WRFotron

Release 2.1.0

Nov 23, 2020

Contents

Tools to automatise WRF-Chem runs with re-initialised meteorology.

WRFotron created by Christoph Knote (christoph.knote@lmu.de).

User guide created by Luke Conibear (l.a.conibear@leeds.ac.uk).

Helpful additions from Helen Burns, Carly Reddington, Ben Silver, Laura Kiely, Thomas Thorp, Ailish Graham, Douglas Lowe, Scott Archer-Nicholls, and Edward Butt.

CHAPTER 1

Contents

1.1 Background

- National Center for Atmospheric Research (NCAR) Weather Research and Forecasting (WRF) Model Users.
- NCAR WRF with chemistry (WRFChem).
- National Oceanic and Atmospheric Administration (NOAA) WRFChem.
- WRF version 4.0 user guide.
- WRFChem version 3.9.1.1 user guide.
- WRFChem emissions version 3.9.1.1 user guide.
- WRF workshops, in person, UK.
- Past WRFChem presentations.
- WRFChem tutorials here.
- Fire INventory from NCAR (FINN) emission preparation.
- Guidance for Model for Ozone and Related Chemical Tracers (MOZART) gas scheme with Model for Simulating Aerosol Interactions and Chemistry (MOSAIC) aerosol scheme.
- Key papers:
  - Grell, G.A., et al. (2005). Fully coupled "online" chemistry within the WRF model. Atmos. Environ. 39, 69576975..
  - Fast, J. D., et al. (2006). Evolution of ozone, particulates, and aerosol direct radiative forcing in the vicinity of Houston using a fully coupled meteorology-chemistry-aerosol model. Journal of Geophysical Research: Atmospheres, 111(21), 129..
  - Skamarock, W. C., & Klemp, J. B. (2008). A time-split nonhydrostatic atmospheric model for weather research and forecasting applications. Journal of Computational Physics, 227(7), 34653485..
- Information regarding WRF Model citations.

- WRF public domain notice.

1.2 Before you start

- Register as a WRFChem user.
- Require basic knowledge of Linux (bash shell, grep, cut, sed, sort, awk, cat, scp) / FORTRAN (basics) / Python / high-performance computing (HPC, qsub) / NCO / NCL.
- Request account on your local HPC. For those at the University of Leeds visit Advanced Research Computing (ARC) site.

1.3 Compilation

1.3.1 CEMAC (recommended)

Pre-built executables

- WRFChem has been built with all compiler and MPI combinations on ARC4 here:

```
/nobackup/cemac/software/build/WRFChem/
```

- The pre-built executables are found in these locations:

```
/nobackup/cemac/software/apps/WRFChem/
```

Custom executables

- To build your own versions:
  - First create the empty build directories.
  - Then copy the `build.sh` and the WRFChem .tar.gz file for the version of choice (e.g. 4.2 below).
  - Then run the build script as below.
  - This copies over the code, builds everything, puts the executables in `software/apps/WRFChem/`, and hardlinks in the correct NetCDF libraries to avoid accidentally pointing to the wrong NetCDF libraries (e.g. from conda) through `/nobackup/WRFChem/build_scripts/linknchdf5.sh`.
  - When finished, update `WRFotron/config.bash` to direct to this new build.
  - Optional: Can then add any custom edits and manually recompile.

```
cd /nobackup/${USER}
mkdir -p software/build/WRFChem/4.2/1
mkdir -p software/build/WRFChem/4.2/src
cd software/build/WRFChem/4.2/1
cp /nobackup/WRFChem/build_scripts/4.2/build.sh .
cp /nobackup/cemac/software/build/WRFChem/4.2/src/WRFChem4.2.tar.gz ../src/
./build.sh
# update WRFotron/config.bash to point to this new build
```

- To build and use a custom preprocessor:

  - First copy over the default preprocessor code from `/nobackup/WRFChem` (e.g. anthro_emis).
  - Then copy over the makefile modifier to the same folder.
  - Then add your custom edits to the preprocessor.
  - Then create the custom preprocessor.
  - When finished, update `WRFotron/config.bash` to direct to this new custom preprocessor.

```
cd /nobackup/${USER}
cp -r /nobackup/WRFChem/anthro_emis .
cd anthro_emis
cp /nobackup/WRFChem/build_scripts/fix_makefile.sh .
./fix_makefile.sh
# make your custom edits
make_anthro
# update WRFotron/config.bash to point to this new processor
```

Misc

- To always be able to view and use all the software CEMAC has built when you run `module avail`, add the following lines to `.bashrc`:

```
if [ -r /nobackup/cemac/cemac.sh ] ; then
  . /nobackup/cemac/cemac.sh
fi
```

