ATSC 5004 (Problems in Dynamic Meteorology)
WRF I
October 16, 2014

Weather Research and Forecasting Model (WRF)

Introduction

This week we will become familiar with the Weather Research and Forecasting (WRF) Model. WRF is a next-generation mesoscale numerical weather prediction system that was originally designed to serve both operational forecasting and atmospheric research needs. WRF is suitable for a broad spectrum of applications across scales ranging from meters to thousands of kilometers. WRF is in the public domain and is freely available for community use. Currently it has two flavors. The Advanced Research WRF (WRF-ARW) developed and maintained by the Mesoscale and Microscale Meteorology Division of NCAR and the Nonhydrostatic Mesoscale Model (WRF-NMM) developed by the National Centers for Environmental Prediction. For our applications and in synoptic meteorology next semester we will use the WRF-ARW.

The effort to develop WRF has been a collaborative partnership, principally among the National Center for Atmospheric Research (NCAR), the National Oceanic and Atmospheric Administration (the National Centers for Environmental Prediction (NCEP) and the Forecast Systems Laboratory (FSL), the Air Force Weather Agency (AFWA), the Naval Research Laboratory, the University of Oklahoma, and the Federal Aviation Administration (FAA). WRF provides operational forecasting a model that is flexible and efficient computationally, while offering the advances in physics, numerics, and data assimilation contributed by the research community.

To start we will need to set up the WRF environment in our home directories on bat. To begin, make a directory called WRF on bat where we will copy the compressed files that contain the WRF source code. Each of us will maintain our own copies of the WRF software. Keep in mind that we will use WRF for simulations in this class as well as for ATSC 5007/5160 next semester. Two separate components are required for our WRF applications. The first is what is known as the WRF Pre-Processing System or WPS. The purpose of WPS is to prepare input to the main WRF program for real-data simulations. WPS consists of three programs. Each of the programs performs one stage of the preparation: *geogrid* defines model domains and interpolates static geographical data to the grids; *ungrib* extracts meteorological fields from GRIB-formatted files; and *metgrid* horizontally interpolates the meteorological fields extracted by *ungrib* to the model grids defined by *geogrid*. For our purposes, we will need to copy the file *WPSV3.5.1.TAR.gz* into your WRF directory. You can find that in /netdata/ R1/data/parish/class/atsc5004/2014. This file is written in a compressed format known as tar. The term ‘tar’ can refer to either the format or the program that can be used to uncompress the file.

The second component to WRF is the actual WRF source code, *WRFV3.5.1.TAR.gz*. Again, you can find this code in /netdata/R1/data/parish/class/atsc5004/2014. Copy this tar file over in your WRF directory as well. As we proceed, we will be following steps that are also outlined in an excellent online WRF tutorial (http://www.mmm.ucar.edu/wrf/OnLineTutorial/index.htm).

Step 1: Unpacking, configuring and compiling WPS, WRF

We will start with the WRF tar file. To untar the file, type:

tar –zxvf WRFV3.5.1.TAR.gz

You will note that a number of directories will be created under your WRF directory including the main directory WRFV3. The unpacking will take about a minute or so and the entire contents will take up about 59 Mb. Once this is complete, untar the WPS file:

tar –zxvf WPSV3.5.1.TAR.gz

Again you will note that more directories will be created including the main WPS directory. Unpacking should take only a few seconds and the files created only take up about 4 Mb. At this point you can delete the original tar files if you wish.

