Running the WRF Model

(for *real* and *Ideal* cases)

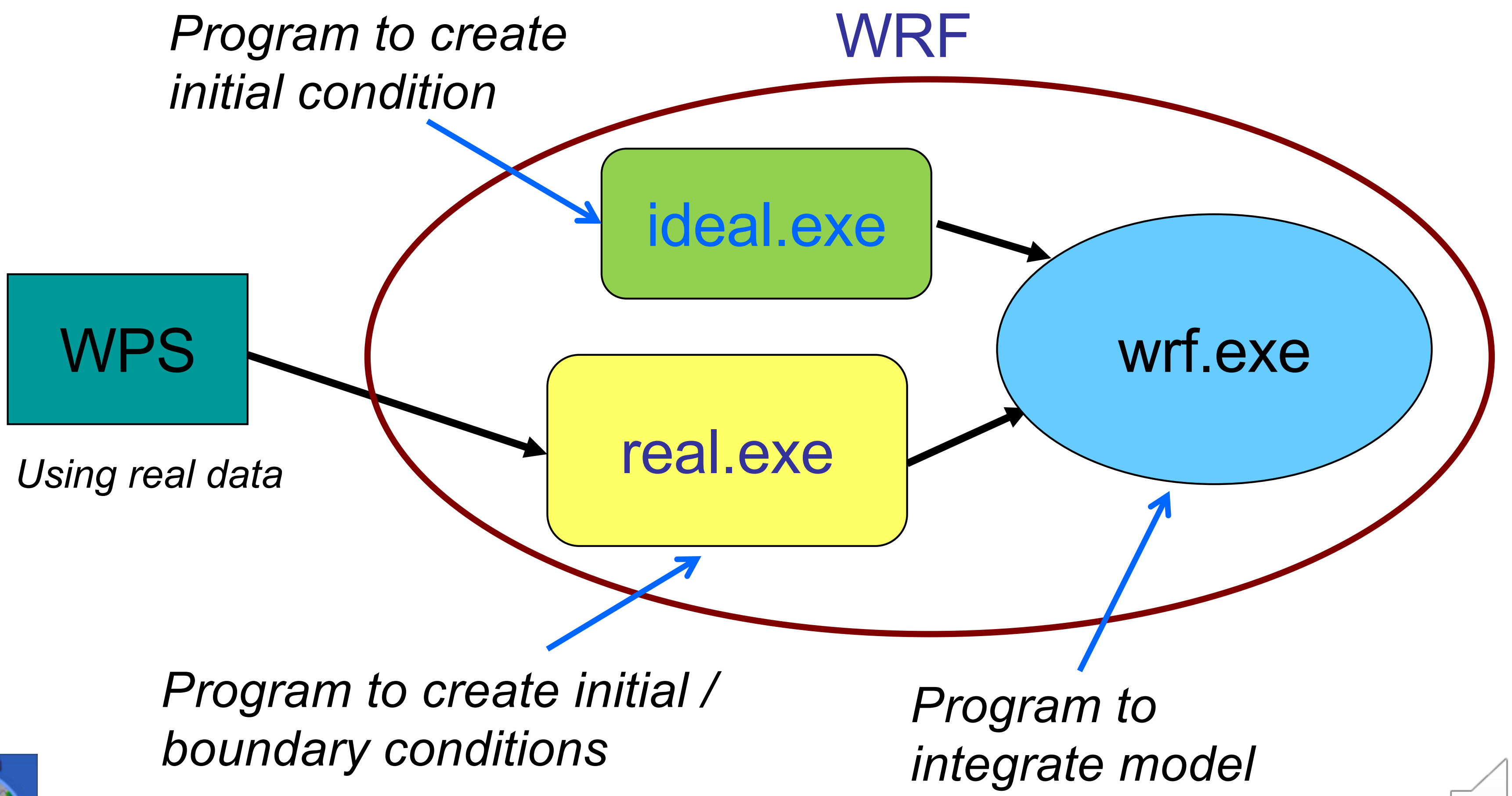
Wei Wang

January 2021

Mesoscale and Microscale Meteorology Laboratory, NCAR

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

WRF System Flowchart


Program to create initial condition
WRF
ideal.exe
WPS
wrf.exe
real.exe
Using real data
Program to create initial / boundary conditions
Program to integrate model

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 ..

