

WRF/WRF-Chem

- Glossary
- Super quick start (Harvard cluster users only, real case, no compilation)
 - One-time setup
 - Run WPS
 - Run WRF
- REAL cases
 - PART I. Setting up/Configuration/Compilation.
 - PART II. Running WPS and WRF: Overview/General Steps
 - Step 1: Running WPS (WRF Pre-processing System)
 - Step 2: Running WRF
- Detailed examples
 - WRF-Chem: A PM2.5 example
 - Step 0.1.
 - Step 0.2
 - Step 1. Run WPS
 - Step 2. Run real.exe to get necessary intermediate files for WRF-Chem utilities (chem_opt off).
 - Step 3. Generate your bio emissions using MEGAN
 - Step 4. OPTION 1 (USE THIS FOR NOW): Prep your anthropogenic emissions using anthro_emis
 - Step 4. OPTION 2 (FYI for now): Prep your anthropogenic emissions using prep-chem-sources
 - Step 5. Run real.exe again, with chem_opt turned on
 - Step 6. Prep the chemical data initial and boundary conditions using MOZBC
 - Step 7. You're (FINALLY) ready to run WRF-Chem
 - Step 8. Post-Processing and data visualization
 - Running WRF-Chem for real cases in Large Eddy Simulation (LES) Mode: A Beijing PM2.5 Case Study
- IDEALIZED cases
- Miscellaneous links

This page documents the steps to run WRF and WRF-Chem on Harvard cluster.

We invite Archana Dayalu, Packard Chan, Lee Miller and Jiahua Guo to co-edit this page.

Glossary

- **WRF**: Weather Research and Forecasting Model
- **ARW** / **EM**: Advanced Research WRF. As opposed to NMM.
- **NMM**: Nonhydrostatic Mesoscale Model
- **WPS**: WRF Preprocessing System
- **DA** / **VAR**: Variational Data Assimilation
- **KPP**: Kinetic Pre-Processor (related to chemistry)
- chem, ...
- real
- ideal
- grib: Gridded Binary
- grid

Super quick start (Harvard cluster users only, real case, no compilation)

One-time setup

Register in http://www2.mmm.ucar.edu/wrf/users/download/get_source.html

```
ssh datamover01 # for faster copying
```

Choose a wrf-root directory: recommend to be on lfs disk, but not scratchlfs. If available, LAB storage on holylfs, kuanglfs are good choices. Below uses /n/holylfs/LABS/id -gn`/\$USER/new-wrf-root/.

```
rsync -a /n/holylfs/INTERNAL_REPOS/CLIMATE_MODELS/wrf/410-WPS /n/holylfs/LABS/id -gn`/$USER/new-wrf-root/ --exclude='.git*' # 15s on datamover01
```

```
rsync -a /n/holylfs/INTERNAL_REPOS/CLIMATE_MODELS/wrf/412-WRF /n/holylfs/LABS/id -gn`/$USER/new-wrf-root/ --exclude='.git*' # 2m22s on datamover01
```

Run WPS

```
Exit datamover01 node. You can use login node for this default case.
cd /n/holyfls/LABS/id -gn`$USER/new-wrf-root/410-WPS/
source source_wrf # 25s on holylogin03
vim namelist.wps # no edits
./geogrid.exe # 8s on holylogin03, v410, Jan00
ln -sf ungrib/Variable_Tables/Vtable.GFS-PRMSL Vtable # for ungrib
```

```
./link_grib.csh /n/holyfls/INTERNAL_REPOS/CLIMATE_MODELS/wrf/data-tutorial-case/JAN00/fnl_2000012* # for ungrib
```

```
./ungrib.exe # 2s on holylogin03, v410, Jan00
./metgrid.exe # 1s on holylogin03, v410, Jan00
```

Run WRF

```
cd /n/holyfls/LABS/id -gn`$USER/new-wrf-root/412-WRF/
cp -ai run 00run # 5s on holylogin03
cd 00run/
# make sure you have sourced source_wrf
ln -s ../410-WPS/met_em.d01.2000-01-2* ./
vim namelist.input # no edits
./real.exe # 3s on holylogin03
tail rsl.error.0000 # expect to see "SUCCESS COMPLETE REAL_EM INIT"
vim job_wrf.sbatch # no required edits
sbatch job_wrf.sbatch # 2m36s with 4 huge_intel cpus
tail rsl.error.0000* # expect to see "SUCCESS COMPLETE WRF"
```

REAL cases

Choose versions of WRF & WPS:

[WRF v4.1 WPS v4.1](#)

Latest releases on official GitHub:

WRF v4.1.2 (precompiled without chem) is downloaded in /n/holyfls/INTERNAL_REPOS/CLIMATE_MODELS/wrf/412-WRF
WPS v4.1 (precompiled) is downloaded in /n/holyfls/INTERNAL_REPOS/CLIMATE_MODELS/wrf/410-WPS

WRF v3.6.1 (precompiled with chem) is downloaded in /n/holyfls/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/WRFV3
WPS v3.6.1 (precompiled) is downloaded in /n/holyfls/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/WPS

WRF v3.0beta with RAVE (pls recompile) is in ~kuang/Model/WRFV3
WPS v3.2.1 (pls recompile) is in ~dingma/Model2/WPSv3

Read the user's guide:

WRF (ARW) User's Guides: v3, v4

WRF-Chem: https://ruc.noaa.gov/wrf/wrf-chem/Users_guide.pdf #This is for a different WRF-Chem version (3.9), but it's still a relevant guide.
https://ruc.noaa.gov/wrf/wrf-chem/Emission_guide.pdf #This is a separate supplementary WRF-Chem guide to chemical input data processing.
https://ruc.noaa.gov/wrf/wrf-chem/wrf_tutorial_nepal/talks/Setup.pdf #Some helpful WRF-Chem slides from NOAA

PART I. Setting up/Configuration/Compilation.

