Chapter 4: WRF Initialization

Table of Contents

Introduction

The WRF model has two classes of simulations it can generate: those with an *ideal* initialization and those utilizing *real* data. Idealized simulations typically manufacture an initial condition file for the WRF model from an existing 1-D or 2-D sounding and assume a simplified analytic orography. Real-data cases usually require pre-processing from the WPS package, which provides each atmospheric and static field with fidelity appropriate to the chosen grid resolution for the model. The WRF model executable itself is not altered by choosing one initialization option over another (idealized *vs*. real), but the WRF model pre-processors (the real.exe and ideal.exe programs) are specifically built based upon a user's selection. Either real.exe or ideal.exe will be run prior to running the WRF model.

Ideal *vs.* real cases are divided as follows:

- Ideal cases – initialization programs named "ideal.exe"
    - 3d
        - em_b_wave - baroclinic wave, 100 km
        - em_convrad – convective radiation, 1 km
        - em_fire – surface fire, 50 m
        - em_heldsuarez – global case with polar filtering, 625 km
        - em_les – large eddy simulation, 100 m
        - em_quarter_ss - super cell, 2 km
        - em_tropical_cyclone – hurricane, 15 km
    - 2d
        - em_grav2d_x – gravity current, 100 m
        - em_hill2d_x – flow over a hill, 2 km
        - em_seabreeze2d_x – water and land, 2 km, full physics
        - em_squall2d_x – squall line, 250 m
        - em_squall2d_y – transpose of above problem

- 1d
  - em_scm_xy – single column model, 4 km, full physics

- Real data cases – initialization program named “real.exe”
  - em_real – examples from 4 to 30 km, full physics

Selection of the type of forecast is made when issuing the **./compile** statement. When selecting a different case to study, the code must be re-compiled to choose the correct initialization for the model. For example, after configuring the setup for the **architecture** (with the **./configure** command), if users issues the command **./compile em_real**, then the initialization program is built using **module_initialize_real.F** as the target module (one of the **./WRF/dyn_em/module_initialize_*.F** files).

For ideal initializations, there is a combination of files that may used to build the executable. For the **em_fire**, **em_heldsuarez**, **em_scm_xy**, and **em_tropical_cyclone** cases, a separate initialization file exists (e.g., module_initialize_fire.F for the em_fire case). For the remaining idealized cases, the file **./WRF/dyn_em/module_initialize_ideal.F** is used. Note the WRF forecast model is identical for both of these initialization programs. In each of these initialization modules, the same type of activities occur:

- compute a base state / reference profile for geopotential and column pressure
- compute the perturbations from the base state for geopotential and column pressure
- initialize meteorological variables: u, v, potential temperature, vapor mixing ratio
- define a vertical coordinate
- interpolate data to the model’s vertical coordinate
- initialize static fields for the map projection and the physical surface; for many of the idealized cases, these are simplified initializations, such as map factors set to one, and topography elevation set to zero
- just prior to exiting the routine that handles these pre-processing activities, if users have requested that the thermal field use moist potential temperature, that diagnostic variable is computed

Both the real.exe program and ideal.exe programs share a large portion of source code to handle the following duties:

- read data from the namelist
- allocate space for the requested domain, with model variables specified at run-time
- generate initial condition file

The real-data case does some additional processing:

- read meteorological and static input data from the WRF Preprocessing System (WPS)
- prepare soil fields for use in the model (usually, vertical interpolation to the required levels for the specified land surface scheme)
- check to verify that soil categories, land use, land mask, soil temperature, sea surface temperature are all consistent with each other
- multiple input time periods are processed to generate the lateral boundary conditions, which are required unless processing a global forecast
- 3d boundary data (u, v, potential temperature, vapor mixing ratio, total geopotential) are coupled with total column pressure

The “real.exe” program may be run as either a serial or a distributed memory job. Since the idealized cases only require that the initialization run for a single time period (no lateral boundary file is required) and are, therefore, quick to process, all of the “ideal.exe” programs should be run on a single processor. The Makefile for 2-D cases will not allow users to build the code with distributed memory parallelism. For large 2-D cases, if users requires OpenMP, the variables **`numtiles_x`** and **`numtiles_y`** must be set in the **`domains`** portion of the namelist file **`namelist.input`** (**`numtiles_y`** must be set to 1, and **`numtile_x`** then set to the number of OpenMP threads).