To begin we will 'cd' into the WRFV3 directory. We have to make certain we tell WRF where the NETCDF libraries reside. We only need to do this once. The complete path will be input into subsequent files when we build WRF. Type

setenv NETCDF /opt/pgi/linux86-64

to inform WRF of the correct path for the required files. Once this is done, we have to configure the WRF environment. To do this, simply type './configure'. You will be prompted for a selection of the supported platforms. For the new bat, select choice 2 (Linux x86_64, PGI complier with gcc (smpar)). This selection will allow us to run WRF with multiple processors. You will then be prompted for a selection regarding nesting. Here choose 1= basic. This should allow you to do anything you wish in the future. Once you have run './configure', a file *configure.wrf* will be created. A change needs to be performed before we can compile WRF. We need to add a library in the *configure.wrf* file. Open *configure.wrf* in gedit or any text editor. Then search for the following line:

```
LIB_EXTERNAL    = \

        -L$(WRF_SRC_ROOT_DIR)/external/io_netcdf -lwrfio_nf -L/usr/local/pgi/lib -lnetcdff -lnetcdf
```

You can search for 'LIB_EXTERNAL' in gedit and it will take you to the line (it is on line 207 for my build of WRF). We need to add another library – the 'curl' library. To do this, simply insert '-lcurl' to this line. When done, you should have:

```
LIB_EXTERNAL    = \

        -L$(WRF_SRC_ROOT_DIR)/external/io_netcdf -lwrfio_nf -L/usr/local/pgi/lib -lnetcdff –lnetcdf -lcurl
```

Save the configure.wrf file and then we are ready to compile WRF. We have a choice where we can run WRF but we will choose to run in the directory 'test/em_real' found within the WRFV3 directory. To compile simply type (within the WRFV3 directory):

./compile em_real >& compile.log

This will take a bit of time. For my build, nearly 30 minutes were required. Here we are outputting the results of the compile statement into the file *compile.log* (you can name this log file anything you like – it doesn't have to be *compile.log*). The log file helps us examine the compiling process to make certain everything is correct. If the compiling is successful, you will see the following five files in the /test/em_real directory: *ndown.exe, nup.exe, real.exe, tc.exe and wrf.exe* (note that file names are in the light blue color to show they are soft links to actual executables in the 'main' subdirectory within the WRFV3). If the files exist we are done for now with WRF and need to compile WPS.

To compile WPS, first, 'cd' into the WPS directory and again make certain that WPS knows the NETCDF path. To be safe again type: 'setenv NETCDF /opt/pgi/linux86-64'. We also need to make certain WRF can find appropriate library paths. To do this, we can again set an environmental variable:

setenv LD_LIBRARY_PATH /opt/pgi/linux86-64/lib

We can then configure WPS by typing './configure'. You again will be prompted with a series of choices. For now, choose 5 (Linux, x86_64, PGI compiler (serial)). If the configuration is successful, the file *configure.wps* will be created. You may see a message saying "Your FORTRAN + NETCDF did not run successfully." or "Your versions of Fortran and NETCDF are not consistent.", which we can ignore. Once again we have to add the curl library to the *configure.wps* file. Similar to before, open *configure.wps* and look for the following statements:

```
WRF_LIB      =      -L$(WRF_DIR)/external/io_grib1 -lio_grib1 \
                    -L$(WRF_DIR)/external/io_grib_share -lio_grib_share \
                    -L$(WRF_DIR)/external/io_int -lwrfio_int \
                    -L$(WRF_DIR)/external/io_netcdf -lwrfio_nf \
                    -L$(NETCDF)/lib -lnetcdff -lnetcdf
```

We only need to change the last line (line 47 in my build) by adding '–lcurl'. You can again search in gedit for "WRF_LIB" to find the first line shown. After adding the 'lcurl', the last line should look like:

```
                    -L$(NETCDF)/lib -lnetcdff -lnetcdf -lcurl
```

You can then compile WPS. Simply type './compile'. Compiling WPS will require a couple minutes and if successful you will see three important files have been created: *geogrid.exe, metgrid.exe, ungrib.exe.* Once you have those files, you have completed installing WRF and WPS.

