

# Set Up and Run WRF (Ideal and real data)

Wei Wang NCAR/MMM July 2017



#### WRF System Flowchart



#### Outline

- Running WRF code
  - Things to check before you run..
  - Running real-data case
  - Running idealized case
- Basic runtime options for a *single* domain run (*namelist*)
- Check output
- Simple trouble shooting
- Running a nested case: later



#### Before You Run ..

- Make sure appropriate executables are created in WRFV3/main/ directory:
  - ideal.exe executable to create idealized IC
  - real.exe executable to create IC/BC
  - wrf.exe executable for model integration
  - ndown.exe utility
  - tc.exe utility routine for TC bogusing
- If you are working with real data, be sure that files for <u>a few time periods</u> from WPS are correctly generated:
  - met\_em.d01.\*



#### WRF test case directories

You have these choices in WRFV3/test/ (made at compile time):



#### Steps to Run

- 1. cd to *run/* or one of the *test case* directories
- 2. Move or link WPS output files to the directory for real-data cases
- 3. Edit *namelist.input* file for the appropriate grid dimensions and times of the case
- 4. Run a initialization program (*ideal.exe* or *real.exe*)
- 5. Run model executable, wrf.exe



#### WRFV3/run directory

README.namelist LANDUSE. TBL GENPARM. TBL SOILPARM. TBL **VEGPARM**. TBL **URBPARM. TBL** RRTM DATA RRTMG SW DATA RRTMG LW DATA CAM ABS DATA CAM AEROPT DATA ozone.formatted ozone lat.formatted ozone plev.formatted aerosol.formatted aerosol lat.formatted aerosol lon.formatted aerosol plev.formatted gribmap.txt grib2map.tbl .... (a total of 60 files)

description of namelists

These are model physics data files: they are used to either initialize physics variables, or make physics computation faster \* Some of these files are text files, hence editable

for grib IO



#### WRFV3/run directory after compile

LANDUSE.TBL SOILPARM.TBL VEGPARM.TBL GENPARM.TBL URBPARM.TBL RRTM\_DATA RRTMG\_SW\_DATA RRTMG\_LW\_DATA ozone.formatted ozone\_lat.formatted ozone\_plev.formatted

An example after em\_real case compile

```
namelist.input - copied from ../test/em_real/namelist.input
real.exe -> ../main/real.exe
wrf.exe -> ../main/wrf.exe
ndown.exe -> ../main/ndown.exe
....(a few more)
```





Mesoscale & Microscale Meteorological Laboratory / NCAR

9

 If you have compiled the <u>em\_real</u> case, you should have:

real.exe - real data initialization program
wrf.exe - model executable
ndown.exe - program for doing one-way nesting
tc.exe - program for TC bogusing

These executables are linked to:

WRFV3/run

and

WRFV3/test/<u>em\_real</u>



 $\rightarrow$  One can go to either directory to run.

#### WRFV3/test/em\_real directory

LANDUSE.TBL -> ../../run/LANDUSE.TBL GENPARM.TBL -> ../../run/GENPARM.TBL SOILPARM.TBL -> ../../run/SOILPARM.TBL VEGPARM.TBL -> ../../run/VEGPARM.TBL URBPARM.TBL -> ../../run/URBPARM.TBL RRTM\_DATA -> ../../run/RRTM\_DATA RRTMG\_SW\_DATA -> ../../run/RRTMG\_SW\_DATA RRTMG\_LW\_DATA -> ../../run/RRTMG\_LW\_DATA ozone.formatted -> ../../run/ozone.formatted ozone\_lat.formatted -> ../../run/ozone\_lat.formatted

namelist.input - editing required
real.exe -> ../../main/real.exe
wrf.exe -> ../../main/wrf.exe
ndown.exe -> ../../main/ndown.exe
....(many more)



...

- One must successfully run WPS to prepare data required, and create met\_em.\* file for <u>multiple time period</u> for initial and boundary conditions
- Move or link WPS/metgrid output files to the run directory:

cd test/em\_real



ln -s ../../WPS/met\_em.d01.\*

- Edit namelist.input file for runtime options (at mininum, one must edit &time\_control for start, end and integration times, and &domains for grid dimensions)
- Run the real-data initialization program:
   ./real.exe if compiled serially / SMP, or

**mpirun** -np N ./real.exe for a MPI job where N is the number of processors requested.



 Successfully running real.exe will create model initial and boundary files:

wrfinput\_d01
wrfbdy\_d01
N-1 time-level data for
lateral boundaries, and
only for domain 1



N: the number of time periods processed

ncdump -v Times wrfbdy\_d01

• Typing 'ncdump -v Times wrfbdy\_d01' will give you, for a 24 hour period, 6 hourly data interval:

.. a bunch of prints and at the end:

data: Times = "2005-08-28\_00:00:00", "2005-08-28\_06:00:00", "2005-08-28\_12:00:00", "2005-08-28\_18:00:00";

\* BC data consists of values at the start of the time interval and rate of change in the time interval.



