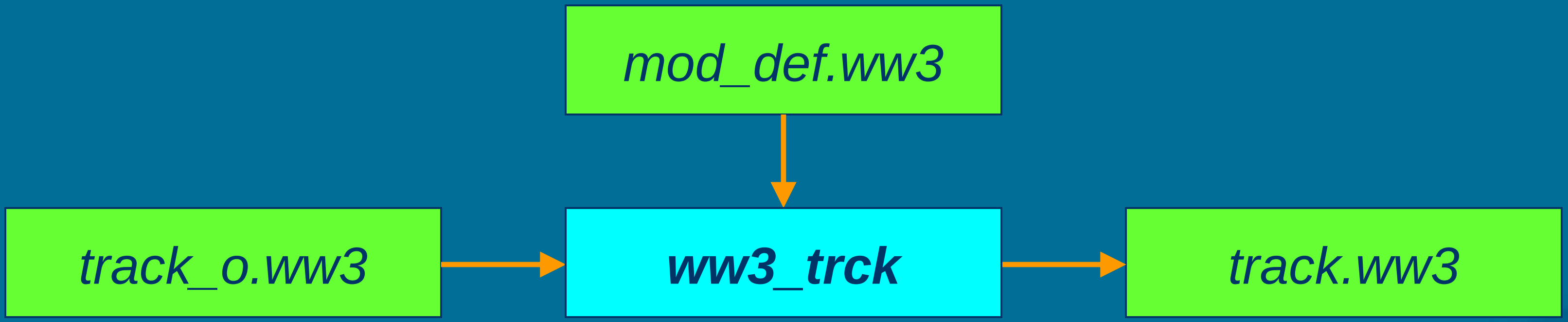
Model output

- ***ww3_grib*** takes the raw data file ***out_grd.ww3*** and converts it to a GRIB or GRIB2 file.
- GRIB and GRIB2 packing routines are not provided with WAVEWATCH III yet.



Model output

- ***ww3_trck*** takes the unformatted, direct access file ***track_o.ww3***, and repacks it in a free format readable, integer packed data file ***track.ww3***. No file with command data is needed.



Model input

So far, we have been running the model with homogeneous input fields for

- water levels
- currents,
- winds,
- and ice.

None of these input fields are mandatory, and a mixture of homogeneous on non-homogeneous fields can be used.

Model input

Dynamic input fields are pre-processed and written to files with a standard WAVEWATCH III format.

- This separates processing input from the wave model.

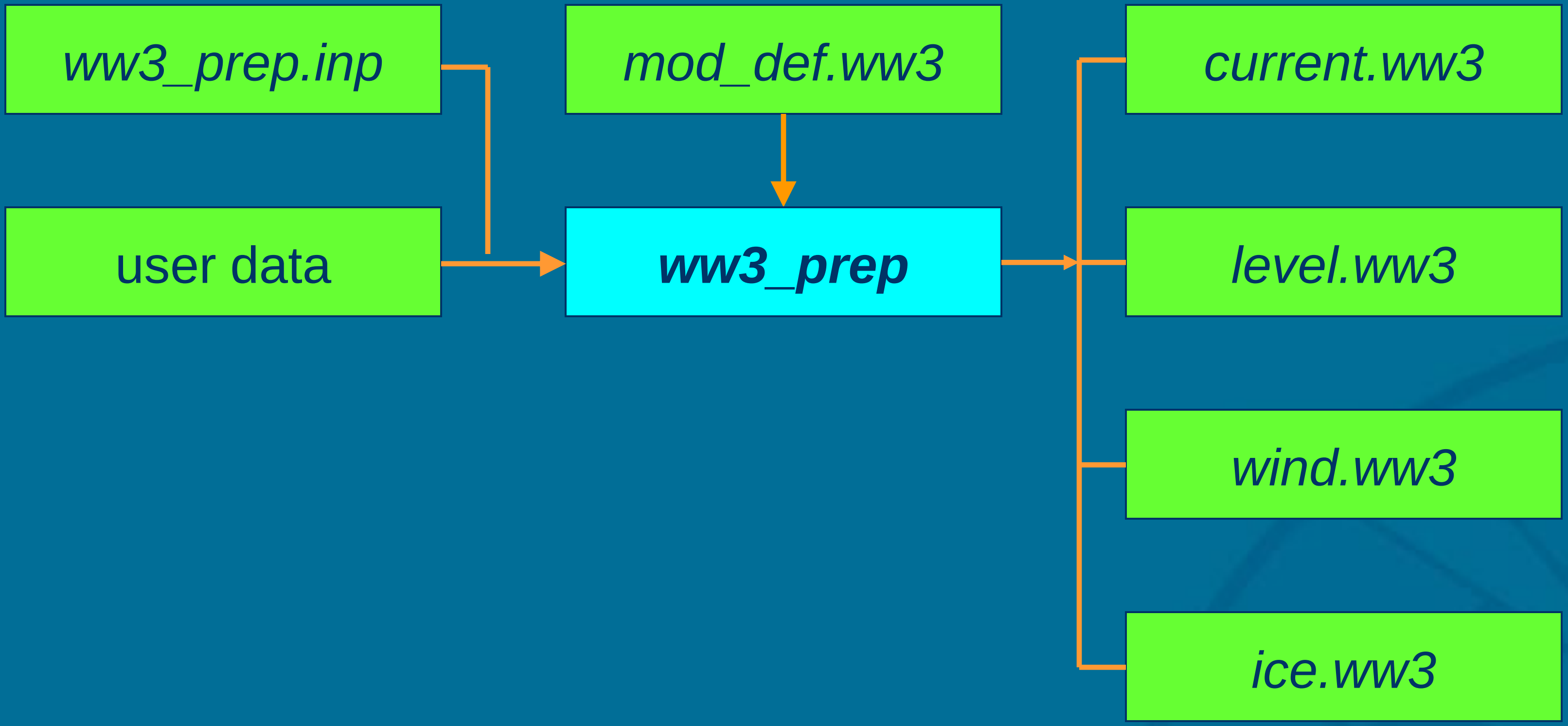
In the standard WAVEWATCH III data files, each data field has its own time stamp. ***ww3_shel*** deals with these time stamps on a field by field basis. This implies that :