Step 2: Running the preprocessor programs in WPS

To run WRF we first have to decide on a particular case, including selecting grid spacing and the number of nests used, etc. For our class, we will have as a theme the Great Plains low-level jet and pick from a number of days for which the jet was active during the summer months of 2010. Each of you will be assigned a particular case. We will then have to obtain gridded information

for that case, usually in the form of some output from a previous model simulation such as from the NAM, GFS, etc. The process by which output grids are obtained will be shown in class. For your simulations of the low-level jet, appropriate files have been downloaded already and can be found in /netdata/R1/data/nam12km.

a) Running *geogrid.exe*

We start in the WPS directory. The key file to select the various parameters for WPS is the *namelist.wps* file. This is simply a text file that we can edit with any text editor. A sample version of this file is available in /netdata/R1/data/parish/class/atsc5004/2014. In this file we specify the number of domains (we will use 2) and the start date and end date for the run. For our simulations we will begin the run at 1200 UTC the day before the jet case and run until 1200 UTC on the day of your case study for a 24-hour simulation. Also specified in *namelist.wps* is the interval at which we will have output grids for our run, set to 10800 (units in seconds) implying output grids are available at three-hour increments. Domain information is also set in *namelist.wps* such as the number of grid points in the x- and y-directions, where the nests begin, grid spacing of the outer grid and center latitude and longitude. Here are the specifications for your run:

Outer domain: 125 x 125 grid points, grid spacing 15 km, center of grid at 36°N, 98°W, set
geog_data_res = '10m'

Inner domain: 181 x 181 grid points, grid spacing 5 km (set parent_grid_ratio equal to 3), set
geog_data_res = '2m'

You are to set up your domains such that the inner domain is centered over the central Oklahoma region. This will require you to set appropriate values for variables i_parent_start and ji_parent_start, locations where your inner domain begins with regard to your outer domain. You can view the domains using the command 'ncl /util/plotgrids.ncl' that will open a display window with your grids.

As noted above, there are three programs to run to prepare for a WRF simulation. Each of those programs use information in *namelist.wps*. The first is *geogrid.exe* that extracts terrain information. To run *geogrid.exe*, geographical data sets are required. Although each of you can download the complete data set, it is far better to link to an existing data set since the terrain files are quite large. The topographic files used are on /netdata/R1/data/parish/class/atsc5004/WRF/geog and all we need to do is to make certain we link to those files. The sample *namelist.wps* file discussed above already has the proper link. It is our task to define an area for our simulation. In our case, we will choose to run with an outer grid and one nest. The area we will use is centered on northern Oklahoma. This setup has been incorporated into the *namelist.wps* file above, but feel free to modify the geographical region if you would like to experiment. To run *geogrid.exe* simply type ./geogrid.exe. Information will scroll on the screen and, if successful, you will see a banner confirming "Successful completion of geogrid". You will see two files created: *geo_em.d01.nc* and *geo_em.d02.nc*, representing the two domains for your simulation.

b) Running *ungrib.exe*

The second program we need to run in the WPS directory is *ungrib.exe*. This is the start of the process by which we take grids from another model and incorporate them into the WRF environment for the upcoming simulation. The *ungrib.exe* program unpacks the gridded model information we use as input to WRF. As noted above, model output grids have already been obtained and untarred. It may be of use to create a directory within your WPS directory (call it 'data') where you can first create soft links for the files you need. To do this, first create soft links to all files in /netdata/R1/data/nam12km/2010${month}/2010${month}${day} from 1200 UTC the day before your case to 1200 UTC the day of your case. This will be demonstrated in class. We simply need to create another set of soft links to the files in our data directory. To create soft links for WRF in WPS, we use a special program in WRF called link_grib.csh. To link our grids, type the following (here it is assumed we have the directory 'data' that resides within WPS):

./link_grib.csh data/nam_218_2010${month}

Before running *ungrib.exe*, we need to inform WRF about the type of model output files we are using to initialize WRF. This information is passed by way of a file known as a Vtable. It informs WRF as to what variables are found in the file, etc. For our simulations we will be using output from the North American Model or NAM. WRF has built in Vtables in your WPS directory under the subdirectory 'ungrib/Variable_Tables'. If you 'cd' into this directory, you can see WRF has Vtables for many different model formats. We will use the Vtable called Vtable.NAM. You will need to create a soft link into the WPS directory. In your WPS directory, simply type

ln –sf ungrib/Variable_Tables/Vtable.NAM Vtable

Once you have this done, you can run the *ungrib.exe* program. Type:

./ungrib.exe >& ungrib.log

where again you will be creating a log file (called *ungrib.log*, although you can name it whatever you like). Once *ungrib.exe* has completed (may take a couple minutes with the NAM 218 grids), examine the log file in a text editor. If it has run successfully, you should see a line in the text file informing you of the "Successful completion of ungrib".

c) Running *metgrid.exe*

The final program we need to consider in WPS is *metgrid.exe.* This program serves to interpolate the gridded output from *ungrib.exe* to the WRF grid specified in *namelist.wps*. If *ungrib.exe* has been successfully performed, running *metgrid.exe* is usually quick and simple. Simply type './metgrid.exe' and you should see text scrolling through the processing of the various domains. After processing the final domain (domain 2 in our case), text should announce "Successful completion of metgrid" and you have completed all tasks preprocessing tasks required.

Step 3: Running WRF

We have a few steps before we run WRF. First, we need to 'cd' into the WRFV3/test/em_real directory. The key file here is *namelist.input*. This file contains similar information to what is found in *namelist.wps*. Often it is easier to have both files open since it is necessary to have consistent information in each. Make certain the grid sizes, times, etc. match! To help, a sample version of *namelist.input* can be found in '/netdata/R1/data/parish/class/atsc5004/2014'. You will have to edit for the times of your simulation and if you have made changes to the number of grid points, etc. For your run, specify a time step of 90 seconds. In general, the maximum time step (in seconds) to specify is 6 times the horizontal grid spacing in kilometers. Thus, for our 15-km outer domain, 90 seconds is about the largest time step we can specify. If during your run you see massages regarding 'cfl errors', simply decrease your time steep and run again. In our case, if you get cfl errors, try a time step of 60 seconds. This will take longer, but you should not have numerical stability issues.

Before we can run WRF, we need to first link to the *met_em** grids we created in WPS. The link is simple. Just type:

ln –sf ../../../WPS/met_em* .

and you should see the familiar light blue color to signify the names of the linked files. To use the multiple processor option, we need to specify an environmental variable. For now, let's use eight of the 16 processors on bat. To do this, type:

setenv NCPUS 8

If you wanted to be really greedy, you could specify all 16 processors here. At the same time, preliminary tests indicate that running with 16 processors provides little (if any!) improvement from running with eight processors in terms of the time required to complete a WRF simulation. Besides, you will find yourself stepping on others running on bat at the same time!

The last step is to run the program *real.exe*. The work of vertically interpolating meteorological fields to WRF vertical levels is performed within the *real.exe* program and is the final step before running WRF. To run type: ./real.exe. After completion, you should see three files: *wrfinput_d01, wrfinput_d02 and wrfbdy_d01*. These are the files WRF needs to run. Once you have these, you are ready to run WRF!!

To run WRF, simply type

./wrf.exe >& wrf.log &

where the ampersand simply indicates that the run is in the background and you can then logoff bat and it will continue to run! You can check on your WRF performance by the 'top' command that lists processes running (a control-c will terminate the 'top' process). If you have specified NCPUS 8, the %CPU usage listed in 'top' for your WRF run should be around 800%. Try it!

Creating a GEMPAK file from WRF output

Here are the directions for conversion of WRF netcdf (.nc) files to .gem files. **Students that took ATSC5004 last semester are familiar with this, they can simply copy the 'convert.nl' file from their atsc5004 directory and then edit it**. *Make sure to include the non-divergent wind, fields 507 and 508.*