• Run the model executable by typing:

```
./wrf.exe >& wrf.out &
Or
mpirun -np N ./wrf.exe &
```

 Successfully running the model will a create model <u>history</u> file:

wrfout\_d01\_2005-08-28\_00:00:00

Based on start date set in namelist

and a <u>restart</u> file if **restart\_interval** is set to a time within the range of the forecast time:

wrfrst\_d01\_2005-08-28\_12:00:00



Exact time at a restart

#### wrfout d01 2005-08-28 00:00:00 Based on start date set in namelist ∠ 2008, 2008, 2008, 2008, start year 08, = 08,08, start month start day = 28, 28, 28, = 00, 00, 00,start hour = 00, 00, 00, start minute 00, **→** 00, / 00, start second = 2008, 2008, 2008,end year end month = 08, 08, 08,



end day

end hour

end minute

end second

Mesoscale & Microscale Meteorological Laboratory / NCAR	

= 00, 00,

= 00, 00,

= 29, 29, 29,

= 00, 00, 00,

00,

00,

17



Mesoscale & Microscale Meteorological Laboratory / NCAR

18

- An idealized case refers to data in the initial condition file (no need to run WPS)
- If you have compiled an ideal case, you should have:
   ideal.exe program to create idealized initial condition
   wrf.exe model executable
- These executables are linked to: WRFV3/run

and

WRFV3/test/em\_test-case



 $\rightarrow$  One can go to either directory to run.

Go to the desired *ideal* test case directory: e.g. cd test/em\_quarter\_ss

If there is 'run\_me\_first.csh' in the directory, run it first - this links relevant physics data files to the current directory:

./run\_me\_first.csh



Then run the ideal initialization program:

./ideal.exe

The input to this program is typically a sounding file (file named *input\_sounding*), or a pre-defined 2D input (e.g. *input\_jet* in **em\_b\_wave** case).

Running ideal.exe only creates WRF initial condition file: wrfinput\_d01



Note that wrfbdy file is not needed for idealized cases.

Instead, the boundary condition options are set in the **namelist.input** file. For example, these are for options in east-west, or x direction:

periodic_x	<pre>= .true.,.false.,.false.,</pre>
symmetric_xs	<pre>= .false.,.false.,.false.,</pre>
symmetric_xe	<pre>= .false.,.false.,.false.,</pre>
open_xs	<pre>= .false.,.false.,.false.,</pre>
open_xe	<pre>= .false.,.false.,.false.,</pre>



• To run the model interactively, type

./wrf.exe >& wrf.out &

for single processor (serial) or SMP run. Or

mpirun -np N ./wrf.exe &

for a MPI run (3D cases only)

 Successful running of the model executable will create a model history file called wrfout\_d01\_<date>

e.g. wrfout\_d01\_0001-01-01\_00:00:00

Based on start date set in namelist



#### wrfout\_d01\_0001-01-01 00:00:00 Based on start date set in namelist 6 0001 0001, 0001, start year = 01,01, 01, start month start day = 01, 01, 01,= 00, 00, 00,start hour = 00, /00, 00, start minute $\geq$ 00, / 00, 00, start second = 0001, 0001, 0001, end year end month = 01, 01, 01,= 01, 01, 01,end day = 00, 00, 00,end hour end minute = 120, 120, 120,= 00, 00, 00,end second



- Edit **namelist.input** file to change options.
- For your own case, you may provide a different sounding.
- You may also edit dyn\_em/ module\_initialize\_<case>. F to change other aspects of the initialization (more on Thur.)

