



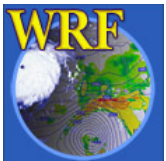
WRF Nesting: Set Up and Run

Wei Wang
NCAR/NESL/MMM
July 2014



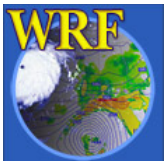
Outline

- General comments
- Nest namelist options
- Running WRF with nests
 - two-way nesting
 - moving nest
 - one-way nesting
- Summary



Before You Run ..

- Make sure you have selected **basic nest** compile options and appropriate executables are created in **WRFV3/main/** directory:
 - **real.exe**
 - **wrf.exe**
 - **ndown.exe**
 - **tc.exe**
- If you are running a real-data case, be sure that files for *nest* domains from WPS are generated:
 - **met_em.d01.<date>**, **met_em.d0*<date>**
(* terrain, static land data are in the nested files)



Steps to Run (same as before)

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 and times of the case
4. Run initialization program (*real.exe*)
5. Run model executable, *wrf.exe*



All in the namelist...

- Nearly all controls for a nested run can be achieved by editing the namelist file.
- Nest-specific namelist options will be explained next.

Important to note:

- Key variable: **max_dom** must be set to ≥ 2
- Need to pay attention to multi-column namelists



Nest namelist Options



&time_control

```
run_days      = 0,  
run_hours     = 24,  
run_minutes   = 0,  
run_seconds   = 0,  
start_year    = 2000, 2000, 2000,  
start_month   = 01, 01, 01,  
start_day     = 24, 24, 24,  
start_hour    = 12, 12, 12,  
start_minute  = 00, 00, 00,  
start_second  = 00, 00, 00,  
end_year      = 2000, 2000, 2000,  
end_month     = 01, 01, 01,  
end_day       = 25, 25, 25,  
end_hour      = 12, 12, 12,  
end_minute    = 00, 00, 00,  
end_second    = 00, 00, 00,  
interval_seconds = 21600
```

First column: domain 1 option

These control the start and end times of the nests. They can be different from the parent domain, but must fit in the time window of the parent domain



&time_control

```
interval_seconds      = 21600  
history_interval      = 180, 60, 60,  
frame_per_outfile     = 1000, 1000, 1000,  
restart_interval      = 360,
```

History output may
be split into multiple
files

- History files are written one for each domain
- History intervals may be different for different domains
- restart files are also written one per domain



&time_control

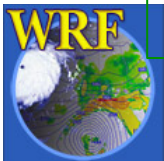
Nest input option:

```
input_from_file = .true., .true., .true.,  
fine_input_stream = 0, 2, 2,
```

Specify what fields to use in nest input: they can be all (0), or data specified in I/O stream 2 in Registry (2).

Limited use: if a nest starts at a later time, or have an updated analysis only on domain 1.

Whether to produce in *real.exe* and use nest wrfinput files in *wrf.exe*. This is usually the case for real-data runs. For idealized nest runs, set it to *.false.*



```
state real ht ij misc 1 - i012rhds "HGT" "Terrain Height" "m"
```

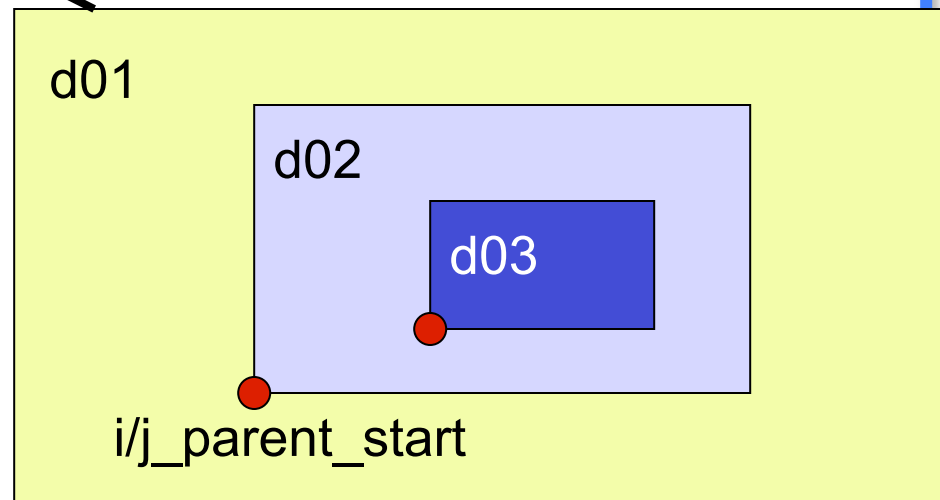
&domains

```
max_dom = 3,  
e_we    = 74, 112, 94,  
e_sn    = 61, 97, 91,  
e_vert  = 28, 28, 28,  
grid_id      = 1, 2, 3,  
parent_id    = 0, 1, 2,  
i_parent_start = 0, 31, 30,  
j_parent_start = 0, 17, 30,
```

Activate nests: no. of domains to run

Dimensions of all domains; same as in WPS.

Make sure the nest domain parameters match those defined in WPS



&domains

```
dx = 30000., 10000., 3333.33,  
dy = 30000., 10000., 3333.33,  
parent_grid_ratio = 1, 3, 3,  
parent_time_step_ratio = 1,3,3,
```

For fractional grid distance,
use at least 2 decimal places

All 4 variables must be specified. *Grid ratio* can be any integer (3 and 5 are recommended), and *time step ratio* can be different from grid ratio. Grid distance is in meters, even for lat/lon map projection.



&domains

```
feedback      = 1,  
smooth_option = 2,
```

When feedback is on, this option can be selected to smooth the area in the parent domain where the nest is. Valid values are 0,1,2.