You should have compiled both WRFV3 and WPS. From WPS we should have run *geogrid.exe*, *ungrib.exe*, and *metgrid.*exe. Once we have the met_em* files, we can link them within the test/em_real subdirectory in the WRFV3 directory, run *real.exe* and then *wrf.exe.* After WRF has completed running, you should be informed by the last few lines of your WRF log file that the run was successful. For our class we have used a single domain and thus the output from WRF will be in one output file in the test/em_real subdirectory called 'wrfout_d01_2012-12-19_12:00:00'. This naming convention is straightforward with the domain, and initial time specified in the file name. WRF output files are in netCDF format. Remember that we can see what variables are contained in the netCDF file by the *ncdump* utility. For example, to see variables in the outer domain just use

```
ncdump –h wrfout_d1* > output_file
```

which will redirect the output to an output file that you can examine with any text editor. There are a host of applications that can be used to examine and manipulate WRF output. Among them are the NCAR Command Language or NCL, Unidata IDV, and GEMPAK. NCL and IDV are nice in that they can analyze the netCDF WRF files directly without additional processing. For our class, we will build on our previous experience in ATSC5004 and work with GEMPAK. To work in this environment, however, an additional step is required to convert from the WRF netCDF format to GEMPAK format. The program we will work with is *wrf2gem*, a program written by graduate student Steve Decker at the University of Wisconsin. A paper describing the *wrf2gem* program can be found at http://www.mmm.ucar.edu/wrf/users/workshops/WS2005/abstracts/Session3/44-Decker.pdf

For your WRF simulation, first create a directory called output within your main WRF directory and copy the file 'convert.nl' and the file README. The README file discusses the variables output from the program. In general, we can output variables on the native sigma coordinate system or interpolated to isobaric surfaces. Sigma levels are normalized heights, ranging in value between 1.0 at the surface to 0.0 at the top of the model domain, which a default value of 50 mb in your namelist (you may have changed this). Sigma coordinates are a common means to represent the effects of irregular terrain. The bulk of what we will do in this class will focus on the isobaric interpolation so that the model output can be analyzed in gempak.

First, we need to add two fields to the WRF output file, the zonal and meridional components of the non-divergent wind. To do this, run a script developed by Xia Chu, called **add_variable_to_wrf_output.ncl**. Follow these steps:

1. Copy add_variable_to_wrf_output.ncl from http://www.atmos.uwyo.edu/~geerts/atsc5160/ to your work directory on bat.
2. Open add_variable_to_wrf_output.ncl with gedit or vi or any other editors. Change the directory and file name (dir and filename) to match yours. Save and close the program.
   Note: I suggest that you back up your original wrf output files before you run add_variable_to_wrf_output.ncl.
3. To run the program, simply type in **ncl add_variable_to _wrf_output.ncl**.
4. If your screen shows you text as shown below, the new variables should be in your wrf output file now. You can double check it with ncdump or ncl_filedump:

```
ncdump -h wrfout_dXX_YYYY-MM-DD_HH:MM:00|grep UND
ncl_filedump wrfout_ dXX_YYYY-MM-DD_HH:MM:00.nc|grep VND
    Variable: und
    Type: float
    Total Size: 21168000 bytes
            5292000 values
```

Number of Dimensions: 4
Dimensions and sizes: [Time | **] x [bottom_top | **] x [south_north | **] x [west_east | **]
Coordinates:
Number Of Attributes: 4
units : m/s
short_name : u wind
long_name : Non-divergent zonal wind
_FillValue : 9.96921e+36

Variable: vnd
Type: float
Total Size: 21168000 bytes
5292000 values
Number of Dimensions: 4
Dimensions and sizes: [Time | **] x [bottom_top | **] x [south_north | **] x [west_east | **]
Coordinates:
Number Of Attributes: 4
units : m/s
short_name : v wind
long_name : Non-divergent meridional wind
_FillValue : 9.96921e+36