- **Instructions in this part assume you want to compile and run your own version of WRF. However, note that a compiled usable version of WRF/WRF-Chem v3.6.1 including all external utilities and supplementary geography datasets that you can copy to your preferred run directory is already located at:**

```
/n/holyfls/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1
```

This folder (hereafter \$CLIMATE_MODELS) contains the WRF-ARW model, the WRF Pre-processing system (WPS; used for real test cases), the chemistry module add-on, the complete WRF geography dataset (for use with WPS and WRF-Chem), and other utilities needed for WRF-Chem. **Note:** WPS, WRF-Chem not relevant for idealized cases.

Note 2: With the exception of the geography data set which is really big, copy the WRF_CHEM_3-6-1 folder to a location you are going to run it from. Soft link to the geography data set in the \$CLIMATE_MODELS folder.

- **If you want to use another version, you will first need to register as a user to download WRF source codes**
WRF: http://www2.mmm.ucar.edu/wrf/users/download/get_source.html
- **Once you've registered, you will be able to navigate to the downloads section. Select the WRF components relevant to you. For example:**
<http://www2.mmm.ucar.edu/wrf/src/WPSV3.6.1.TAR.gz> #WPS 3.6.1
<http://www2.mmm.ucar.edu/wrf/src/WRFV3.6.1.TAR.gz> #WRF 3.6.1
<http://www2.mmm.ucar.edu/wrf/src/WRFV3-Chem-3.6.1.TAR.gz> #Chem module add-on for WRF 3.6.1
- If you're going to be using WRF meteorology output to drive the STILT LPDM (<http://stilt-model.org/index.php/Main/HomePage>) you will need to make some changes to the Registry file before compiling. Some basic instructions for this are located in \$CLIMATE_MODELS folder as a

README_WRF-STILT_modifications.txt file.

- Once you've downloaded the necessary tar.gz files, confirm your bashrc file looks something like:

```
# (1) Load required modules (here we use Intel and Intel MPI)

module load intel/17.0.4-fasrc01
module load impi/2017.2.174-fasrc01
module load netcdf/4.1.3-fasrc02
module load libpng/1.6.25-fasrc01
module load jasper/1.900.1-fasrc02
module load intel/17.0.4-fasrc01 impi/2017.2.174-fasrc01 ncview/2.1.7-fasrc01
module load ncl_ncarg/6.4.0-fasrc01

# (2) Define required environment variables

export NETCDF=${NETCDF_FORTRAN_HOME:-${NETCDF_HOME}}
export JASPERLIB=${JASPER_LIB}
export JASPERINC=${JASPER_INCLUDE}
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
export HDF5=${HDF5_HOME}
unset MPI_LIB #unset for WPS, where WPS is used for real WRF simulations

### ..... For WRF-Chem: ..... ###

export WRF_EM_CORE=1
export WRF_NMM_CORE=0
export WRF_CHEM=1
```

- Now, configure and compile. **WRF must be compiled before WPS.**

#(3) Configure WRF

```
./configure
```

#(4) Choose 15. (dmpar) to compile MPI version with Intel compilers

****Note! Do not use dm+sm or smpar options with WRF-Chem!! Choose either serial or dmpar****

#(5) Modify the file "configure.wrf" (around lines 149-150) to read the following. Note that you have to do this each time you run ./configure, because the configure.wrf script is overwritten each time.

```
DM_FC = mpiifort -f90=$(SFC)
DM_CC = mpiicc -cc=$(SCC) -DMPI2_SUPPORT
```

#(6) Compile WRF before WPS!! Compilation will take a while. If you're on an interactive shell, remove the "&" to avoid timing out.

For real cases:

```
./compile em_real &> compile.log
```

#(7) Configure WPS

#Likely you will need GRIB2 Support...so Choose option #19:

#Linux x86_64, Intel compiler (dmpar)

```
./configure
```

#(8) Compile WPS

```
./compile &> compile.output
```

#(1)-(8) are adapted from p.6 of [Plamen's advice](#) for v3.9.1 (6/8/2018)

- **Now that the main WRF and WPS programs are compiled, it's time to think about utilities specific WRF-Chem.** If you are planning on running WRF-Chem, you will additionally need to compile some utilities to make sure your chemical data is in the right format. Specifically, you will likely need to compile `convert_emiss.exe`; `prep-chem-src`, and `mozbc`. These are just examples; you may find that you may need additional ones.

#(9) Compile `convert_emiss.exe`, a program that converts binary intermediate chemical fields into netcdf WRF-compatible format. Navigate to your WRFV3 folder and type:

```
./compile_emi_conv
```

If compilation was successful, you should see `convert_emiss.exe` in the WRFV3/chem folder.