Initialization for Ideal Cases

"ideal.exe" is the program in the WRF system that allows users to run a controlled scenario. Typically this program requires no input except for the **`namelist.input`** and the **`input_sounding`** files. There are exceptions, for example the baroclinic wave case uses a 2-D binary sounding file. The program outputs the **`wrfinput_d01`** file that is read by the WRF model executable ("wrf.exe"). Since no external data is required to run idealized cases, even for researchers interested in real-data cases, idealized simulations are an easy way to ensure the model is working correctly on a particular architecture and compiler.

Idealized runs can use any of the boundary conditions except "**`specified`**" and are not, in general, set up to run with sophisticated physics. Most have no radiation, surface fluxes or frictional effects (other than the sea breeze case, LES, and the global Held-Suarez). Idealized cases are mostly useful for dynamical studies, reproducing converged or otherwise known solutions, and idealized cloud modeling. Again, there are exceptions. The tropical cyclone case lacks only radiation schemes, and the sea breeze case has a full complement of parameterization options.

There are 1-D, 2-D and 3-D examples of idealized cases, with and without topography, and with and without an initial thermal perturbation. The namelist controls the size of the domain, number of vertical levels, model top height, grid size, time step, diffusion and damping properties, boundary conditions, and physics options. A large number of settings already exist in the default namelists found in each case directory.

The **`input_sounding`** file (already in appropriate case directories) can be any set of levels that goes at least up to the model top height (**`ztop`**) in the namelist. The first line includes surface pressure (hPa), potential temperature (K) and moisture mixing ratio (g/kg). Each subsequent line has five input values: height (meters above sea-level), dry potential temperature (K), vapor mixing ratio (g/kg), x-direction wind component (m/s), and y-direction wind component (m/s). The "ideal.exe" program interpolates data from the **`input_sounding`** file and will extrapolate if enough data is not provided.

The base state sounding for idealized cases is the initial sounding, minus moisture, and therefore does not have to be defined separately. Note for the baroclinic wave case: a 1-D input sounding is not used because the initial 3-D arrays are read-in from the file **`input_jet`**. This means for the baroclinic wave case, the **`namelist.input`** file cannot be used to change the horizontal or vertical dimensions since they are specified in the **`input_jet`** file.

Making modifications, apart from namelist-controlled options or soundings, must be done by editing the Fortran code. Such modifications would include changing the topography, distribution of vertical levels, properties of an initialization thermal bubble, or preparing a case to use more physics, such as a land-surface model. The Fortran code to edit is contained in **`./WRF/dyn_em/module_initialize_[`*`case`*`].F`**, where **`[`*`case`*`]`** is the case chosen during compilation (e.g., **`module_initialize_fire.F`**. or **`module_initialize_ideal.F).`** The subroutine to modify is **`init_domain_rk`**. To change the vertical levels, only the 1-D array **`znw`** must be defined, containing the full levels, starting from 1 at k=1, and ending with 0 at k=kde. To change the topography, only the 2-D array **`ht`** must be defined, making sure it is periodic if those boundary conditions are used. To change the thermal perturbation bubble, search for the string "bubble" to locate the code to modify.

Each ideal case provides an excellent set of default examples for users. The method to specify a thermal bubble is given in the super cell case. In the hill2d case, the topography is accounted for properly in setting up the initial 3-D arrays, so that example should be followed for any topography cases. A symmetry example in the squall line cases tests that your indexing modifications are correct. Full physics options are demonstrated in the seabreeze2d_x case.