- Top directory is now **WRF/**
- Make sure appropriate executables are created in **WRF/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 ***a few time periods*** from WPS are correctly generated:
  - `met_em.d01.*`

WRF test case/run directories

You have these choices in **`WRF/test/`**

(choices made at compile time. e.g. ***`compile em_real`***, and different compile creates different <u>initialization</u> program):

Idealized Cases			***Real-data***
1D	2D	3D	3D only
`em_scm_xy`	**`em_hill2d_x`**	**`em_quarter_ss`**	**`em_real`**
	`em_squall2d_x`	**`em_b_wave`**	
	`em_squall2d_y`	**`em_les`**	
	`em_grav2d_x`	**`em_tropical_cyclone`**	
	`em_seabreeze2d_x`	**`em_heldsuarez`**	

Steps to Run

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

WRF/run directory

```
README.namelist
```
} *description of namelists*

```
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
```
…. (a total of 60 files)

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*

WRF/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)

Running a Real-Data Case

WRF/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
ozone_plev.formatted -> ../../run/ozone_plev.formatted
…
namelist.input
real.exe -> ../../main/real.exe
wrf.exe -> ../../main/wrf.exe
```

→ runtime option file, editing required

…. (many more)

Running a Real-data Case

- One must successfully run WPS to prepare data required, and create **`met_em.*`** files for *multiple time periods* for initial and lateral boundary conditions

- Move (`mv`) or link (`ln -s`) WPS/metgrid output files to the current directory:

```
cd test/em_real
ln -s ../../../WPS/met_em.d01.* .
```

Running a Real-data Case

- 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:

  `mpirun -np N ./real.exe` for a MPI job

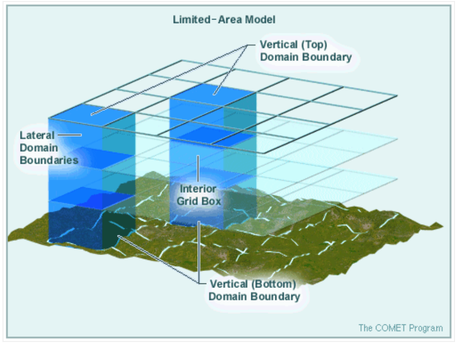
  where ***N*** is the number of processors requested.

Running a Real-data Case

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

`wrfinput_d01` ← *Single time level data at model's start time*

`wrfbdy_d01` ← *M-1 time-level data for lateral boundaries*


Limited-Area Model
Vertical (Top) Domain Boundary
Lateral Domain Boundaries
Interior Grid Box
Vertical (Bottom) Domain Boundary
The COMET Program

M: the number of time periods processed

```
ncdump -v Times wrfbdy_d01
```

Running a Real-data Case

- Typing '**`ncdump -v Times wrfbdy_d01`**' will give you these boundary times for a 24 hour period, 3 hourly data interval:

.. a bunch of prints and then at the end:

```
data:
 Times =
  "2005-08-28_00:00:00",
  "2005-08-28_03:00:00",
  "2005-08-28_06:00:00",
  "2005-08-28_09:00:00",
  "2005-08-28_12:00:00",
  "2005-08-28_15:00:00",
  "2005-08-28_18:00:00",
  "2005-08-28_21:00:00" ;
```

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

Running a Real-data Case

- Run the model executable by typing:

```
mpirun -np N ./wrf.exe &
```

- Successfully running the model will create one or more model *history* file:

$$\texttt{wrfout\_d01\_}\underbrace{\texttt{2005-08-28\_00:00:00}}_{\textit{Based on start date set in namelist}}$$

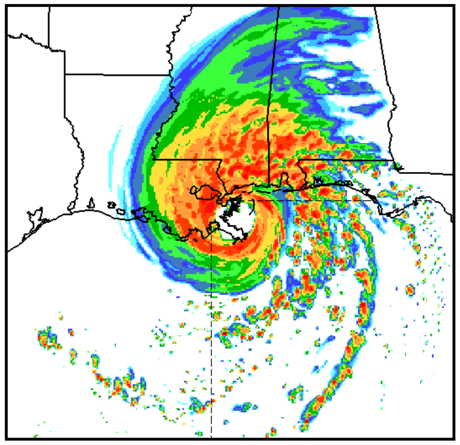
  and a *restart* file if **`restart_interval`** is set to a time within the range of the forecast time:

$$\texttt{wrfrst\_d01\_}\underbrace{\texttt{2005-08-28\_12:00:00}}_{\textit{Exact time at a restart}}$$

Running a Real Data Case

$$\texttt{wrfout\_d01\_}\underbrace{\texttt{2005-08-28\_00:00:00}}_{\textit{Based on start date set in namelist}}$$

```
start_year                     = 2008, 2008, 2008,
start_month                    = 08,   08,   08,
start_day                      = 28,   28,   28,
start_hour                     = 00,   00,   00,
start_minute                   = 00,   00,   00,
start_second                   = 00,   00,   00,
end_year                       = 2008, 2008, 2008,
end_month                      = 08,   08,   08,
end_day                        = 29,   29,   29,
end_hour                       = 00,   00,   00,
end_minute                     = 00,   00,   00,
end_second                     = 00,   00,   00,
restart_interval               = 720,
```



Running an Idealized Case

Running an *Idealized* Case

- 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:

  `WRF/run`

  and

  `WRF/test/`*`em_test-case`*

➔ One can use either directory to run.

Running an *Idealized* Case

Go to the desired *ideal* test case directory: e.g.

```
cd test/em_quarter_ss
```

You should see these files:

```
README.quarter_ss
input_sounding
namelist.input
run_me_first.csh
```

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
```

Running an *Idealized* Case

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**

Running an *Idealized* Case

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    = .false.,
symmetric_xs  = .false.,
symmetric_xe  = .false.,
open_xs       = .true.,
open_xe       = .true.,
```

Running an *Idealized* Case

- To run the model interactively, type

  `./wrf.exe`

  to use a single processor. 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_$\underbrace{\texttt{0001-01-01_00:00:00}}_{\textit{Based on start date set in namelist}}$

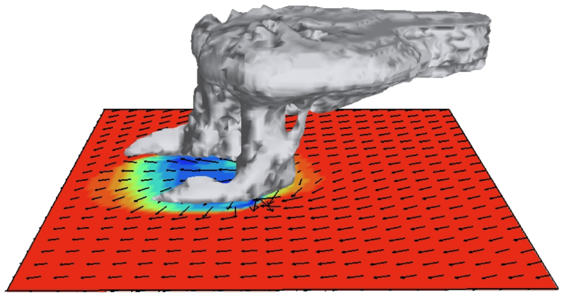
  (dates are important for radiation physics)

Running an *Idealized* Case

$\underbrace{\texttt{wrfout\_d01\_}\texttt{0001-01-01\_00:00:00}}$

Based on start date set in namelist

```
start_year       = 0001, 0001, 0001,
start_month      = 01,   01,   01,
start_day        = 01,   01,   01,
start_hour       = 00,   00,   00,
start_minute     = 00,   00,   00,
start_second     = 00,   00,   00,
end_year         = 0001, 0001, 0001,
end_month        = 01,   01,   01,
end_day          = 01,   01,   01,
end_hour         = 00,   00,   00,
end_minute       = 120,  120,  120,
end_second       = 00,   00,   00,
```



Basic Runtime Options

What is a runtime option?

- A *runtime* option is an option that can be read in at the model execution time. Use of a runtime option allows a user to change model configuration without the need to recompile the model source code.
- Runtime options are employed in the model using Fortran 90 namelist construct, and placed in a file named *namelist.input*:

```
&namelist-name    - start
/                 - end
```

- A runtime option can have a single, or an array of values, and they can integer, real, or logical
  - Multiple columns: domain dependent
  - Single column: value valid for all domains
  - The order of variables in a namelist does not matter
- There are multiple namelists in the *namelist.input* file.

What are in namelist.input?

- A typical **`namelist.input`** file for WRF model has these namelist records:

```
&time_control
&domains
&physics
&dynamics
&bdy_control
&namelist_quilt
```

$\left.\begin{array}{l}\texttt{\&time\_control}\\ \texttt{\&domains}\end{array}\right\}$ *essential*

$\left.\begin{array}{l}\texttt{\&physics}\\ \texttt{\&dynamics}\\ \texttt{\&bdy\_control}\end{array}\right\}$ *Interesting to browse*

`&namelist_quilt` (*ignore for the time being*)

namelist record &time_control

```
run_days                 = 0,
run_hours                = 24,
run_minutes              = 0,
run_seconds              = 0,
start_year               = 2005, 2000, 2000,
start_month              = 08,   01,   01,
start_day                = 28,   24,   24,
start_hour               = 00,   12,   12,
start_minute             = 00,   00,   00,
start_second             = 00,   00,   00,
end_year                 = 2005, 2000, 2000,
end_month                = 08,   01,   01,
end_day                  = 29,   25,   25,
end_hour                 = 00,   12,   12,
end_minute               = 00,   00,   00,
end_second               = 00,   00,   00,
interval_seconds         = 10800
history_interval         = 180,  60,   60,
frames_per_outfile       = 1000, 1000, 1000,
restart_interval         = 360,
restart                  = .true.,
```

for nests

namelist record &time_control

```
run_days                = 0,
run_hours               = 24,
run_minutes             = 0,
run_seconds             = 0,
start_year              = 2005, 2000, 2000,
start_month             = 08,   01,   01,
start_day               = 28,   24,   24,
start_hour              = 00,   12,   12,
start_minute            = 00,   00,   00,
start_second            = 00,   00,   00,
end_year                = 2005, 2000, 2000,
end_month               = 08,   01,   01,
end_day                 = 29,   25,   25,
end_hour                = 00,   12,   12,
end_minute              = 00,   00,   00,
end_second              = 00,   00,   00,
interval_seconds        = 10800
…
```

Model simulation length, domain 1 and **wrf.exe** only

Start and end of simulation times, used by both **real.exe** and **wrf.exe.** For the model, run_* times override end_* times

Time interval between WPS data times

namelist record &time_control

```
…
history_interval      = 180,   60,    60,
frames_per_outfile    = 1000, 1000, 1000,
…
```

Model output data interval in minutes

How many time periods of model output in a single file

```
restart_interval      = 360,
restart               = .true.,
```

Model restart time interval in minutes

Model restart time interval in minutes

Notes on `&time_control`

- **`history_interval`** and history file:
  - If the **`time_step`** variable in **`&domains`** 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 open for writing, and multiple time periods may be written in one file. e.g. a history file for domain 1 that is first written for 0000 UTC Aug 28 2005 is

    `wrfout_d01_2005-08-28_00:00:00`
  - The history output is *instantaneous,* or a *snapshot* of the model atmosphere at the output time*.*

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_2005-08-28_00:00:00
```

Example 2: each output file only contains a single time

```
history_interval   = 180, 60,   60,
frames_per_outfile = 1,    1,    1,
  wrfout_d01_2005-08-28_00:00:00
  wrfout_d01_2005-08-28_03:00:00
  wrfout_d01_2005-08-28_06:00:00
```

Notes on `&time_control`

- `restart_interval`:
  - The time unit for this variable is minutes;
  - By default, restart file is not written at hour 0.
- restart file: `wrfrst_*`
  - A restart file contains only one time level data, and its valid time is in its file name, e.g. a restart file for domain 1 valid for 1200 UTC Aug 28 2005 is

    `wrfrst_d01_2005-08-28_12:00:00`
  - A restart file size is much larger than the size of a single-time history file;
  - The only purpose of `wrfrst` file is to restart the model.