#(10) Compile PREP-CHEM-SOURCES (available [HERE](#)), a mapping utility for converting raw anthropogenic chemical fields to binary intermediates that are then fed into `convert_emiss.exe`. Unzip the tar.gz file into your main WRF folder. There are some typos, and missing details in the pdf guide above, so a modified version of the instructions (and Paul Edmon's help rebuilding HDF5 to fix an error message) enabled successful compilation of the utility. The modified instructions are located here:

```
/n/holy1fs/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/WRF/README_prepchem_modifications
```

In any case, a compiled version of prep chem sources utility using the instructions above is located here:

```
/n/holy1fs/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/WRF/PREP-CHEM-SRC-1.5/bin  
/prep_chem_sources_RADM_WRF_FIM_SIMPLE.exe
```

#(11) The `anthro_emiss` utility (ANTHRO). Like `prep-chem`, this is another (possibly less versatile) option for getting your **anthropogenic chemical fields** the `wrf-chem` format. It appears that it's useful for very specific data sets (like EDGAR-HTAP). But it also appears to be more straightforward to use, if you aren't rigid about your emissions inventory choice (i.e., if EDGAR-HTAP is sufficient for your purposes). If you want to set it up on your own, go here (<https://www.acom.ucar.edu/wrf-chem/download.shtml>) and click "anthro_emiss" at the bottom. Or you could use what is already downloaded and compiled (following the instructions in the README):

```
/n/holy1fs/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/WRF/ANTHRO
```

#(12) The MOZBC utility for mapping **chemical boundary conditions** to your WRF-Chem domain has already been compiled and saved in the \$CLIMATE_MODELS WRF-Chem folder, following the instructions in the `README_mozbc` file. You can use that, or if you wanted, [download and compile MOZBC](#) on your own. The initial files are based on MOZART 4-GEOS 5 output (6hr, 1.9x2.5deg, 56-level) datasets (<https://www.acom.ucar.edu/wrf-chem/mozart.shtml>). Read the README file for compilation instructions if you're doing this on your own; on the Harvard cluster you might have to do the following: `export NETCDF_DIR=$NETCDF_HOME` before compilation, and same with MEGAN instructions (#13 below). Otherwise you can use the compiled version located at:

```
/n/holy1fs/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/MOZBC
```

#(13) MEGAN **bio emissions**. A compiled version is located at the path below. (Read the README file for details if you want to compile your own version. Make sure your `NETCDF_DIR` variable is set if so, and make sure your `make_util` file is executable. Then run `./make_util megan_bio_emiss`). Make sure you also have the input data you need from <https://www.acom.ucar.edu/wrf-chem/download.shtml>. Scroll to the bottom "For bio_emiss input files only" and fill out the requested spatiotemporal information. Then select "bio_emiss input files".

```
/n/holy1fs/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/MEGAN
```

#(14) EDGAR-HTAP anthropogenic chemical fields, for use with `anthro_emiss` utility. Download from here: <https://www2.acom.ucar.edu/wrf-chem/wrf-chem-tools-community>. Otherwise you can link to the already downloaded version:

```
/n/holy1fs/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/anthro_data
```

PART II. Running WPS and WRF: Overview/General Steps

Now that you have a compiled version of WRF and WPS, you are ready to set up your model runs. Since these are for real cases, you need access to initialization data sets. You will need to figure out the initialization data set best suited to your domain and purposes. For the examples for China provided here, we are using GRIB2 NCEP FNL Operational Model Global Tropospheric Analyses, continuing from July 1999 (<http://dx.doi.org/10.5065/D6M043C6>). For these examples, the files are already downloaded. Instructions to link to them are noted where necessary.

Regardless of whether you are running WRF or WRF-Chem, it is important that you do the following first and in this order (detailed instructions follow, including in the examples in Part III and IV):

- (1) Run WPS (geogrid.exe, ungrib.exe, metgrid.exe) to create real data-based initialization files with of form met_em.d0X.YYYY-MM-DD_hh:mm:ss.nc
- (2) Run real.exe to generate input and boundary files of form wrfinput_d0*, wrfbdy_d01 (and optionally wrffdda_d0*) to initialize WRF model

If you are running WRF without chemistry, you can go ahead and run the main WRF model at this point. **If you are running WRF-Chem, this is the point at which you run your chemistry data prep program (i.e., prep-chem-src, anthro_emis, and/or convert_emiss) which requires wrfinput_d0* files to actually work. Once you have your correctly formatted chemical data (they should be in the form wrfchemi_00z_d01 and wrfchemi_12z_d01). Once you are done with this and have all your requisite chem data in the right format stored or linked in the WRFV3/test/em_real folder, you can run the wrf.exe model.**

Step 1: Running WPS (WRF Pre-processing System)

The pre-processing to create a real data-based initialization file. Read the README file in the parent WPS folder for a quick guide on what the programs do.