Next, you need to edit the file 'convert.nl'. It can be opened in any text editor. You will see the following variables in 'convert.nl'. Edit these as follows:

```
&OUTFILES
numFiles = 1 /
&WRFFILE
fName = "../WRFV3/test/em_real/wrfout_d01_2012-12-19_12:00:00" /
&GEMFILE
gFName = "./dec19_12z.gem",
maxGrd = 30000,
pack = .true.,
overwrite = .true. /
&NUMOUTS
num = 25 /
&VARS
outFields = 1, 11,12,13,30,101,103,109,110,111,112,114,121,123,124,125,138,501,502, 503,504,505,506, 507, 508
pb = 1000,
pt = 100,
dp = 25 /
```

where

- **fName**: the path and filename for your WRF output file. In the edited example above, the output from domain 1 is being used with a path that follows from the directory contained in the output website. Another option is to create a soft link to the WRF output file within the output directory. Then you can set fName = "wrfout_d01_2012-12-19_12:00:00" /.
- **gFName**: the name you will give for the GEMPAK file that is to be created (your choice).
- **maxGrd**: the maximum number of GEMPAK grids (leave as is)
- **pack**: packing information (leave as is)
- **overwrite**: whether to overwrite a file if it already exists (leave as is). In some instances, you may want to add additional grids later on to the same GEMPAK file. This will allow for that to happen. If you set overwrite=.false/ you will get an error message if you want to add additional grids to a preexisting GEMPAK file.

- **num**: the number of variables you wish to output (25 in this case). Make certain that 'num' is set to the same number of grids specified under 'outFields'. The 'outFields' command specifies which grids are to be created within the GEMPAK file. See list below. Details can be found in the README file.
- **pb**: lowest pressure level (mb)
- **pt**: highest pressure level (mb)
- **dp**: vertical resolution of isobaric levels in .gem file (25 mb is a good choice)

note: if you have a lot of grid points, convert.nl may give an error message about too large a file. It then aborts without converting all desired fields. You can reduce the .gem file by excluding several 3D variables in sigma coordinates, since we do not use them in the future lab assignments. You can make the list shorter as follows:
outFields = 1, 11,12,13,21,30,101,103,109,121,138,501,502, 503,504,505,506, 507, 508

outField identifiers:

1 - HGT: Terrain height (m)
2 - T2: 2-m temperature (K)
3 - Q2: 2-m water vapor mixing ratio (kg/kg)
4 - ZNU: Eta coordinate on mass (half) levels
5 - PB: Base state pressure (Pa)
6 - P: Perturbation pressure (Pa)
7 - T: Perturbation potential temperature (K)
8 - U: U component of wind (m/s)
9 - V: V component of wind (m/s)
10 - W: W component of wind (m/s)
11 - U10: U at 10 m (m/s)
12 - V10: V at 10 m (m/s)
13 - QVAPOR: Water vapor mixing ratio (kg/kg)
14 - QCLOUD: Cloud water mixing ratio (kg/kg)
15 - QICE: Ice mixing ratio (kg/kg)
16 - RAINC: Accumulated cumulus precip (mm)
17 - RAINNC: Accumulated grid-scale precip (mm)
18 - PH: Perturbation geopotential (m^2/s^2)
19 - PHB: Base-state geopotential (m^2/s^2)
20 - ZNW: Eta coordinate on w (full) levels
21 - LU_INDEX: Land use category
22 - MU: Perturbation dry air mass in column (Pa)
23 - MUB: Base-state dry air mass in column (Pa)
24 - TH2: 2-m potential temperature (K)
25 - QRAIN: Rain water mixing ratio (kg/kg)
26 - QSNOW: Snow mixing ratio (kg/kg)
27 - QGRAUP: Graupel mixing ratio (kg/kg)
28 - PBLH: PBL height (m)
29 - SST: Sea surface temperature (K)
30 - TSK: Surface skin temperature (K)
31 - SWDOWN: Downward short wave flux at ground surface (W/m^2)
32 - GLW: Downward long wave flux at ground surface (W/m^2)
33 - GRDFLX: Ground heat flux (W/m^2)
34 - HFX: Upward heat flux at the surface (W/m^2)
35 - QFX: Upward moisture flux at the surface (W/m^2)
36 - LH: Latent heat flux at the surface (W/m^2)
40 - UND: zonal component of the nondivergent wind (m/s)
41 - VND: zonal component of the nondivergent wind (m/s)