Available Ideal Test Cases

The available test cases are

1. 2-D squall2d_x (test/em_squall2d_x)
    - 2D squall line (x,z) using Kessler microphysics and a fixed 300 m^2/s viscosity.
    - periodicity condition used in y so that 3D model produces 2D simulation.
    - v velocity should be zero and there should be no variation in y in the results.
2. 2-D squall2d_y (test/em_squall2d_y)
    - Same as squall2d_x, except with (x) rotated to (y).
    - u velocity should be zero and there should be no variation in x in the results.
3. 3-D quarter-circle shear supercell simulation (test/em_quarter_ss).
    - Left and right moving supercells are produced.
    - See the README.quarter_ss file in the test directory for more information.
4. 2-D flow over a bell-shaped hill (x,z) (test/em_hill2d_x)
    - 10 km half-width, 2 km grid-length, 100 m high hill, 10 m/s flow, N=0.01/s, 30 km high domain, 80 levels, open radiative boundaries, absorbing upper boundary.
    - Case is in linear hydrostatic regime, so vertical tilted waves with ~6-km vertical wavelength.
5. 3-D baroclinic waves (test/em_b_wave)
    - Baroclinically unstable jet u(y,z) on an f-plane.
    - Symmetric north and south, periodic east and west boundaries.
    - 100-km grid size, 16-km top, with 4-km damping layer.
    - 41x81 points in (x,y), 64 layers.
6. 2-D gravity current (test/em_grav2d_x)
    - Test case is described in Straka et al, *INT J NUMER METH FL* **17** (1): 1-22 July 15 1993.
    - See the README.grav2d_x file in the test directory.
7. 2-D sea breeze (test/em_seabreeze_x)
    - 2-km grid size, 20-km top, land/water.
    - Can be run with full physics, radiation, surface, boundary layer, and land options.
8. 3-D large eddy simulation (test/em_les)
    - 100-m grid size, 2-km top.
    - Surface layer physics with fluxes.
    - Doubly periodic
9. 3-D Held-Suarez (test/em_heldsuarez)
    - global domain, 625 km in x-direction, 556 km in y-direction, 120-km top.
    - Radiation, polar filter above 45°.
    - Period in x-direction, polar boundary conditions in y-direction
10. 1-D single column model (test/em_scm_xy)

- 4-km grid size, 12-km top
- Full physics
- Doubly periodic

11. 3-D surface fire (test/em_fire)
    - Geoscientific Model Development Discussions (*GMDD*) **4**, 497-545, 2011, http://www.geosci-model-dev-discuss.net/4/497/2011/gmdd-4-497-2011.html
    - 50-m, 4.5-km top
    - 10:1 subgrid ratio, no physics
    - Open boundaries
12. 3-D tropical cyclone (test/em_tropical_cyclone)
    - Test case described in Jordan, *J METEOR* **15**, 91-97, 1958.
    - 15-km, 25-km top
    - f-plane (f=0.5e-5, about 20 N), SST=28 C
    - Full physics with a simple radiative cooling, no cumulus
    - Doubly periodic
13. 3-D convective-radiative equilibrium (test/em_convrad)
    - 1 km grid size, 30 km model top
    - tropical condition, small f, weak wind, constant SST
    - full physics
    - doubly periodic

Initialization for Real Data Cases

Real-data WRF cases use input data to the "real.exe" program provided by the WRF Preprocessing System (WPS), which was originally generated from a previously-run external analysis or forecast model (e.g., GFS).

Suppose a single-domain WRF forecast is desired, with the following criteria:

- 2000 January 24 1200 UTC through January 25 1200 UTC
- the original GriB data is available at 6-h increments

The following coarse-grid files will be generated by the WPS (starting date through ending date, at 6-h increments):

- **`met_em.d01.2000-01-24_12:00:00.nc`**
- **`met_em.d01.2000-01-24_18:00:00.nc`**
- **`met_em.d01.2000-01-25_00:00:00.nc`**
- **`met_em.d01.2000-01-25_06:00:00.nc`**
- **`met_em.d01.2000-01-25_12:00:00.nc`**

The convention is to use "**met_**" to signify data output from the WPS “metgrid.exe” program and input into the “real.exe” program. The "**d01**" portion of the name identifies to which domain this data refers (which permits nesting). The next set of characters is the validation date/time (UTC), where each WPS output file has only a single time-slice of processed data. The file extension suffix “**.nc**” refers to the output format from WPS, which must be in netCDF for the “real.exe” program. For regional forecasts, multiple time periods must be processed by “real.exe” so that a lateral boundary file is available to the model. The global option for WRF requires only an initial condition.