- **Navigate to your WRF/WPS folder**
- **Step 1a. PROGRAM GEOGRID: define simulation domains, interpolate terrestrial data to model grids**

1. Define model domains with geogrid. Edit the namelist.wps file, specifically:

```
&share (e.g., core='ARW', max_dom, io_form_geogrid=2, opt_output_from_geogrid_path='/n
/your_scratch_dir_path/GEOGRID_DUMP/')

&geogrid
```

2. View domain configuration, confirm it's correct.

```
cp util/plotgrids_new.ncl . ; ncl plotgrids.ncl
```

3. Run geogrid.

```
cd geogrid; ln -s GEOGRID.TBL.ARW_CHEM GEOGRID.TBL
mpirun -np 1 ./geogrid.exe
```

- **Step 1b. PROGRAM UNGRIB: get gribbed met data into usable format to ingest into WRF**

1. Examine GRIB data (most likely GRIB2).

```
./util/g1print.exe /your_GRIB1_data_path/filename | more
./util/g2print.exe /your_GRIB2_data_path/filename | more
```

2. Edit namelist.wps as needed.

```
&share (start date, end date, interval seconds=21600)

&ungrib (outformat='WPS', prefix='/n/your_scratch_dir_path/METEM_FILEDUMP/FILE')
```

#note: be aware of 90-day retention policy of scratch data. Make sure you've emptied any files from previous runs in the dump dirs.

3. link Vtable (it's like a Rosetta Stone) to appropriate GRIB data type to prepare for ungribbing. e.g., for GFS data. WRF has a bunch of Vtables already in the Variable_Table subfolder of ungrib, but sometimes you will need to do some sleuthing and download the one that you actually need into the Variable_Tables folder. For the examples that follow, this is exactly what we'll need to do.

```
ln -sf ungrib/Variable_Tables/Vtable.theVtableYouWantToUse Vtable
```

4. link the GRIB data that you are going to use

```
./link_grib.csh /your_GRIB_data_path/XXX*
```

#This should update GRIBFILE.AAA, .AAB, etc links in pwd.

5. Run ungrib

```
mpirun -np 1 ./ungrib.exe >& ungrib.output
```

#You should see FILE:YYYY-MM-DD_hh in your folder prescribed by "prefix"

- **Step 1c. PROGRAM METGRID: horizontally interpolate ungribbed met data onto geogrid domains**

1. Edit namelist.wps as needed.

```
&share (this should be exactly as you need it already)
&metgrid (fg_name = &ungrib_prefix; io_form_metgrid=2, outpath)
```

2. Run metgrid

```
mpirun -np 1 ./metgrid.exe
```

3. You should see met_em.d01(or d02 or d03....).YYYY-MM-DD_hh:mm:ss.nc files created. These are your WPS final output files that real.exe ingests.

Step 2. Running WRF

- **Navigate to your WRF/WRFV3/run folder**
- **Step 2b. PROGRAM REAL: real data initialization program to set up the model domain and initialize WRF**

1. First link your met files from metgrid.exe

```
cd run
ln -s /n/your_scratch_dir_path/METEM_FILEDUMP/met_em* .
```

2. Edit namelist.input file.

```
&time_control #obviously should match namelist.wps.
history_outname = 'the/path/where/you/want/final/output/files/and/filename_d<domain>_<date>'
&domains #obviously should match namelist.wps
#Customize other sections as needed.
```

#Particularly examine num_metgrid_levels and num_metgrid soil levels so they match with the input specified by the met_em* files. At command line:

```
ncdump -h met_em_d0XXXX.nc | more
num_metgrid_levels is typically in the header of the file
num_metgrid_soil_levels is close to the footer of the file
```

#Execute real.exe ; no real benefit to running as a parallel job so...

```
mpirun -np 1 ./real.exe >& run_real.log #make sure namelist.wrf chem_opt = 0 for this step for now.
```

3. Examine the run_real.log file. You should see SUCCESS EM_REAL INIT printed at the end.

4. Make sure you have the following files output from this step:

```
wrfinput_d0* (one for each domain)
wrfbdy_d01 (just for d01)
wrffdda_d01 (just for d01 since nudging is happening only in this domain)
```

- **This is the point at which you have some separate additional steps (in red font, below) if you are running WRF-Chem. This is because successfully running these utilities for your simulation domain(s) requires the wrfinput (and wrfbdy, if using mozbc) files from real.exe. If you are not running WRF-Chem, skip this subsection and continue with Step 10 below.**

5. Process gridded chemical data (biogenic, boundary conditions, anthropogenic) using specified utilities (e.g., MEGAN bio, prep-chem-src, mozbc). Note: there are other options, like fire emissions ... you need to figure out what is relevant to your question.

6. Make sure your binary emissions files are saved in WRFV3/test/em_real either as hard or soft links. They MUST be in this directory in some form.

7. Turn your chemistry back on (chem_opt=XXX) in the namelist.wrf file.

8. Run convert_emiss.exe to convert the chemistry data from binary intermediate to WRF input form

9. You should have your relevant chem data in WRFV3/test/em_real at this point, ready for ingestion by the wrf model. At this point, the key files input files that WRF-Chem expects in order to run successfully may now include, but is not limited, to the following:

```
wrfbiochemi_<domain> #if you planned to include biogenic chem
wrfchemi_00z_<domain> #if you planned to use anthropogenic chem, for two sets of time 00z and 12z
wrfchemi_12z_<domain>
wrfbdy_d01 #Boundary file, should include chemical boundary conditions from mozbc for example if you chose to go that route.
wrfinput_<domain> #your standard initialization file from real.exe
wrffdda_<domain> #for the domain that you're nudging in (if you are nudging).
```

10. If everything looks in order, run the wrf model. This is the only step that has significant benefit from running in parallel!

```
sbatch run_wrf.sh #Edit this script as needed; a template is provided.
```

