Real-data WRF checklist (for generic model runs; batch/burst/burst-daes)

ATM419/563 Fovell Fall 2024

* ---------- **general** ---------- *

• **Some** lines in `namelist.wps` and `namelist.input` permit multiple columns, one for each domain when `max_dom` > 1

• Copy and unpack $LAB/SCRIPTS/WRF_REAL_SETUP.TAR

• execute make_all_links.sh

• **Submit scripts may also need editing**

• **Submit a job to Burst: sbatch -p burst-daes submit_XXXXX**

• **Check on your jobs: `myjobs`**

* ---------- **WPS ----[namelist.wps]---------- ***

* ---------- ***geogrid*** ---------- *

• Modify `&geogrid` section of `namelist.wps` or your domain sizes and resolution, geographic data, set resolution, and map projection particulars

• For multiple domains (nests), set not only the size of the interior domains, but also their placement within their parent domain

• You are free to use my geographic database. For WRFV4.0+ it is
`geog_data_path = '/network/rit/lab/atm419lab/GEOG4.0/',`

• To flight test your domain use `plot_WRF_domain.ipynb`

• Run `geogrid.exe`. Can run on multiple CPUs. Use `submit_geogrid` OR
`srun -p burst-daes -n 4 geogrid.exe`

• Check for "Successful completion"

• Check your **map factors**! {`ncview` your file, look at MAPFAC_M}

* ---------- ***ungrib*** ---------- *

• Edit `&share` section of `namelist.wps` for `start_date, end_date, and interval_seconds`

• The `&ungrib` section declares the ungrib output filename (i.e., `prefix='FILE'`)

• Use `link_grib.csh` to link to your parent model data

• Make sure you are using the correct `Vtable`

• Run `ungrib.exe`. **One cpu only!!!!**

• This may run slowly on global, high-resolution, and/or temporally dense datasets.

Use batch script `submit_ungrib` OR `srun -p burst-daes ungrib.exe`

• Check for "Successful completion"

* ---------- ***metgrid*** ---------- *

• '`fg_name`' in the `&metgrid` section tells metgrid what the ungrib-produced files Are named. Most often, "`fg_name`" is the **same** as "`prefix`".

• Run `metgrid.exe`. Can run on multiple CPUs. Consider `submit_metgrid`

• Check for "Successful completion"

• Use `ncdump` on a `met_em.d0*` output file for `num_metgrid_levels, num_st_layers,` and `num_sm_layers`

*** ---------- WRF ----[namelist.input]---------- ***

*** ---------- *real* ---------- ***

- Edit `namelist.input` to make sure your `start_*` and `end_*` times, `interval_seconds`, number of domains, domain resolutions and sizes, and nest placement details, are **consistent** with what you specified in `namelist.wps`
- Make sure `num_metgrid_levels` is correct, and `num_metgrid_soil_levels` matches `num_st_layers` ***and*** `num_sm_layers`
- Choose your model physics. Some physics selections need to be set prior to running `real.exe`. To be safe, it is best to rerun `real.exe` when you make any `namelist.input` changes
- For `max_dom` > 1 some lines require a selection for **each** domain, some don't. If you need to provide a selection and do not, **the default is used instead**. This is often not a good choice!!! For some lines, you can select a different physics option for each domain – for some physics, that's not only good but necessary, but for others it's highly inadvised
- The `real.exe` program uses `start_*` and `end_*` times, and `interval_seconds`, to control how often and how long boundary tendency information is generated, and when such generation starts and ends. The `run_*` information (i.e., `run_hours`) is ignored
- Run `real.exe`. You can run this on multiple CPUs. For smaller, less complex problems, '`srun`' is fine. Otherwise, use submit script `submit_real`

- Follow progress with `tail -f rsl.out.0000` ["`trsl`"]. Break out with **ctrl-c.**

- Look for "`SUCCESS COMPLETE REAL_EM`"

*** ---------- *wrf.exe* ---------- ***

- The `wrf.exe` program uses `start_*` and `run_*` information to determine when The model starts and how long it runs. For the outermost domain, `end_*` information is ignored. Predictions in the nests can be ended earlier than the parent domain and/or started later (can be problematic).

- Submit batch script `submit_wrf`, note job number.

- Follow progress with `tail -f rsl.out.0000` ["`trsl`"]. Break out with **ctrl-c**.

- Look for "`SUCCESS COMPLETE WRF`"

[end]