#### Note:

- For 2D cases and baroclinic wave case, ideal.exe must be run serially
- For all 2D cases, wrf.exe must be run serially or with SMP



For the 1D case, compile and run serially

#### **Basic namelist Options**



# What is a namelist?

- A Fortran namelist contains a list of *runtime* options for the code to read in during its execution. Use of a namelist allows one to change runtime configuration without the need to recompile the source code.
- Fortran 90 namelist has very specific format, so edit with care:

```
&namelist-record - start
/ - end
```

- As a general rule:
  - Multiple columns: domain dependent
  - Single column: value valid for all domains



A namelist file may contain a number of records

#### namelist record &time control

run days run hours run minutes run seconds start year start month start day start hour start minute start second end year end month end day end hour end minute end second interval seconds history interval frames per outfile restart interval restart





#### Notes on &time\_control

- *run\_*\* time variables:
  - Model simulation length: wrf.exe and domain 1 only
- start\_\* and end\_\* time variables:
  - Program *real* will use WPS output between these times to produce lateral (and lower) boundary file
  - They can also be used to specify the start and end of simulation times for the coarse grid if *run\_\** variables are not set (or set to 0)



#### Notes on &time\_control

- *interval\_seconds*:
  - Time interval between WPS output times, and lateral BC (and lower BC) update frequency
- *history\_interval*:
  - Time interval in <u>minutes</u> when a history output is written (<u>note</u> output is instantaneous)
  - If the time\_step cannot be evenly divided by
     history\_interval, then nearest time step output is used
  - The time stamp in a history file name is the time when the history file is first written, and multiple time periods may be written in one file. e.g. a history file for domain 1 that is first written for 1200 UTC Jan 24 2000 is

wrfout\_d01\_2000-01-24\_12:00:00



#### Notes on &time\_control

- frames\_per\_outfile:
  - Number of history times written to one file
- restart\_interval:
  - Time interval in minutes when a restart file is written
  - By default, restart file is not written at hour 0
  - A restart file contains only one time level data, and its <u>valid time</u> is in its file name, e.g. a restart file for domain 1 valid for 0000 UTC Jan 25 2000 is

wrfrst\_d01\_2000-01-25\_00:00:00

• restart:



– whether this is a restart run

#### Notes on &time control

Example 1: all output times are in a single file

history\_interval = 180, 60, 60, frames\_per\_outfile = 1000, 1000, 1000, wrfout\_d01\_2000-01-24\_12:00:00

Example 2: each output file only contains a single time history\_interval = 180, 60, 60, frames\_per\_outfile = 1, 1, 1, wrfout\_d01\_2000-01-24\_12:00:00 wrfout\_d01\_2000-01-24\_15:00:00



#### Notes on *restart*

- What is a *restart* run?
  - A restart run is a continuation of a model run
- How to do a *restart* run:
  - In the first run, set <u>restart\_interval</u> to a value that is within the model integration time
  - A restart file will be created. e.g.

wrfrst\_d01\_2000-01-25\_00:00:00

- When doing a restart run:
  - Set restart = .true.,
  - Set start time to restart time



#### &time\_control

- io\_form\_history
- io\_form\_restart
- io form input
- io\_form\_boundary



For large files:

io\_form\_restart = 102 :
write output in patch
sizes: fast for large grid
and useful for restart file

IO format options: = 1, binary = 2, netcdf (most common) = 4, PHDF5 = 5, Grib 1 =10, Grib 2 =11, pnetCDF



#### namelist record &domains





#### Notes on &domains

- *time\_step, time\_step\_fract\_num, time\_step\_frac\_den*:
  - Time step for model integration in seconds
  - Fractional time step specified in separate integers of numerator and denominator
  - Typically 5 to 6xDX (DX is grid distance in km)
- e\_we, e\_sn, e\_vert:
  - Model grid dimensions (staggered) in X, Y and Z directions
- *num\_metgrid\_levels*:
  - Number of *metgrid* (input) data levels
- *num\_metgrid\_soil\_levels*:
  - Number of soil data levels in the input data

Found by typing ncdump -h met\_em.d01.<date> | more

- *dx, dy*:
  - grid distance: in meters


### Notes on &domains

- *p\_top\_requested*:
  - Pressure value at the model top
  - Constrained by the available data from WPS
  - Default is 5000 Pa (recommended as lowest Ptop)
- eta\_levels:
  - Specify your own model levels from 1.0 to 0.0
  - If not specified, program *real* will calculate a set of levels
  - Use a minimum of 30 levels with 5000 Pa model top to limit vertical grid distance < 1 km. Use more vertical levels when decreasing horizontal grid sizes.