- The model does not require input fields at constant intervals.
- The user need not supply field interval time steps in any of the programs as they are meaningless.

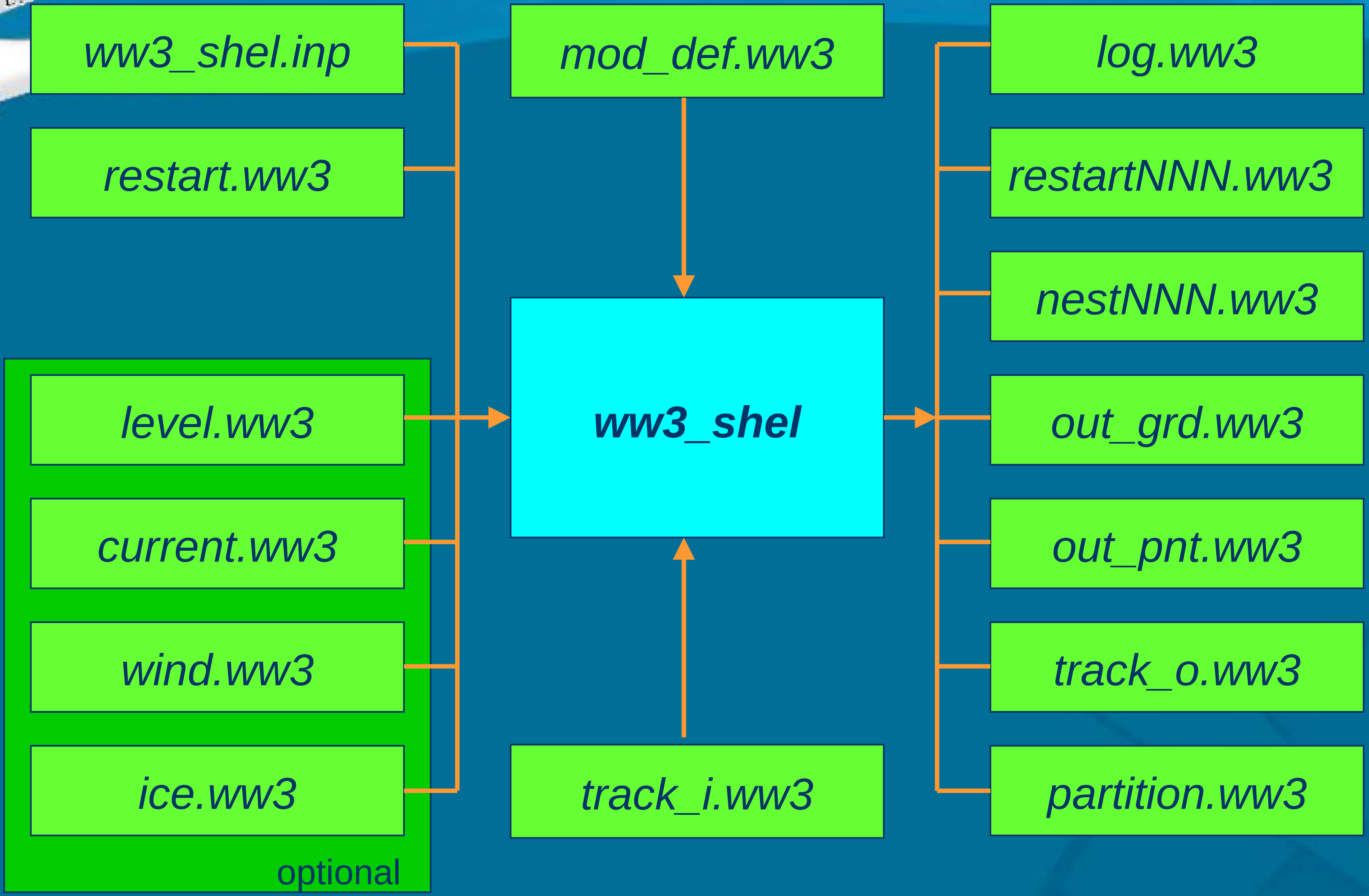
Input data are kept constant before the first and after the last time stamp in the file.

