Tutorial for Running WRFV4 on Optimum

Updated: January 2023

Preliminaries

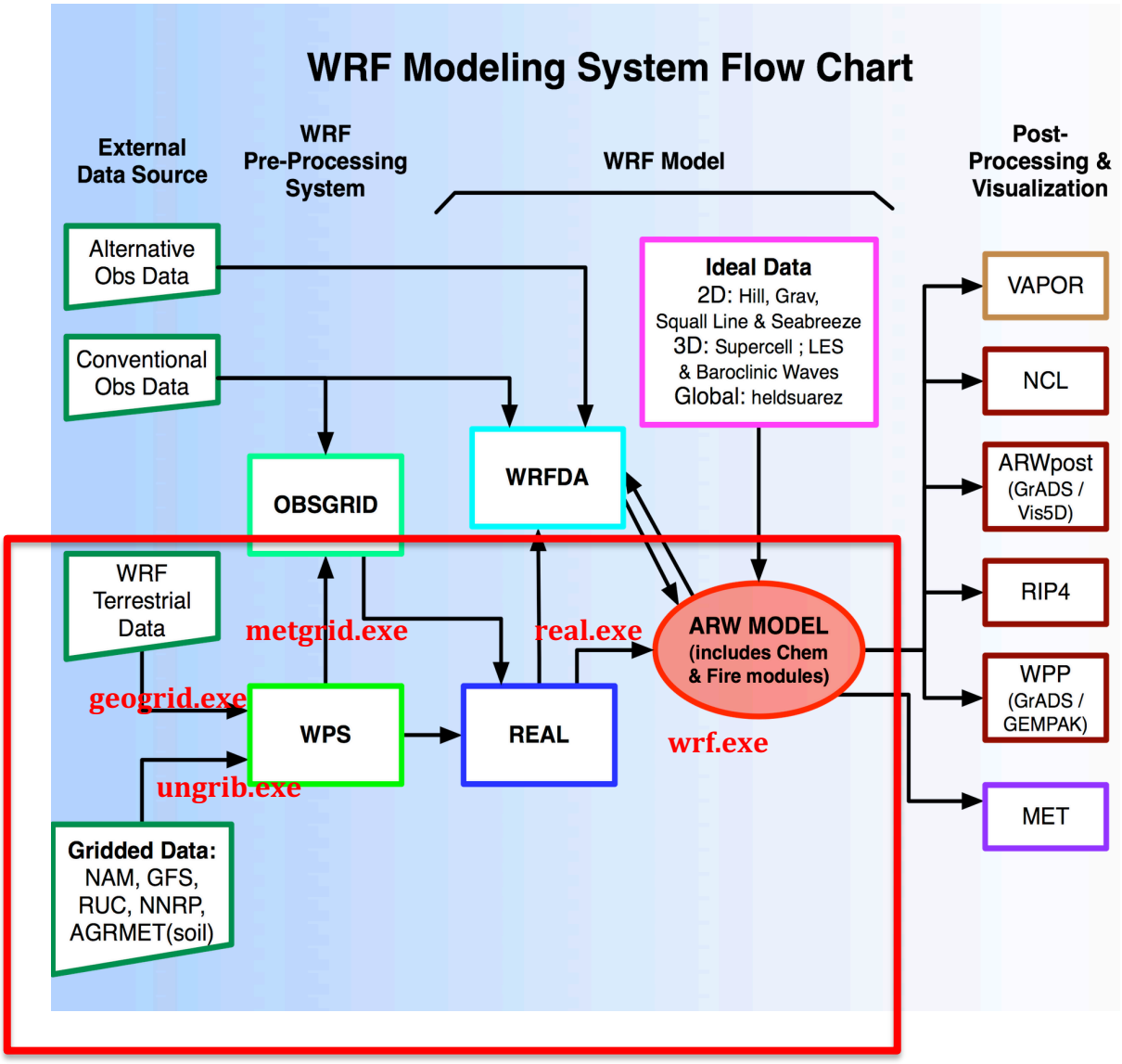
- Ensure that WRF has been installed in ~/WRF/WRF
- Ensure that WPS has been installed in ~/WRF/WPS
- Ensure that the proper modules have been loaded
    - `module load GCC/8.3/0`
    - `module load OpenMPI/4.0.0/GCC/8.3`
    - Can add these to your ~/.bashrc as follows for automatic startup:

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific aliases and functions

module load GCC/8.3/0
module load OpenMPI/4.0.0/GCC/8.3
```

- Relevant data directories have been placed in /data/rstull/shared/ATSC507
    - **DO NOT WRITE INTO THIS DIRECTORY; YOU SHOULD ONLY BE COPYING STUFF FROM THIS DIRECTORY**
    - GEOG
        - Terrestrial data needed for running geogrid.exe
    - IBCS

        - U.S. Global Forecast System (GFS) Initial-boundary conditions needed for running ungrib.exe
        - Valid from 2018-12-18 0000 UTC to 2018-12-23 0000 UTC
            - Wind storm case study (highest number of BC Hydro outages in provincial history)
    - metgrid
        - Sample metgrid files produced from metgrid.exe
    - ungrib (will not be used)
        - Sample ungrib files produced from ungrib.exe
    - WRF
        - Sample wrfout files produced from real.exe and wrf.exe; baseline case

Workflow



- **geogrid.exe (~/WRF/WPS/geogrid)**
  - Interpolates terrestrial data to user-defined model domain
    - Includes terrain heights, land-use data (e.g. vegetation types), land/sea flags, etc.
    - Interpolation options controlled by GEOGRID.TBL
    - **Terrestrial data already downloaded in /data/rstull/shared/ATSC507/GEOG**
      - You generally need to do this yourself, but there's no point in everyone each having a copy of the same 30+ GB directory, and because it takes time to download and unpack

- Creates geo_em*.nc files containing terrestrial data for the domains
  - Used as input into metgrid.exe

- Controlled by namelist.wps (copy found in ~/WRF/WPS)
  - Only domain information matters (i.e. projection, domain bounds, nest starting points)
  - Ignores timing information

- Will only need to run once for a new domain; because domains don't change day-to-day, we don't run geogrid.exe daily

- **ungrib.exe (~/WRF/WPS/ungrib)**
  - Translates raw national centre Grib/Grib2 meteorological data files into an intermediate format (FILE*) for metgrid.exe

  - Controlled by namelist.wps
    - Only timing information matters (i.e. start date, end date, frequency of file output)
    - Ignores domain information

  - Variables in Grib/Grib2 files have an encoding that matches World Meteorological Organization (WMO) standard
    - Different national centres have different names for the same variables, e.g. Meteo-France may have a different name for 2-m Temperature than Environment Canada, but 2-m Temperature is 2-m Temperature…so the WMO Grib code is the same for both

  - Grib codes are translated by ungrib.exe based on information given in variable tables (Vtables → found in ~/WRF/WPS/ungrib/Variable_Tables)

  - Needs to be run each time a new forecast initialization time is used, i.e. run once for 2019-12-18 0000 UTC; run once for 2019-12-18 0600 UTC; run once for 2019-12-18 1200 UTC; run once for 2019-12-18 1800 UTC, etc.

- **metgrid.exe (~/WRF/WPS/metgrid)**
    - Combines the output from geogrid.exe (geo_em*.nc) and ungrib.exe (FILE*)
        - Horizontally interpolates intermediate meteorological data files onto the domain created by geogrid.exe
        - Interpolation options controlled by METGRID.TBL
    - Controlled by namelist.wps
        - All domain and timing information needed
    - Outputs met_em*.nc files, for use in real.exe
    - Needs to be run after each new run of ungrib.exe

- **real.exe (~/WRF/WRF/main/real.exe)**
    - Takes the output from metgrid.exe and performs the required initializations prior to wrf.exe
    - Vertical interpolation of fields given by metgrid.exe onto user-defined model vertical levels
    - Pre-allocation of needed arrays (including scalar arrays in microphysics schemes)
    - Creation of initial condition (wrfinput*) and boundary condition (wrfbdy*) files
    - Controlled by namelist.input
        - All information needed
    - Needs to be run after each new run of metgrid.exe
    - If namelist.input is changed after real.exe is run, real.exe must be run again prior to running wrf.exe

- **wrf.exe (~/WRF/WRF/main/wrf.exe)**
    - Runs WRF, and outputs wrfout* files containing model-produced fields
    - Controlled by namelist.input
        - Must match namelist.input for real.exe

Tutorial

- **GOAL: Make a single-domain simulation of the December 20, 2018 wind storm**
    - 5-day forecast initialized on December 18, 2018 at 0000 UTC, centred over British Columbia
    - Experiment with different planetary boundary-layer schemes to see how they affect the forecast
    - Gain expertise in initializing and running a real-data WRF simulation
    - Gain expertise in simple WRF output visualization
- Log onto optimum
    - `ssh username@optimum.eos.ubc.ca`
- Ensure modules are loaded (they should be in ~/.bashrc)
    - `module load GCC/8.3/0`
    - `module load OpenMPI/4.0.0/GCC/8.3`
- Go into your user-allocated scratch directory; this is the directory we'll be writing into for our WRF runs
    - `cd $SCRATCHDIR`
    - `pwd`
        - You should see that you're in /scratch/rstull/username
    - **DO NOT MAKE WRF RUNS IN ~/ (see warning message when you first log onto optimum)**
- Make a tutorial directory and cd into it
    - `mkdir tutorial`
    - `cd tutorial`
    - **We should never make runs in the source WRF or WPS directories; those directories are meant to be originals. Instead, always make runs by copying or linking required files into your own directories.**
- Make a WPS directory and cd into it
    - `mkdir WPS`
    - `cd WPS`
- Copy over a blank namelist.wps
    - `cp ~/WRF/WPS/namelist.wps .`

- Edit your namelist.wps with the following geographic information:

```
&share
 wrf_core = 'ARW',
 max_dom = 1,   Total number of domains
 start_date = '2018-12-18_00:00:00',
 end_date   = '2018-12-23_00:00:00',
 interval_seconds = 10800
 io_form_geogrid = 2,
/

&geogrid
 parent_id         =   1,        ID of parent domain
 parent_grid_ratio =   1,      Grid-spacing ratio of parent
 i_parent_start    =   1,   x-coordinate starting location relative to parent
 j_parent_start    =   1,   y-coordinate starting location relative to parent
 e_we              =  121, Number of west-east (x-direction) points
 e_sn              =  121, Number of south-north (y-direction) points
 !
 !!!!!!!!!!!!!!!!!!!!!!!!!! IMPORTANT NOTE !!!!!!!!!!!!!!!!!!!!!!!!!!
 ! The default datasets used to produce the MAXSNOALB and ALBEDO12M
 ! fields have changed in WPS v4.0. These fields are now interpolated
 ! from MODIS-based datasets.
 !
 ! To match the output given by the default namelist.wps in WPS v3.9.1,
 ! the following setting for geog_data_res may be used:
 !
 ! geog_data_res = 'maxsnowalb_ncep+albedo_ncep+default', 'maxsnowalb_ncep+albedo_ncep+default',
 !
 !!!!!!!!!!!!!!!!!!!!!!!!!! IMPORTANT NOTE !!!!!!!!!!!!!!!!!!!!!!!!!!
 !
 geog_data_res = 'default',   Interpolation resolution (defaults set in GEOGRID.TBL)
 dx = 36000,    x-direction grid spacing (m)
 dy = 36000,    y-direction grid spacing (m)
 map_proj = 'polar',    Map projection (polar stereographic for high latitudes)
 ref_lat   =  47.83,    Latitude of center of coarsest domain
 ref_lon   = -127.3,    Longitude of center of coarsest domain
 truelat1  =  60.0,     True latitude (i.e. no distortion)
 truelat2  =  90.0,     Second true latitude (i.e. not used for polar stereographic)
 stand_lon = -90.0,     Longitude parallel to y-axis
 geog_data_path = '/data/rstull/shared/ATSC507/GEOG'  Location of terrestrial data
 opt_geogrid_tbl_path = '.'   Location of GEOGRID.TBL
/