The WPS package delivers data that is ready to be used in the WRF system by the “real.exe” program.

- The data adheres to the WRF IO API. Unless developing special tools, stick with the netCDF option to communicate between the WPS package and “real.exe”.
- The data has already been horizontally interpolated to the correct grid-point staggering for each variable, and winds are correctly rotated to the WRF model map projection.
- 3-D meteorological data required from the WPS: pressure, u, v, temperature, relative humidity, geopotential height
- Optional 3-D hydrometeor data may be provided to the real program at run-time, but these fields will not be used in the coarse-grid lateral boundary file. Fields named: QR, QC, QS, QI, QG, QH, QNI (mixing ratio for rain, cloud, snow, ice, graupel, hail, and number concentration) are eligible for input from the metgrid output files.
- 3D soil data from the WPS: soil temperature, soil moisture, soil liquid (optional, depending on physics choices in the WRF model)
- 2D meteorological data from the WPS: sea level pressure, surface pressure, surface u and v, surface temperature, surface relative humidity, first-guess topography elevation
- 2-D meteorological optional data from WPS: sea surface temperature, physical snow depth, water equivalent snow depth
- 2D static data for the physical surface: terrain elevation, land use categories, soil texture categories, temporally-interpolated monthly data, land sea mask, elevation of the input model’s topography
- 2D static data for the projection: map factors, Coriolis, projection rotation, computational latitude
- constants: domain size, grid distances, date
- WPS data may either be isobaric or a more-generalized vertical coordinate, where each column is monotonic in pressure
- The output from WPS may include meteorological data on one vertical coordinate, and other 3-D data (such as emissions) on an entirely different vertical coordinate
- All 3-D meteorological data (wind, temperature, height, moisture, pressure) must have the same number of levels, and variables must have the exact same levels. For example, it is not acceptable to have more levels for temperature (for

example) than height. Also, it is not acceptable to have vertical level 8 = 700 hPa for temperature, but have level 8 = 750 hPa for the horizontal wind components.

Real Data Test Case: 2000 January 24/12 through 25/12

- A test data set is accessible from the WRF download page. Under the "WRF Model Test Data" list, select the January data. This is a 74x61, 30-km domain centered over the eastern US.
- Make sure you have successfully built the code (fine-grid nested initial data is available in the download, so the code may be built with the basic nest option), **`./WRF/main/real.exe`** and **`./WRF/main/wrf.exe`** must both exist.
- In the **`./WRF/test/em_real`** directory, copy the namelist for the January case to the default name

    - **`cp namelist.input.jan00 namelist.input`**

- Link the WPS files (the "**`met_em*`**" files from the download) into the **`./WRF/test/em_real`** directory.
- For a single processor, to execute the real program, type **`real.exe`** (this should take less than a minute for this small case with five time periods).
- After running the "real.exe" program, the files "**`wrfinput_d01`**" and "**`wrfbdy_d01`**" should be in this directory; these files will be directly used by the WRF model.
- The "wrf.exe" program is executed next (type **`wrf.exe`**), this should only take a few minutes (only a 12-h forecast is requested in the namelist file).
- The output file **`wrfout_d01:2000-01-24_12:00:00`** should contain a 12-h forecast at 3-h intervals.

Considerations for Recent Releases

The default behavior is to include the moist potential temperature option and to include the hybrid vertical coordinate. These two options make backward compatibility difficult.