11. You should now have netcdf files of format, stored in /n/your_scratch_dir_path/WRFOUT/. If you are planning to drive an LPDM like STILT with these met files, these need to eventually be converted to .arl format using WRF2ARL directory contents. This is something that will be treated in a separate page. It is highly recommended that you get familiar with NCL (and have access to ncview) for post-processing and visualization of the wrfout files.

```
wrfout_d0#_YYYY-MM-DD_hh:mm:ss
```

Detailed examples

WRF-Chem: A PM2.5 example

The purpose of this example is to take the general steps listed above and actually run a three nested domain WRF-Chem PM2.5 simulation over Beijing during the January 2013 severe haze event and compare with observations. We are going to run WRF-Chem for a total of 10 days from Jan 6 2013 00:00UTC to Jan 16 2013 00:00UTC. We establish 5 days for model spin-up such that the usable simulation time period is 5 days. Make sure you have a local copy of the /n/holyfys/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/WRF directory and contents. You don't need to copy the geography data set.

Thanks to Meng Gao at Harvard China Project and Thomas Lauvaux at PSU for help with this and for providing their WRF-Chem namelist templates for this example!

NOTE: Running "plain wrf" without chemistry is everything below except you would ignore steps 3 through 6. If you're not interested in WRF-Chem, and wanted to do a plain WRF tutorial there are much better ones online, or you could just follow along here and skip steps 3 to 6.

At the end of this example you will have learned how to:

- Get in the habit of using `ncview` for sanity-check visuals
- Use the dust map from the WPS high-res geographical data set (`namelist.input dust_opt=3`), although this is less important for this winter exercise.
- Use `prep-chem-sources` and `convert_emis` to format a gridded anthropogenic PM2.5 emissions file to the WRF forecast domain
- Use gridded chemical data for biogenic emissions and chemical boundary conditions
- Run WRF-Chem with PM2.5-radiation feedbacks to simulate PM2.5 concentrations in Beijing in January 2013.
- Extract relevant components from `wrfout netcdf` files and examine/visualize using `NCL`.

Step 0.1.

At any point where you want to check the contents of a netcdf file as a sanity-check use `ncview`! This is an excellent habit to develop. Just type

```
ncview your_file.nc
```

and navigate through the variables and panes and make sure things look realistic.

Step 0.2

For `wrf-chem`, it's good practice to create a folder for use with various external utilities that you link your intermediate `wrf` files to. This will become clearer, but for now make sure you have a directory in your `$CLIMATE_MODELS` location that's entitled "UTILINP".

Step 1. Run WPS

- Navigate to the WRF/WPS folder.
- First customize the WPS `namelist.wps` file such that it looks like the following. Note that with the exception of the `geog_data_path`, all path variables should point to a writable dump directory of your choice. It is recommended you write to `regal` or a similar scratch space you have access to; the dump files take up space and aren't needed again after the run so they don't have to be stored in a permanent location. Consult the WRF user guide for detailed explanation of the namelist variables.

```
&share
wrf_core = 'ARW',
max_dom = 3,
start_date = '2013-01-06_00:00:00','2013-01-06_00:00:00','2013-01-06_00:00:00'
end_date = '2013-01-16_00:00:00','2013-01-16_00:00:00','2013-01-16_00:00:00'
interval_seconds = 21600,
opt_output_from_geogrid_path = '/n/regal/wofsy_lab/adayalu/GEOGRID_DUMP/WRF01/'
io_form_geogrid = 2,

&geogrid
parent_id = 1, 1, 2,
parent_grid_ratio = 1, 3, 3,
i_parent_start = 1, 40, 17,
j_parent_start = 1, 26, 21,
e_we = 81, 49, 55,
e_sn = 57, 49, 55,
geog_data_res = 'usgs_30s+5m','usgs_30s+2m','usgs_30s+30s',
dx = 81000,
dy = 81000,
map_proj = 'lambert',
ref_lat = 35,
ref_lon = 110,
truelat1 = 30,
truelat2 = 60,
stand_lon = 116.397,
geog_data_path = '/n/holyllfs/INTERNAL_REPOS/CLIMATE_MODELS/WRF_CHEM_3-6-1/WRF_GEOG_COMPLETE/geog'
/

&ungrib
out_format = 'WPS',
prefix = '/n/regal/wofsy_lab/adayalu/METEM_FILEDUMP/WRF01/FILE',
/

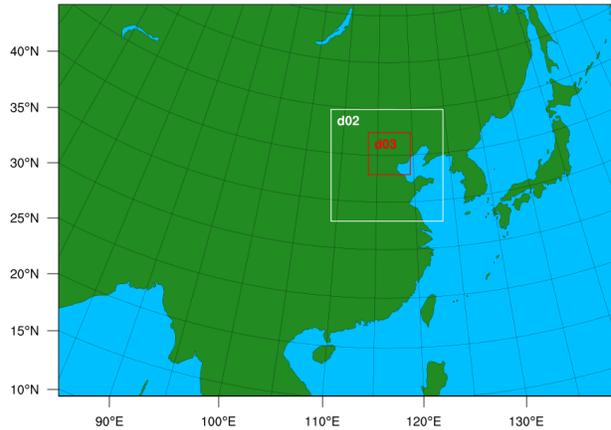
&metgrid
fg_name = '/n/regal/wofsy_lab/adayalu/METEM_FILEDUMP/WRF01/FILE'
io_form_metgrid = 2,
opt_output_from_metgrid_path = '/n/regal/wofsy_lab/adayalu/METEM_FILEDUMP/WRF01/'
/
```

- Type:

```
ncl plotgrids_new.ncl
```

- Your domain should look like the following.