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 *restart_interval* to a value that is within the model integration time
  - A restart file will be created. e.g.

    `wrfrst_d01_2005-08-28_12:00:00`
- When doing a restart run:
  - Set *restart* = .true.,
  - Set start time to restart time
  - Set run_* to be the hours remaining in the run

&time_control

```
io_form_history      = 2,
io_form_restart      = 2,
io_form_input        = 2,
io_form_boundary     = 2,
```

IO format options:

= 1, binary
= 2, netCDF (most common)
= 4, PHDF5
= 5, Grib 1
=10, Grib 2
=11, pnetCDF

For large files:

`io_form_restart = 102` :
write output in patch sizes: fast for large grid and useful for restart file

namelist record &domains

```
time_step                  = 180
time_step_fract_num        = 0,
time_step_fract_den        = 1,
max_dom                    = 1,
e_we                       = 174,
e_sn                       = 151
e_vert                     = 40,
num_metgrid_levels         = 32,
num_metgrid_soil_levels    = 4,
dx                         = 30000,
dy                         = 30000,
eta_levels                 = 1.0,0.996,0.99,0.98,… 0.0
p_top_requested            = 5000,
```

namelist record **&domains**

```
max_dom                  = 1,

e_we                     = 174,
e_sn                     = 151,
e_vert                   = 40,
```

How many domains in this run

Model domain dimensions in west-east (x) and south-north (y); need to match what are defined in **geogrid**

Defined when running **real**

```
num_metgrid_levels       = 32,
num_metgrid_soil_levels  = 4,
```

Input data dimensions: number of atmospheric data levels and soil levels from **metgrid**
(ncdump -h met_em.d01*)

namelist record &domains

```
time_step                 = 180  (in seconds)
time_step_fract_num       = 0,
time_step_fract_den       = 1,
```

Time stepping for model integration: 4-6*DX (in km)

Used to specify fractional time step

```
dx           = 30000,
dy           = 30000,
```

Grid distances in x, y; must match those defined in **geogrid**

```
p_top_requested           = 5000,
```

Model top pressure

namelist record `&domains`

```
eta_levels   = 1.0,0.996,0.99,0.98,… 0.0
```

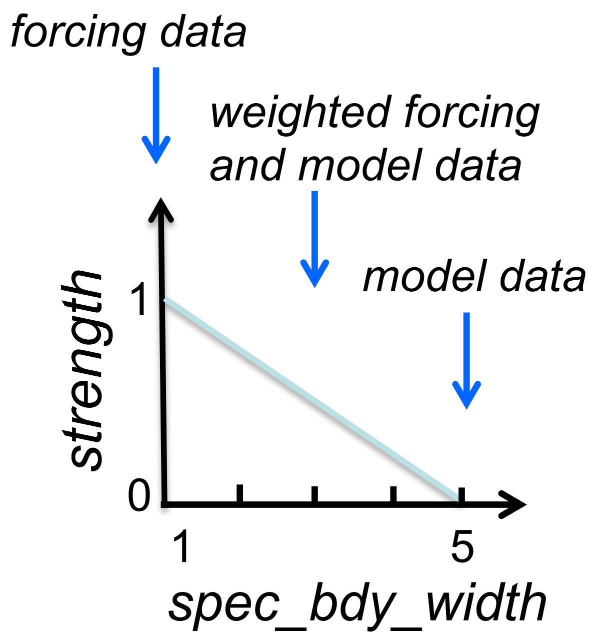
- Define the model levels by yourself
- The values must start with 1. and end with 0.
- The number of levels must match vertical dimension of the model: `e_vert`.
- Optional. If not used, the program `real` will compute a set of levels for you.
- Use adequate number of vertical levels; use more levels with higher horizontal model resolution

namelist record &bdy_control

Lateral boundary width

```
spec_bdy_width                 = 5,
specified                      = .true.,
```