Diagnostics
101 - Surface pressure (hPa)
102 - Pressure at 2 m (hPa)

103 - Sea level pressure (hPa)
104 - Convective precipitation accumulated over one hour (mm)
105 - Total precipitation accumulated over one hour (mm)
106 - Total precipitation accumulated over three hours (mm)
107 - Total precipitation accumulated over six hours (mm)
108 - Total accumulated precipitation (mm)
109 - Pressure (hPa)
110 - Potential temperature (K)
111 - Unstaggered U component of wind (m/s)
112 - Unstaggered V component of wind (m/s)
113 - Cloud water + Ice mixing ratio (kg/kg)
114 - Unstaggered W component of wind (m/s)
115 - 2-m temperature including at initial time (K)
116 - 2-m water vapor mixing ratio including at initial time (kg/kg)
117 - U at 10 m including at initial time (m/s)
118 - V at 10 m including at initial time (m/s)
119 - Lifted Index using lowest 4 levels (K)
120 - Mean relative humidity (%) in 850-500-hPa layer
121 - Geopotential height (m) on w (full) levels
122 - 2-m specific humidity (kg/kg)
123 - Specific humidity (kg/kg)
124 - Temperature (K)
125 - Geopotential height (m) on half levels
126 - Dry air mass in column (Pa)
127 - Most unstable CAPE w/ virt. temp. correction (J/kg)
128 - Lifted parcel level of the most unstable parcel (m)
129 - Surface-based CAPE w/ virt. temp. correction (J/kg)
130 - Surface-based CIN w/ virt. temp. correction (J/kg)
131 - Mixed-layer CAPE w/ virt. temp. correction (J/kg)
132 - Mixed-layer CIN w/ virt. temp. correction (J/kg)
133 - Mixed-layer LCL (m)
134 - Mixed-layer LFC (m)
135 - U component of Bunkers storm motion (m/s)
136 - V component of Bunkers storm motion (m/s)
137 - 0-1-km storm relative helicity (m2/s2)
138 - Column integrated precipitable water (cm)
139 - Reflectivity (dBZ)
140 - Composite reflectivity (dBZ)

Diagnostics interpolated to isobaric levels
501 - Temperature (K)
502 - U component of wind (m/s)
503 - V component of wind (m/s)
504 - W component of wind (m/s)
505 - Geopotential height (m)
506 - Specific humidity (kg/kg)
507 - UND: zonal component of the nondivergent wind (m/s)
508 - VND: meridional component of the nondivergent wind (m/s)

pb -- Bottom pressure level for isobaric diagnostics
pt -- Top pressure level for isobaric diagnostics
dp -- Pressure interval for isobaric diagnostics

We can interpret the list such that numbers 1 through 140 are output on the native sigma surfaces. Variables from the 500 group are the interpolated isobaric variables. For the example line from above, we will output 23 variables.

From the sigma levels, we output terrain height (1), 10-m winds U (11) and V (12), water vapor (13), land use category (21) and surface skin temperature (30). These are directly output from WRF. We also output on sigma surfaces some "diagnostic" variables, computed from the WRF output. You can match the output from the list above. We also output the entire 500 group to isobaric surfaces. You will note that for our case we will need fields 501 (temperature), 502 (u), 503 (v), 504 (w), 505 (Z), 506 (specific humidity q), 507 & 508 (U and V components of the non-divergent wind), all at increments of 25 mb, and 2D fields 1 (terrain height), 106 (3 hour precip P3), and 107 (6 hour precip P6). You can output whatever variables you see appropriate. It is also possible to output additional variables. The README file discusses this briefly. Feel free to experiment.

Once you have edited the *convert.nl* program, save and close it. To run simply type *wrf2gem* and the program will start.

wrf2gem (not ./wrf2gem)

You will note that it informs you of the progress as the GEMPAK file is being created. Once completed, move this .gem file to your ATSC5007 folder for further analysis using gdplot2. The first thing to do is to check the contents of the GEMPAK field created using the 'gdinfo' command**.**