&ungrib
 out_format = 'WPS',
 prefix = 'FILE',
/

&metgrid
 fg_name = 'FILE'
 io_form_metgrid = 2,
 opt_metgrid_tbl_path = '.'
/
```

- Link over geogrid.exe and GEOGRID.TBL into tutorial/WPS
    - `ln -s ~/WRF/WPS/geogrid.exe .`
    - `ln -s ~/WRF/WPS/geogrid/GEOGRID.TBL`

- Run geogrid.exe by invoking an interactive session
    - `Iqsub`
        - Interactive job submission, so that we can log onto a compute node for runs
        - We cannot make large compute/memory-heavy runs on the login node (i.e. sigma, delta)

    - `Iqsub 0.5 1 1`
        - Request an interactive session for half an hour, with 1 node and 1 processor on the node

    - `./geogrid.exe`

    - `ls`
        - Should see that geo_em.d01.nc has been produced

    - `exit`
        - Log off the compute node to return it back to the queue; we don't want to waste unused resources

- Edit your namelist.wps with the following timing information:

```
&share
 wrf_core = 'ARW',
 max_dom = 1,
 start_date = '2018-12-18_00:00:00',
 end_date   = '2018-12-18_12:00:00',
 interval_seconds = 10800
 io_form_geogrid = 2,
/

&geogrid
 parent_id         =   1,
 parent_grid_ratio =   1,
 i_parent_start    =   1,
 j_parent_start    =   1,
 e_we              =  121,
 e_sn              =  121,
 !
 !!!!!!!!!!!!!!!!!!!!!!!!!!! IMPORTANT NOTE !!!!!!!!!!!!!!!!!!!!!!!!!!!
 ! The default datasets used to produce the MAXSNOALB and ALBEDO12M
 ! fields have changed in WPS v4.0. These fields are now interpolated
 ! from MODIS-based datasets.
 !
 ! To match the output given by the default namelist.wps in WPS v3.9.1,
 ! the following setting for geog_data_res may be used:
 !
 ! geog_data_res = 'maxsnowalb_ncep+albedo_ncep+default', 'maxsnowalb_ncep+albedo_ncep+default',
 !
 !!!!!!!!!!!!!!!!!!!!!!!!!!! IMPORTANT NOTE !!!!!!!!!!!!!!!!!!!!!!!!!!!
 !
 geog_data_res = 'default',
 dx = 36000,
 dy = 36000,
 map_proj = 'polar',
 ref_lat   =  47.83,
 ref_lon   = -127.3,
 truelat1  =  60.0,
 truelat2  =  90.0,
 stand_lon = -90.0,
 geog_data_path = '/data/rstull/shared/ATSC507/GEOG'
 opt_geogrid_tbl_path = '.'
/