Type of boundary conditions: **true** for real-data runs



May change ***spec_bdy_width*** to a larger value (e.g. 10)

* Wider boundary zone may work better for coarser driving data

Other namelists

&physics:

- Model physics options

&dynamics:

- Damping, diffusion options
- Advection options
- In 4.0, the hybrid vertical coordinate option is the default. Turn it off by setting the following for *real* and *wrf*:

```
hybrid_opt            = 0
```

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 “Examples of namelists for various applications” in the Chapter 5 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)

https://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/v4.2/contents.html

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

Output After a Model Run

- Standard out/error files:

  `rsl.out.* and rsl.error.*` files for a MPI run
- Model history file(s):

  `wrfout_d01_2005-08-28_12:00:00`
- Model restart file(s), maybe

  `wrfrst_d01_2005-08-28_12:00:00`

Output from a MPI 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. The `*.0000` files have the most info.

What to look for in a standard out file?

Check run log file by typing

```
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?

- List the files, and they should have reasonable size

```
ls -ls wrfout*
```

- 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!) for every program output

```
ncview wrfout_d01_*
```

- Use post-processing tools (see post-processing talks)

What is in a *rsl* file?

- Model version, decomposition info:

```
 Ntasks in X            2, ntasks in Y            4
  WRF V4.0 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:

```
5 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
```

→ An indication the model has become numerically unstable

What is in a *wrfout_d01_<date>* File?

- A few 1D fields, e.g.
  `Times, ZNW, ZNU,`
- Many 2D fields, for example:
  `T2, Q2, PSFC, MU, U10, V10, RAINC, RAINNC, SWDOWN, OLR, etc.`
- Fewer 3D fields, for example:
  `U, V, W, T, P, PB, PH, PHB, QVAPOR, QCLOUD, QICE, QRAIN, QSNOW, etc.`
- Use **`ncdump`** to get a list of fields,

```
ncdump -h wrfout_d01_<date> > list
```

- Model output fields are generally instantaneous.
- Output file size depends on model options used

Simple Trouble Shooting

Often-seen runtime errors

```
  ------ ERROR while reading namelist dynamics ------
Maybe here?:  scalar_adv_opt          = 1,      1,      1,
Maybe here?:  gwd_option              = 0,
```

> Typos or erroneous namelist variables exist in namelist record *&dynamics*

```
input_wrf.F:SIZE MISMATCH:namelist e_we                       = 70
input_wrf.F:SIZE MISMATCH:input file WEST-EAST_GRID_DIMENSION = 74
```

> Grid dimension **`e_we`** is wrong when compared to input data dimension

Often-seen runtime problems

– **Segmentation fault** which happens at the beginning of a model run:
  - > This usually means there isn't enough memory to run
  - > It can happen when using a small computer. Often typing '**unlimit**' or '**ulimit -s unlimited**' may help.
  - > On a large computer, this can happen if not enough processors are used.

– If you do: **grep cfl rsl.error.*** and see

```
27  points exceeded cfl=2 in domain d01 at time 2001-07-
   07_01:30:00 hours
MAX AT i,j,k: 116 49  7  vert_cfl,w,d(eta)=   4.703753  -
0.1980351  3.1363964E-03
```

  - > Model becomes unstable due to various reasons. The first thing to try is to reduce mode time step (**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

- This talk should be helpful for the first exercise case you will be doing on the first day of the tutorial.