WPS Domain Configuration



- Clear dump directories

```
rm -f /n/regal/wofsy_lab/adayalu/GEOGRID_DUMP/WRF01/*
rm -f /n/regal/wofsy_lab/adayalu/METEM_FILEDUMP/WRF01/*
```

- First call an interactive shell, then start running geogrid, ungrib, metgrid etc.

```
srun -n 1 --mem=10000 --pty --x11=first -p test -t 200 bash
cd geogrid; ln -s GEOGRID.TBL.ARW_CHEM GEOGRID.TBL
mpirun -np 1 ./geogrid.exe
```

And you should see something like below. Once this is done, navigate to your geogrid dump folder that you specified in your namelist.wps file and visualize the three geo_em.<domain>.nc files using ncview. Make sure things look reasonable.

```
Parsed 22 entries in GEOGRID.TBL
Processing domain 1 of 3
  Processing XLAT and XLONG
  Processing MAPFAC

[...etc for other variables, domains ...]

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Successful completion of geogrid. !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

- Now get ready to run ungrib. Since we will be using [NCEP fnl data](#) (GRIB2 format) for this run, we do some sleuthing to confirm the correct Vtable to use. Following the discussion [here](#) we download the [appropriate Vtable](#) separately (this one isn't included with the standard set of Vtables in WRFv3.6.1 because they recently updated the dataset). Note: I have already downloaded it and saved it: ungrib/Variable_Tables/Vtable.GFS_new

```
ln -sf ungrib/Variable_Tables/Vtable.GFS_new Vtable
./link_grib.csh /n/holylyfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130106* \
/n/holylyfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130107* \
/n/holylyfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130108* \
```

```
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130109* \  
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130110* \  
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130111* \  
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130112* \  
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130113* \  
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130114* \  
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130115* \  
/n/holylfs/LABS/kuang_lab/adayalu/WRF_GRIB2_NCEP_FNL/fnl_20130116*
```

- Examine a sample GRIB2 files using the g2print.exe tool

```
./util/g2print.exe /your_GRIB2_data_path/filename | more  
mpirun -np 1 ./ungrib.exe > ungrib.out
```

- Examine the end of the ungrib.out file and make sure you see the following. Also, check out the contents of the file to get an idea of what ungrid does.

```
tail -3 ungrib.out  
  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
! Successful completion of ungrid. !  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

- Run metgrid and make sure you get the "successful completion" message.

```
mpirun -np 1 ./metgrid.exe
```

Step 2. Run real.exe to get necessary intermediate files for WRF-Chem utilities (chem_opt off).

- Navigate to WRFV3/test/em_real
- Copy namelist.input_pm25Beijing to namelist.input. Take a look at the namelist and confirm the dates and domain match your wps namelist
- Make sure to turn off chemistry (chem_opt = 0) for this step.
- Change the filepath in history_outname to reflect where you want your wrfchem output files to be saved.
- Link the metgrid files to the test/em_real directory and make sure the namelist num_metgrid_levels and num_metgrid_soil_levels match what's in the metgrid files.

```
ln -s /n/your_scratch_dir_path/METEM_FILEDUMP/met_em* .
```

- Run real.exe to output the necessary intermediate files for chem data processing. Check that the tail of the file says SUCCESS COMPLETE REAL_EM INIT

```
mpirun -np 1 ./real.exe >& run_real.log
```

- You should have five files that real.exe produced. These currently only contain meteorological info, which is fine for running plain WRF-ARW. But – since we are running WRF-Chem – we need to run the additional utilities that either use the info contained in these files to map chemical data correctly (e.g., prep-chem-src) or modify the contents directly (e.g., mozbc for inputs of chemical boundary conditions into wrfbdy_d01).

```
wrfbdy_d01 #the outermost domain parameter boundary condition file  
wrfddda_d01 #nudging file, we requested nudging in the outer domain  
wrfinput_d<domain> #initial condition files for each of your study domains.
```

- Create soft links of all the met_em.<domain>, wrfinput_<domain>, wrfbd_d01 files in your UTILINP folder. These will be used by the external utilities in the next steps.

Step 3. Generate your bio emissions using MEGAN

- You will need access to the relevant MEGAN initial files, from <https://www.acom.ucar.edu/wrf-chem/download.shtml>
- Make sure you are still using an interactive shell ('srun -n 1 --mem=10000 --pty --x11=first -p test -t 200 bash' should be sufficient). Navigate to the MEGAN directory in the \$CLIMATE_MODELS directory (or wherever you have saved your local copy of everything)
- Create a new text file called megan_bio_emiss.inp. This is your MEGAN namelist file. Note that as the README instructs, the leaf area index (lai) months requires the simulation month and the previous month such that for January (as our example here is) we have to simulate all months. Following the instructions in the README, you should populate to look like follows ... with the paths obviously reflecting where your WRF + external utility directories are located.

```
&control
domains = 3,
start_lai_mnth = 1,
end_lai_mnth = 12,
wrf_dir = '/n/holy1fs/LABS/kuang_lab/adayalu/WRF/UTILINP',
megan_dir = '/n/holy1fs/LABS/kuang_lab/adayalu/WRF/MEGAN'
/
```