&ungrib
 out_format = 'WPS',
 prefix = 'FILE',
/

&metgrid
 fg_name = 'FILE'
 io_form_metgrid = 2,
 opt_metgrid_tbl_path = '.'
/
```

Start date corresponding to Grib files

End date corresponding to Grib files; we only want a 12-hour run for now

Time period between Grib files (3 hours)

Location of METGRID.TBL for metgrid.exe

- Link in ungrib-related files
  - `ln -s ~/WRF/WPS/ungrib.exe .`
  - `ln -s ~/WRF/WPS/ungrib/Variable_Tables/Vtable.GFS ./Vtable`
    - The line above should be on one line
    - Vtable.GFS must be named Vtable in the current directory for ungrib.exe
  - `ln -s ~/WRF/WPS/link_grib.csh .`
    - This is a shell script used to link the Grib files in an alphabetical format recognized by ungrib.exe

- Link in the Grib files using link_grib.csh
  - `./link_grib.csh /data/rstull/shared/ATSC507/IBCS/gfs*`
  - `ls`
    - Should see a how bunch of linked files like GRIBFILE.AAA, GRIBFILE.AAB, etc.
    - You can check that they're linked to the original GFS Grib files by running `ls –lh`

- Run ungrib.exe (we are only doing a 12-hour run; the full 5-day run would take too long in WPS)
  - `Iqsub 0.5 1 1`
  - `./ungrib.exe`
  - `ls`
    - Should see FILE:2018-12-18_00, FILE:2018-12-18_03, etc.
  - `exit`

- Link in metgrid-related files
  - `ln -s ~/WRF/WPS/metgrid.exe .`
  - `ln -s ~/WRF/WPS/metgrid/METGRID.TBL .`

- Run metgrid.exe (we are only doing a 12-hour run; the full 5-day run would take too long in WPS)
  - `Iqsub 0.5 1 1`
  - `./metgrid.exe`
  - `ls`
    - Should see met_em.d01.2018-12-18_00:00:00.nc, etc.
  - `exit`

- WPS is done! For WRF, we'll be using the full 5-day metgrid files that have been pre-made

- Return to the tutorial directory
  - `cd ..`

- Make a WRF directory, and cd into it
  - `mkdir WRF`
  - `cd WRF`

- Link in all required files, including look-up tables for physics schemes
  - `ln -s ~/WRF/WRF/test/em_real/* .`

- However, we don't want a link of namelist.input, because we don't want to change the original; hence, we should remove the link, and replace it with a copy instead
  - `rm namelist.input`
  - `cp ~/WRF/WRF/test/em_real/namelist.input .`

- Make the following changes to namelist.input:

```
&time_control
run_days                            = 0,
run_hours                           = 120,    Number of hours for run
run_minutes                         = 0,
run_seconds                         = 0,
start_year                          = 2018,   Start year
start_month                         = 12,     Start month
start_day                           = 18,     Start day
start_hour                          = 00,     Start hour
end_year                            = 2018,   End year
end_month                           = 12,     End month
end_day                             = 23,     End day
end_hour                            = 00,     End hour
interval_seconds                    = 10800   Frequency between met_em fields (seconds)
input_from_file                     = .true.,
history_interval                    = 60,     Frequency of WRF output (minutes)
frames_per_outfile                  = 1,      How many output times per wrfout file
restart                             = .false.,
restart_interval                    = 20000,  Minutes between restart file output
io_form_history                     = 2
io_form_restart                     = 2
io_form_input                       = 2
io_form_boundary                    = 2
/

&domains
time_step                           = 216,    Base time step (6*dx recommended)
time_step_fract_num                 = 0,
time_step_fract_den                 = 1,
max_dom                             = 1,
e_we                                = 121,    Must match namelist.wps (i.e. met_em*)
e_sn                                = 121,    Must match namelist.wps (i.e. met_em*)
e_vert                              = 41,     Number of vertical levels requested
p_top_requested                     = 5000,
num_metgrid_levels                  = 32,     Number of vertical levels in original Grib files
num_metgrid_soil_levels             = 4,      Number of soil levels in original Grib files
dx                                  = 36000,  Must match namelist.wps (i.e. met_em*)
dy                                  = 36000,  Must match namelist.wps (i.e. met_em*)
grid_id                             = 1,
parent_id                           = 0,
i_parent_start                      = 1,
j_parent_start                      = 1,
parent_grid_ratio                   = 1,
parent_time_step_ratio              = 1,
feedback                            = 1,
smooth_option                       = 0
use_adaptive_time_step              = .true.  Use adaptive time step
step_to_output_time                 = .true.  Align time step to output time so
                                              weird output times don't show up
max_step_increase_pct               = 5,      Maximum time step increase
target_cfl                          = 1,      CFL condition for adaptive time step to meet
/
```