Whether a nest will overwrite parent domain results. Setting feedback=0 → 'one-way' nesting in a concurrent run.



&bdy_control

```
spec_bdy_width = 5,  
spec_zone      = 1, (ARW only)  
relax_zone     = 4, (ARW only)  
specified      = .T., .F., .F.,  
nested        = .F., .T., .T.,
```

Boundary condition
option for domain 1.

Boundary condition
option for nests.

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



Other notes on namelists

- Use same physics options for all domains.
 - An exception is cumulus scheme. One may need to turn it off for a nest that has grid distance of a few kilometers.
- Also use same physics calling frequency (e.g. **radt**, **cudt**, etc.) in all domains.

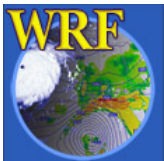


Where do I start?

- Always start with a *namelist* template provided in a test case directory, whether it is a ideal case, or real data case.
- Not all namelists are function of domains. If in doubt, check [Registry.EM_COMMON](#) and [registry.io_boilerplate](#) (look for string ‘[namelist](#)’).
- Use document to guide the modification of the namelist values:
 - run/README.namelist
 - User’s Guide, Chapter 5



Running Nested Case



Running a Nested Case

- Files available from WPS:

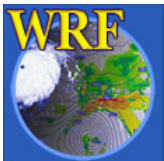
`met_em.d01.<date>` (a few time periods)

`met_em.d02.<date>` (at least one time period data)

- Move or link WPS output files to the run directory:

```
cd test/em_real
```

```
ln -s ../ ../WPS/met_em.*
```



Running a Nested Case

- Edit `namelist.input` file for runtime options (set `max_dom >= 2` in `&domains` for a nested run)
- 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



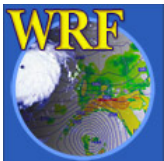
Running a Nested Case

- Successfully running this program will create model initial and boundary files:

wrfinput_d01
wrfinput_d02
wrfbdy_d01

*Single time level
data at model's
start time for all
domains*

*Multiple time-level data
at the lateral boundary,
for domain 1*



Running a Nested Case

- Run the model executable by typing:

```
./wrf.exe >& wrf.out &
```

or

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

- Successfully running the model will create model *history* files, one for each domain:

```
wrfout_d01_2005-08-28_00:00:00
```

```
wrfout_d02_2005-08-28_00:00:00
```

And *restart* file if `restart_interval` is smaller than the integration time:

```
wrfirst_d01_<date>, wrfirst_d02_<date>
```



Moving Nest Case

- The main reason for using this option is to run the model economically.
- Must choose correct compile options when creating **configure.wrf** file
 - Choose [preset move](#), or [vortex following](#)
- Other options are controlled by the namelists.
- Can do specified move, and automatic vortex tracking (for tropical cyclone application).
- All nest domains can move, but driven by the innermost nest



Specified Moving Case

- namelists in **&domains**:

`num_moves, move_id, move_interval,
move_cd_x, move_cd_y`

→ nest can only move one parent-grid-cell at a time.

i.e., `move_cd_x = 1, -1, or 0`

- Also specify initial nest location:

`i_parent_start, j_parent_start`



Automatic Moving Case

- Tropical cyclone applications only.
- Works better for well developed storms.
- Namelists in **&domains**:
 - `vortex_interval` (default 15 min)
 - `max_vortex_speed` (default 40 m/s)
 - `corral_dist` (default 8 coarse grid cells)
 - `track_level` (default 50000 Pa)
 - `time_to_move` (default is 0 h for all nests)
- Also specify initial nest location
 - `i_parent_start, j_parent_start`



One-way Nesting: Two separate runs

Less common option:

- Prepare data as if one were to run a two-way nested case up to program real;
- Run WRF model for coarsest domain first. Should output model frequently (e.g. hourly);
- Use program **ndown.exe**, together with coarsest domain model output and nest domain wrfinput file, to generate wrfinput and wrfbdy file for the next model run;
- Run WRF model for the second domain.

(Also see Chapter 5, pages 15 - 17)



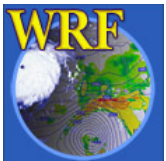
Summary

- Two-way, without nest input files (`input_from_file=.f.`)
- Two-way, with nest input files (`input_from_file=.t.`)
- Two-way, with static nest input only (`input_from_file=.t., fine_input_stream=2`)
- One-way, *concurrent* run (`feedback=0`)
- One-way, *separate* runs (treated like two single-domain runs, with *ndown*)
- Two-way, specified moving nest run
- Two-way, automatic vortex tracking run



Notes about Nesting

- When should I use nests?
 - Input data resolution is too coarse (for example, some reanalysis data: NNRP, NCEP2, climate model data)
 - Would like to simulate localized convection, topography- and/or landuse-forced phenomena, etc.
 - Would like to provide better boundary conditions for the area of interest: boundary conditions from external sources are typically 3 – 6 hourly, while nested boundary conditions are in minutes (coarse domain time step)
 - There isn't sufficient computing resources
- Nest domain sizes should not be too small
 - No less than 100x100
 - Avoid boundary zones that are about 10 grid point wide
 - Avoid 'sweeping' effect from lateral boundaries
 - Avoid placing nest boundaries over high mountains



References

- Information on compiling and running WRF with nests, and a more extensive list of namelist options and their definition / explanation can be found in the [User's Guide, Chapter 5](#)
- Start with namelist templates in [test/](#) directory, and refer to namelist used for different applications on pages 5-28 – 30 in the User's Guide



Practice with online tutorial, and in the class.