**The Weather and Research Forecasting Model (WRF)**

The WRF is a numerical weather prediction model system that can be used for simulation of past and future weather events. It is a grid-based model, meaning that derivatives are solved using finite differencing techniques on gridded fields of meteorological variables. The WRF is a highly technical piece of software that many have made careers out of mastering. The best way to learn WRF is through the use of tutorials and the user’s guide, and the WRF forum (<http://forum.wrfforum.com/>) often has answers to common questions as well. Below is a summary of the steps needed to run a WRF simulation.

1. Installation of WRF
   1. The WRF and its associated utilities can be downloaded from <http://www2.mmm.ucar.edu/wrf/users/download/get_source.html>. You should download the WRFV3 file, the WPS file, and choose a post-processing utility to download (see below for the list of possible options). In addition to downloading the WPS file, you should also consider downloading the subset of land-surface cover data you will require for your simulation. Note the full dataset is roughly 50 GB in size, so be prepared for a lengthy download and considerable disk space to store the files (they only need to be downloaded once).
   2. In addition to downloading the WRF, you will need to download external libraries that the WRF requires, which include: netCDF libraries for reading and writing WRF files; Jasper, PNG, and zlib libraries for grib2 formatted data; and MPICH for running processes in parallel (which is essential for large WRF simulations)
   3. There is a useful tutorial on building the required libraries and the WRF software for any Linux platform with gnu compilers (which is most Linux systems) <http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php>. Note that this tutorial also contains links to the required external libraries.
2. Download required meteorological fields for the given simulation
   1. Each simulation you complete will require a series of meteorological fields. Several gridded datasets which contain required fields exist for which the WPS is pre-configured, including, the NCEP/NCAR reanalysis (NNRP), the North American Reanalysis (NARR), and the Global Forecast System (GFS).
3. Running the WRF Preprocessing System (WPS)
   1. Setting up the *namelist.wps* file – this file contains several important options that should be configured for your given simulation, including starting and ending date, map projection and study domain size, grid spacing, and frequency of the input meteorological fields
   2. Running *geogrid.exe* – this is the first executable to run, and creates a file *geo\_em.d01.nc* (note the *01* can be different if you are doing a nested run). This program takes the geography data you previously downloaded and interpolates it to the domain you specify in *namelist.wps*.
   3. Running *ungrib.exe* – this step is required *only if your input meteorological data are in grib or grib2 format*. This step takes the grib/grib2 formatted input data and converts their format to the standard WRF intermediate format (which is described in the WRF user’s guide). If input data are not in grib/grib2 or the WRF intermediate format, they must be written to the intermediate format before running metgrid.exe. The user’s guide provides sample codes that can do this.
   4. Running *metgrid.exe* – this step interpolates and staggers the meteorological input fields for the case and places everything on the domain specified in *namelist.wps*. It produces one *met\_em.d01.YYYY-MM-DD\_HH:MM:SS.nc* file for each file of input data (specified by the user in *namelist.wps*)
4. Running the WRF
   1. All *.nc* files generated by the WPS should be linked to the WRF run directory (the same directory as the executables)
   2. The user should edit the *namelist.input* file (the configuration file for the simulation). This file contains several of the same options as *namelist.wps* that MUST MATCH the *namelist.wps* file before the model will run. Additionally, the *namelist.input* file has additional configuration options, some of which include:
      1. Setting the timestep – the time interval between forward integrations in the model. This should be set no larger than 6\*dx, where dx is the grid spacing of the domain in km (set this smaller if the model is unstable). *Note, there is an adaptive timestep that the user can configure to speed up the simulation time*
      2. Model physics – the user can configure model parameterizations for several meteorological processes, including cloud microphysics, cumulus physics, boundary and surface layer physics, radiation physics, and land surface physics
      3. Stochastic perturbations – for initial condition ensemble simulations
      4. Model dynamics and finite differencing techniques
      5. Many others, as specified in the WRF users guide
      6. *Note that these options can be configured for each nest within a nested simulation by simply adding a new column to namelist.input for each nest. The default namelist.input contains 3 columns for 3 nests, but more can be added*.
   3. Run *real.exe* – creates an initial condition WRF file (needed for each simulation) and a boundary conditions file (necessary for all non-global simulations). The *wrf\_input* file is the initialization timestep of the model.
   4. Run *wrf.exe* – This is the formal model executable that should be run in parallel. The use of multiple cores helps improve performance and effectively utilizes the available system resources on parallel machines. This step takes the longest, and will generate one *wrfout.d01.YYYY-MM-DD-HH:MM:SS* file for each desired output time, as specified in *namelist.input*. One file will be produced for each nest (*d01* will be modified)
5. Post-processing
   1. The *wrfout* files from the previous step contain staggered meteorological fields that need to be destaggered and placed on appropriate vertical coordinates (e.g. isobaric coordinates) before viewing the output. This is completed through a post-processing routine, of which several are available. These include:
      1. Universal Post-Processor – this one gives the most options for diagnostic output, but is difficult to build. Outputs in grib format
      2. RIP4 – utility based on the NCAR command line language that directly provides written text output as well as grib or netCDF output
      3. ARWpost – utility that can directly write GrADS output. Has the fewest possible diagnostic fields of the three listed here, but still includes the most common ones

It is essential that you read and understand the WRF user’s guide before tackling WRF, as it contains detailed information for each step that is briefly described here. It is available at (<http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/contents.html>).