```
&physics
physics_suite                       = 'CONUS'    CONUS physics suite
mp_physics                          = -1,
cu_physics                          = -1,
ra_lw_physics                       = -1,
ra_sw_physics                       = -1,
bl_pbl_physics                      = -1,    PBL scheme; -1 = use CONUS PBL scheme (MYJ)
sf_sfclay_physics                   = -1,      1     = YSU (sf_sfclay_physics = 1)
sf_surface_physics                  = -1,      5     = MYNN2 (sf_sfclay_physics = 5)
radt                                = 30,      7     = ACM2 (sf_sfclay_physics = 1)
bldt                                = 0,       11    = Shin-Hong (sf_sfclay_physics = 1)
cudt                                = 0,       12    = GBM (sf_sfclay_physics = 1)
icloud                              = 1,
num_land_cat                        = 21, Land-use categories (geogrid default uses MODIS = 21 class)
sf_urban_physics                    = 0,
/

&fdda
/

&dynamics
hybrid_opt                          = 2,     Use hybrid-eta vertical coordinates (hybrid_opt = eta)
w_damping                           = 0,
diff_opt                            = 1,
km_opt                              = 4,
diff_6th_opt                        = 0,
diff_6th_factor                     = 0.12,
base_temp                           = 290.
damp_opt                            = 3,
zdamp                               = 5000.,
dampcoef                            = 0.2,
khdif                               = 0,
kvdif                               = 0,
non_hydrostatic                     = .true.,
moist_adv_opt                       = 1,
scalar_adv_opt                      = 1,
gwd_opt                             = 1,
/

&bdy_control
spec_bdy_width                      = 5,
specified                           = .true.
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/
```

- Link in finished met_em* files
  - `ln -s /data/rstull/shared/ATSC507/metgrid/* .`

- Run real.exe; make sure you've chosen your desired boundary-layer scheme (bl_pbl_physics) and accompanying surface-layer scheme (sf_sfclay_physics)
  - `Iqsub 1 1 10`
    - We're going to request 1 hour, and 10 cores, so that we can run wrf.exe in parallel immediately after
  - `mpirun –np 1 ./real.exe`
  - `cat rsl.error.0000`
    - Log written here; should see SUCCESS COMPLETE REAL_EM INIT at bottom

- Run wrf.exe with 10 processors (ranks; -np 10)
  - `nohup mpirun –np 10 ./wrf.exe &`
    - Note the & at the ver end
    - nohup <blah> & sends <blah> to the background and disowns the process, allowing you to use the command line and log out without fear of losing the process
  - `cat nohup.out`
    - Allows you to see the immediate stdout and stderr output (i.e. what would have been printed to the screen if you hadn't used nohup)
    - You should see
      - starting wrf task 5 of 10
      - starting wrf task 7 of 10
      - etc. (order doesn't matter)
  - `ls -lh rsl*`
    - Should see one rsl.error* and rsl.out* file for each process
    - rsl.error.0000 (log file of master process) has everything printed to it, and is the master log file
    - Other files may contain error information not printed to rsl.error.0000, including CFL-related errors
    - If a run fails unexpectedly and rsl.error.0000 does not give the required information, you can do the following searches:
      - `grep cfl rsl*`
        - Searches all rsl files for the string "cfl"
      - `grep error rsl*`
      - `grep ERROR rsl*`
      - `tail -n5 rsl*`
        - Prints out the last 5 lines in each rsl file
  - `tail -f rsl.error.0000`
    - You can watch the model run as the log is printed in real-time to screen (should see SUCCESS COMPLETE WRF once done)
    - Ctrl + C to escape

- While we're waiting, we can experiment with IDV using already created grids in /data/rstull/shared/ATSC507/WRF/

- Open up a new terminal window on your laptop, and find a location for you to hold some wrfout files

    - `mkdir wrfouts_original`
    - `cd wrfouts_original`
    - `scp username@optimum.eos.ubc.ca:/data/rstull/shared/ATSC507/WRF/wrfout_d01_2018-12-20* .`
        - All on one line
    - `scp username@optimum.eos.ubc.ca:/data/rstull/shared/ATSC507/WRF/wrfout_d01_2018-12-21* .`
        - All on one line

    - Total: 2.8 GB

- Download IDV and install onto your computer
    - https://www.unidata.ucar.edu/downloads/idv/current/index.jsp