- You should be ready to run the MEGAN bio emission utility

```
./megan_bio_emiss < megan_bio_emiss.inp > megan_bio_emiss.out
```

- If the program ran correctly, you should get no messages and you should see three files in your MEGAN folder. Check them out with ncview.

```
wrfbiochemi_d01
wrfbiochemi_d02
wrfbiochemi_d03
```

Step 4. OPTION 1 (USE THIS FOR NOW): Prep your anthropogenic emissions using anthro_emis

- Now that you have your bio emissions, it's time to get your anthropogenic emissions in the right format. We're going to use the EDGAR-HTAP PM2.5 emissions on a 0.1x0.1 deg annual global grid.
- This option has been successfully tested if you're using EDGAR-HTAP emissions or IPCC emissions (for use with CAM-Chem) and seems a little less complicated than prep-chem-sources. In these examples, we will go this route. If you are starting with other emissions, you probably need to go the prep-chem route. I haven't spent much time figuring out how to use it; the anthro_emis option seemed most straightforward for now because I already had some familiarity with utilities in the same family (MEGAN, MOZBC).
- Navigate to your ANTHRO folder

```
cd src
```

- Link the EDGAR_HTAP_emi_PM2.5_2010.0.1x0.1.nc file from the anthro_data/MOZCART folder to this folder

```
ln -sf yourpath_toWRF/anthro_data/MOZCART/EDGAR_HTAP_emi_PM2.5_2010.0.1x0.1.nc .
```

- Create a new text file called anthro_emis.inp. This will be your namelist file. Check out the README file for detailed instructions. Your namelist file should look like the following. This is constructed based on a hybrid of following instructions in the README file and the input file located in the anthro_data/EDGAR-HTAP/INP-Examples folder. Make sure you understand the purpose of the entries.

```
&CONTROL
anthro_dir = '/n/holy1fs/LABS/kuang_lab/adayalu/WRF/ANTHRO/src'
domains = 3
wrf_dir = '/n/holy1fs/LABS/kuang_lab/adayalu/WRF/UTILINP'
src_file_prefix = 'EDGAR_HTAP_emi_'
src_file_suffix = '_2010.0.1x0.1.nc'
```

```
src_names = 'PM2.5(1)'  
sub_categories = 'emis_tot'  
serial_output = .false.  
start_output_time = '2010-12-01_00:00:00'  
emissions_zdim_stag = 1  
emis_map = 'PM_25(a)PM2.5',  
/
```

- Note that the `src_file_prefix` and `src_file_suffix` are concatenated with whatever is specified in `src_names` (here, PM2.5) to generate the full filename string.
- Now run `anthro_emis`.

```
./anthro_emis < anthro_emis.inp > anthro_emis.out
```

and you should see six new files in the ANTHRO/src directory, one for each of the three domains. Check out the contents with `ncview`.

```
wrfchemi_00z_<domain>  
wrfchemi_12z_<domain>
```

Step 4. OPTION 2 (FYI for now): Prep your anthropogenic emissions using prep-chem-sources

- TBD. Watch this space.
- Use this utility if you're using something other than EDGAR-HTAP or things not listed in the `anthro_emis` README/instructions. In theory, `anthro_emis` could work for other emissions fields, but they have just not been tested.
- Now that you have your bio emissions, it's time to get your anthropogenic emissions in the right format. Navigate to your PREP-CHEM-SRC-1.5 folder.

```
cd bin
```

- Edit the `prep_chem_sources.inp` namelist file. Check out the README file for detailed instructions.

Step 5. Run real.exe again, with chem_opt turned on

- Navigate to your `WRFV3/test/em_real/` directory
- Link your bio and anthro files to your `WRFV3/test/em_real/` directory.
- Open up the the `namelist.input` file and turn `chem_opt` back on (set it to 10).
- Make sure `kemit=1` (vertical levels in anthro emissions files...in this case it is 1...surface data).
- Call up an interactive shell if you don't have one running. Run `real.exe` again. This incorporates the chem variables into the initial and boundary condition files so that MOZBC can populate them.

```
mpirun -np 1 ./real.exe > run_real.log
```

- Check that the tail of the `rsl.error.0000` file says "SUCCESS COMPLETE REAL_EM INIT"

Step 6. Prep the chemical data initial and boundary conditions using MOZBC

- This step modifies the wrf initial and boundary condition files that now have space for chemical data (since we re-ran `real.exe` with `chem_opt` on and anthro/bio fields)
- Go to <https://www.acom.ucar.edu/wrf-chem/mozart.shtml> and submit a data request. This can take some time to process depending on your domain size request. Submit your bounding box, and times. For this exercise you can use what was already downloaded. This is available in the `mozbc` directory in the `$CLIMATE_MODELS` folder as a `mozart4geos5-ZZZZ.nc` file.
- Navigate to your `mozbc` folder from the `$CLIMATE_MODELS` folder.

```
cd MOZBC
```