- Moist potential temperature: use_theta_m = 0 (off), 1 (on)

The initial moist potential temperature capability was introduced in the WRF system with v3.8. In v3.8 through v3.9.1.1, all processing for moist theta was handled by the model. This caused repeated toggling back and forth inside the main solver routine, so that the definition of the variable grid%t_2 was dependent on the location within the routine. The code has now been properly ported so that the moist theta option is incorporated in the real/ideal pre-processor, and the meaning of the grid%t_2 variable is always the same.

All real and ideal cases support the moist potential temperature option. Because the code assumes the input variables are consistent with the namelist setting for moist theta, wrfinput_d0x and wrfbdy_d01 files from earlier versions may not generally be used.

> If users have older input data (pre v4.0), and they turn off the moist theta option, then wrfinput_d0x and wrfbdy_d01 data may be used.
>
> If users turns on the moist potential temperature option, only new wrfinput_d0x and wrfbdy_d01 data may be used.

- Hybrid vertical coordinate (HVC): hybrid_opt = 0 (off), 2 (on)

WRF code supports a hybrid vertical coordinate, but only for all real-data and ideal cases. The same sort of proscriptions applies with the hybrid vertical coordinate as with the moist theta option.

> If users have older input data (pre v4.0) that does not use HVC, and they turn off the hybrid vertical coordinate option in the WRF model, then wrfinput_d0x and wrfbdy_d01 data may be used.
>
> If users turn on the HVC option, only new wrfinput_d0x and wrfbdy_d01 data may be used.

A namelist option is available (*force_use_old_data=.TRUE.*) to explicitly allow bringing in old wrfinput_d0x and wrfbdy_d01 files to the WRF model.

Setting Model Vertical Levels

Users may explicitly define full eta levels using the namelist option *eta_levels*. Given are two distributions for 28 and 35 levels. The number of levels must agree with the number of eta surfaces allocated (*e_vert*). Users may alternatively request only the number of levels (with *e_vert*), and the real program will compute values. There are two methods that can be selected: *auto_levels_opt = 1 (old) or 2 (new).* The old computation assumes a known first several layers, then generates equi-height spaced levels up to the top of the model. The new method uses surface and upper stretching factors (*dz_stretch_s* and *dz_stretch_u*) to stretch levels according to log p, up to the point of maximum thickness (*max_dz*), and starting from thickness *dzbot.* The stretching transitions from *dzstretch_s* to *dzstretch_u* by the time the thickness reaches *max_dz*/2.

Minimum number of levels as function of dzstretch and p_top for dzbot=50 m and max_dz=1000 m

dzstretch\ptop	50	30	20	10	1

1.1	44	47	50	54	67
1.2	32	35	37	41	54
1.3*	28	31	33	37	50

*1.3 reaches 1 km thickness below about 5 km (level 13) – probably not recommended
1.2 reaches 1 km thickness at around 7 km (level 19)
1.1 reaches 1 km thickness at around 13 km (level 36)

dzstretch = 1.1 has 12 levels in lowest 1 km, 34 levels below 10 km
dzstretch = 1.2 has 9 levels in lowest 1 km, 22 levels below 10 km
dzstretch = 1.3 has 8 levels in lowest 1 km, 18 levels below 10 km

Minimum number of levels when dzstretch_s and dzstretch_u are used

dzstretch\ptop	50	30	20	10	1
1.2-1.02	53	58	62	67	81
1.2-1.04	46	49	51	55	68
1.2-1.06	41	44	47	50	63
1.3-1.1	33	36	39	43	56

To avoid max thickness in the upper troposphere, stretching levels need to extend above the tropopause before going to constant d (logp). This can be done by using low enough *dzstretch_u* values (but larger than ~1.02) to reach the tropopause, while also stretching fast enough to compensate lapse rate.

Two other namelists can be used to add flexibility: *dzbot*, which is the thickness of the first model layer between full levels (the default value is 50 m), and *max_dz*, which is the maximum layer thickness allowed with the default value of 1000 m.