Model input

- Preprocessing of input fields is performed by ***ww3_prep***. This program deals with all four types of input fields (one at a time) and can deal with user-supplied fields of several formats.



Model input



ww3_shel.inp

restart.ww3

level.ww3

current.ww3

wind.ww3

ice.ww3

optional

mod_def.ww3

ww3_shel

track_i.ww3

log.ww3

restartNNN.ww3

nestNNN.ww3

out_grd.ww3

out_pnt.ww3

track_o.ww3

partition.ww3

Model input

- The user might need to repack his/her fields in order for ***ww3_prep*** to be able to read it. For instance, there is no GRIB or NetCDF option in the code for reasons of portability.

- HINT : Use the dry run option in ***ww3_grid*** to run all input fields through the wave model without doing actual wave calculations. This gives you a change of checking the data flow without having to waste expensive clock cycles.

Nesting (ww3_shel)

- Nesting in ***ww3_shel*** (as in most wave models) is a one way street. Boundary data is generated by a large scale model and read by a small scale model.
- Presently, up to 999 individual sets of boundary data can be generated by a single run.
- Data transfer for nesting is set up in ***ww3_grid(.inp)***.
- Execution of nesting depends on time step requests in ***ww3_shel.inp***, and availability of the file ***nest.ww3*** for the small scale run.
- If the file ***nest.ww3*** is not found, boundary data is kept constant at designated boundary points. This can be a handy feature for steady-state tests.

Nesting (ww3_shel)

Nesting can be set up in the following way : (one-way, ***ww3_shel***)

- Set up a large scale grid and a small scale grid using ***ww3_grid*** with separate input files ***ww3_grid.inp***.
- Define input boundary points in the small scale grid by defining discrete grid points as such. The output of ***ww3_grid*** will echo the corresponding locations if this output is requested at compile level.
- Add these grid points as output grid points in ***ww3_grid.inp*** for the large scale run and run ***ww3_grid*** to assure that data can be generated.

Nesting set-up cont-d :