#### namelist record <a>bdy\_control</a>

spec\_bdy\_width
spec\_zone
relax\_zone
specified
nested



May change relax\_zone
and spec\_bdy\_width
(spec\_zone + relax\_zone
= spec\_bdy\_width)

\* Wider boundary zone may work better for coarser driving data



## Where do I start?

- Always start with a *namelist* template provided in a test case directory, whether it is an ideal case, or a real data case.
  - A number of namelist templates are provided in test/test\_<case>/ directories

For example: in test/em\_real/, there are
 namelist.input.4km ~ 4 km grid size
 namelist.input.jun01 ~ 10 km grid size
 namelist.input.jan00 ~ 30 km grid size



### Where do I start?

- For different applications, please refer to p5-36 to 5-38 of the ARW User's Guide:
  - 2 or 4 km microphysics-only runs
  - 20 30 km, 2 3 day runs
  - Antarctic region
  - Tropical storm forecasting
  - Regional climate
  - Try physics suites (since V3.9)



## Where do I start?

- Use document to guide the modification of the namelist values:
  - run/README.namelist
  - test/em\_real/examples.namelist
  - User's Guide, Chapter 5 (online version has the latest)
  - Full list of namelists and their default values can be found in Registry files: Registry.EM\_COMMON, registry.io\_boilerplate (for IO options) and other registry files - look for character string 'namelist'



## To run a job in a different directory..

- Directories *run*/ and test\_<case>/ are convenient places to run, but it does not have to be.
- Copy or link the content of these directories to another directory, including physics data files, wrf input and boundary files, wrf namelist and executables, and you should be able to run a job anywhere on your system.



## **Check Output**



43

# Output After a Model Run

- Standard out/error files:
   wrf.out, or rs1.\* files
- Model history file(s):

wrfout\_d01\_<date>

Model restart file(s), optional
 wrfrst d01 <date>



### Output from a multi-processor run

The standard out and error will go to the following files for a MPI run: mpirun -np 4 ./wrf.exe →

rsl.out.0000	rsl.error.0000
rsl.out.0001	rsl.error.0001
rsl.out.0002	rsl.error.0002
rsl.out.0003	rsl.error.0003

There is one pair of files for each processor requested



## What to Look for in a standard out File?

Check run log file by typing

tail wrf.out, or

tail rsl.out.0000

You should see the following if the job is successfully completed:

wrf: SUCCESS COMPLETE WRF



# How to Check Model History File?

• Use ncdump:

ncdump -v Times wrfout\_d01\_<date>
to check output times. Or
 ncdump -v U wrfout\_d01\_<date>
to check a particular variable (U)

- Use **ncview** (great tool!)
- Use post-processing tools (see talks later)



# What is in a *wrf.out* or *rsl* file?

- Model version, decomposition info: Ntasks in X 2, ntasks in Y 4 WRF V3.9 MODEL
- Time taken to compute one model step:

 Timing for main: time 2000-01-24\_20:03:00 on domain
 1:
 0.89475 elapsed seconds

 Timing for main: time 2000-01-24\_20:06:00 on domain
 1:
 0.09011 elapsed seconds

 Timing for main: time 2000-01-24\_20:09:00 on domain
 1:
 0.08634 elapsed seconds

 Timing for main: time 2000-01-24\_20:12:00 on domain
 1:
 0.09004 elapsed seconds

• Time taken to write history and restart file:

Timing for Writing wrfout\_d01\_2000-01-25\_00:00:00 for domain 1: 0.07091 elapsed seconds

- Any model error prints:



 $\rightarrow$  An indication the model has become numerically unstable

# Simple Trouble Shooting



49

### Often-seen runtime problems

- module\_configure: initial\_config: error reading
   namelist: &dynamics
  - > Typos or erroneous namelist variables exist in namelist record &dynamics in namelist.input file
- input\_wrf.F: SIZE MISMATCH: namelist ide,jde,num\_metgrid\_levels= 70 61 27 ; input data ide,jde,num\_metgrid\_levels= 74 61 27
  - > Grid dimensions in error



### Often-seen runtime problems

- Segmentation fault (core dumped)

> Often typing 'unlimit' or 'ulimit -s
unlimited' or equivalent can help when this
happens quickly in a run, and on a small computer

- If you do: grep cfl rsl.error.\* and see 121 points exceeded cfl=2 in domain 1 at time 4.200000 MAX AT i,j,k: 123 48 3 cfl,w,d(eta)= 4.165821
  - > Model becomes unstable due to various reasons. If it happens soon after the start time, check input data, and/or reduce time step.



#### References

- Information on compiling and running WRF, and a more extensive list of namelist options and their definition / explanations can be found in the User's Guide, Chapter 5
- Also see nesting talk and demonstration tomorrow.