- Recall the MOZBC utility is used to modify the `wrfinput/wrfbdy` files to contain chemical boundary conditions.
- From the `namelist.input` file for this exercise in `WRFV3/test/em_real`, we see that `chem_opt = 10`. Referring to the WRF-Chem user manual for more details. This chemistry option uses the CBMZ chemical mechanism and MOSAIC using 8 sectional aerosol bins. We match this with what MOZBC provides as sample namelists and we pick `CBMZ-MOSAIC_8bins.inp` as our `mozbc` namelist file. Copy this to a new file and edit the new file as follows

```
cp CBMZ-MOSAIC_8bins.inp mozbc.inp
```

- Now edit this file.

```
do_bc = .true.
do_ic = .true.
domain = 3

#FYI, I've found mozbc can be unhappy when the set directory path is too long.

dir_wrf = '/n/holy1fs/LABS/kuang_lab/adayalu/WRF/UTILINP/' #obviously this should be your
specific path.

dir_moz = '/n/holy1fs/LABS/kuang_lab/adayalu/WRF/MOZBC/' #obviously this should be your specific
path.

# fn_moz should look something like 'ha0004.nc'. #you will have to rename your mozart4geos5-ZZZZ.
nc file whatever this is.

#In the species map (spc_map) section, delete the entry 'op2 -> C2H5OOH'. This isn't in the mozart4geos5 file, and leads to an
error if it remains in there.

#(This knowledge is from trial and error.)
```

- Exit editor.

```
ln -s mozart4geos5-ZZZZ.nc ha0004.nc #copy as link to "rename"
```

- You are now ready to run mozbc. In your mozbc directory type:

```
./mozbc < mozbc.inp > mozbc.out
#tail mozbc.out should have a final line that reads:
bc_wrfchem completed successfully
```

Step 7. You're (FINALLY) ready to run WRF-Chem

- Navigate back to your WRFV3/test/em_real directory. Check that you have the preferred path set to your WRFOUT directory (recommend regal or some scratch space with a lot of space) in your history_outname field. *If you change the filename, note the only part of the filename that you should mess with is the part before '_d<domain>_d<date>'*
- I liked to keep my frames hourly (a file is written out every simulation hour. More helpful with debugging if run fails. Up to you.)
- Open up the script run_wrf.sh and make sure it looks ok.
- Submit your wrf job

```
sbatch run_wrf.sh
squeue -u username #monitor your job status
```

- If your run is successful you should start to see files populating your outpath:

```
cd /your/history_outname/specified/preferably/some/regal/directory/WRFOUT/
```

and you should see for each domain ...

```
wrfchem_d<domain>_2013-01-DD_HH:00:00
```

- With 15 cores on one node (using huce_amd queue) it took about 40 hours for this run to complete:

```
sacct -j 49862718 --format=JobID,JobName,MaxRSS,Elapsed
```

<i>JobID</i>	<i>JobName</i>	<i>MaxRSS</i>	<i>Elapsed</i>
49862718	wrfchem_t+		1-15:32:54
49862718.ba+	batch	11588K	1-15:32:54

Step 8. Post-Processing and data visualization

- I primarily use NCL, and/or some combination of NCL and R. Pick whatever you're used to for processing netcdf files. If you're going to use NCL, I recommend looking at the very well documented examples on their website. For example, start with the one about how to open and read netcdf files and go from there:

<https://www.ncl.ucar.edu/Document/Functions/Built-in/addfile.shtml>

- If you use ncview, you can quickly get a sense of how your simulation worked. Take a look at the PM2.5_DRY variable (3D Vars). You will notice that while the run completed successfully from a technical standpoint, it's actually way off. The PM2.5 values are unrealistic – two orders of magnitude lower than observations in the d03 domain – and this is most likely due to some combination of the following:
 - Accounting only for primary PM2.5. There is obviously all the secondary PM2.5 that needs the appropriate precursor species mapped as well. (25% to nearly 40% of PM2.5 in many cities in the region is secondary inorganic.)
 - My failure to process files correctly. While the wrfchemi, wrfbiochemi, and mozbc utilities seem to have gone through, it may not have done so correctly based on an inappropriate namelist parameter selection. For some reason, the surface emissions are not being read in correctly. We used EDGAR-HTAP from 2010 processed using the anthro_emis utility. In the past I have run a test of this using a more specialized inventory from 2010 pre-processed using NCL which led to a far more realistic PM2.5 simulation (i.e., surface emissions data was being read in).
 - inappropriate choices in the WRF-Chem namelist.input file.

In any case, this exercise should at least get you familiarity with the process of running WRF-Chem and set you up for being able to do second-order troubleshooting (like the more important question of why these values are unrealistic!).

Running WRF-Chem for real cases in Large Eddy Simulation (LES) Mode: A Beijing PM2.5 Case Study

IDEALIZED cases

WRF (ARW) User's Guides: [v3](#), [v4](#)

Miscellaneous links

WRF (ARW) User's Guides: [v3](#), [v4](#)

Google Docs: https://docs.google.com/document/d/1Jls4FIWIOlhMICzMPWm6_aBZqx_Axxe8RMckjdILDFg/

Ding's notes: [global_WRF_on_Odyssey.pdf](#)

ARW Technical Note: <http://www2.mmm.ucar.edu/wrf/users/docs/technote/>

Optimizing performance: <https://www2.cisl.ucar.edu/software/community-models/optimizing-wrf-performance>

Running on Cheyenne: <http://www2.mmm.ucar.edu/wrf/users/cheyenne-note.html>