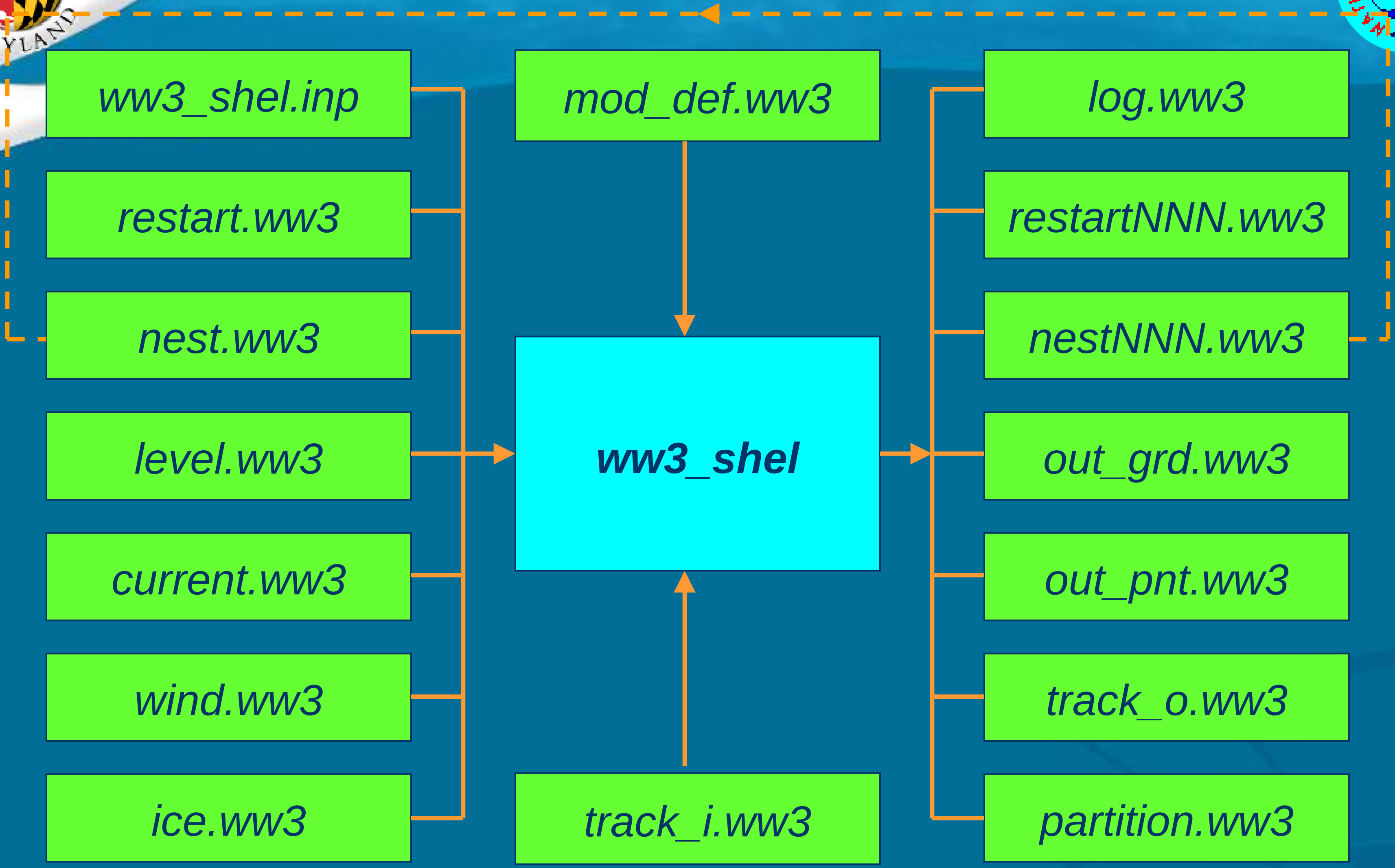
- Switch on boundary data output in ***ww3_shel.inp***, for the large scale model and run this model for a given time frame.
- Copy the file ***nest001.ww3*** as generated by the large scale model as ***nest.ww3*** to the work directory of the small scale model, and run the small scale model for (part of) the time frame.
- Check ***log.ww3*** of the small scale model to assure that new boundary data was read at the appropriate times

Nesting (ww3_shel)

Comments about nesting :

- The model matches up the locations in the file with the locations marked as input point in the grid. The order in which the data points are defined in either model therefore is immaterial.
- The locations of the in and output points may differ by a fraction of the grid size of the small scale grid.
- The data file contains on grid point spectra of the large scale model. The file size can thus be minimized by lining up the input points in the small scale model with the grid in the large scale model.
- Input point for the small scale grid have to coincide with sea point in the large scale grid.

Nesting (ww3_shel)



ww3_shel final remarks

Beware for the following when using example input files as templates. :

- Make sure time steps are set properly for your application to assure economy and stability. Considerations for the four time steps are:
  - Maximum time step for propagation for lowest frequency waves. CFL criterion needs to be satisfied for stable model computations.
    - Time steps for higher frequencies are automatically adjusted.
    - Currents are automatically accounted for.
  - Overall time step can be bigger, but not too much.
  - Refraction time step should be half overall time step or smaller.
  - Minimum source term time step can be very small.

ww3_shel final remarks

Pitfalls, cont’ed:

- The test output time step for restart files is set to 1s. If the range of output times is increased without changing this, the model will practically grind to a halt on I/O.

- In the example the garden sprinkler correction for option PR2 is switched of by setting the swell age to something very small. Make sure you switch this on be defining a realistic but stable 'swell age ‘
  - Better to use PR3 to begin with.

- Note that many compilers are not very graceful about running out of memory …..

The end

End of lecture