- The recommended compiler and MPI combination is:

```
compiler: intel
mpi: openmpi
```

- IntelMPI on ARC4 is not optimized and contains a bug. Run the following command to run smoothly with IntelMPI:

```
export I_MPI_HYDRA_TOPOLIB=ipl
```

1.3.2 Manual (alternative)

Setup

- Download WRFotron, WRFChem, make a copy for WRFMeteo without the chemistry folder, download WPS, download WPS Geography files:

```
cd /nobackup/${USER}
git clone https://github.com/wrfchem-leeds/WRFotron.git
git clone https://github.com/wrf-model/WRF.git WRFChem
git clone https://github.com/wrf-model/WPS.git

cp -r WRFChem WRFMeteo
cd WRFMeteo
rm -rf chem
```

- Or copy these folders over from `/nobackup/WRFChem`:

```
cd /nobackup/${USER}
cp -r /nobackup/WRFChem/{WRFotron,WRFChem4.2,WRFMeteo4.2,WPS4.2,WPSGeog4} .
```

- You will need to remove, or at minimum, change the module load line at the top of `config.bash`. The modules intel, openmpi, and WRFchem are for the CEMAC installation, and keeping these (and potentially others) can interfere with executables. These need to be removed. NCL, NCO, and conda can be used from CEMAC for manual runs, or you could have your own personal conda environments with NCL and NCO (see below). You can see the manual blueprint in the repository: config.bash.blueprint_manual.

```
# cemac compilation uses
module load intel/19.0.4 openmpi/3.1.4 WRFchem/4.2 ncl/6.5.0 nco/4.6.0 wrfchemconda/3.
↪7 sge

# for manual compilation remove (at least) intel, openmpi, and WRFchem
module load ncl/6.5.0 nco/4.6.0 wrfchemconda/3.7 sge
```

- The executables within `pre.bash` need to be copied over directly, rather than just linked which is adequate for the CEMAC method. To do this make both of the following replacements. You can see the manual blueprint in the repository: pre.bash.blueprint_manual.

```
# on line 21, replace:
for aFile in util geogrid ungrib link_grib.csh metgrid
# with:
for aFile in util geogrid geogrid.exe ungrib ungrib.exe link_grib.csh metgrid metgrid.
↪exe

# and then on line 80, replace:
cp -r ${WRFdir}/run/* .
# with:
cp -r ${WRFdir}/run/* .
rm *.exe
cp -r ${WRFdir}/main/*.exe .
cp -r ${WRFmeteodir}/main/wrf.exe wrfmeteo.exe
```

- All executables and preprocessors will need to have `./` before them to execute. This includes `ungrib.exe`, `geogrid.exe`, `metgrid.exe`, `real.exe`, `megan_bio_emiss`, `wesely`, `exo_coldens`, `anthro_emiss`, `fire_emis`, and `mozbc` in `pre.bash`. Also, `wrfmeteo.exe` and `wrf.exe` in `main.bash`. Also, `wrf.exe` in `main_restart.bash`. You can see the manual blueprints in the repository: pre.bash.blueprint_manual, main.bash.blueprint_manual, and main_restart.bash.blueprint_manual.
- Add links to the preprocessor executables `anthro_emis`, `fire_emis`, and `mozbc` by adding the following code. You can see the manual blueprints in the repository: pre.bash.blueprint_manual.

```
ln -s ${WRFanthrodir}/anthro_emis . # section 4.a, line 148
ln -s ${WRFfiredir}/fire_emis .     # section 4.b, line 164
ln -s ${WRFMOZARTdir}/mozbc .         # section 6,   line 186
```

- Download flex (tool for generating scanners: programs which recognize lexical patterns in text).
- **Download and compile (in serial) preprocessors:**
  - anthro_emis (anthropogenic emissions preprocessor).
  - fire_emiss (fire emissions preprocessor).
  - megan (biogenic emissions preprocessor).
  - mozbc (preprocessor for lateral boundary and initial conditions).
  - wes-coldens (exocoldens and season_wesely, O_2 and O_3 column densities and dry deposition).

  - Check preprocessors have the correct modules and libraries linked via: `ldd preprocessor`.

```
conda deactivate
module purge
module load intel netcdf
export NETCDF=$(nc-config --prefix)
export NETCDF_DIR=$NETCDF
export FC=ifort

./make_anthro

./make_fire_emis

./make_util megan_bio_emiss

./make_mozbc

./make_util wesely

./make_util exo_coldens
```

Conda

- Download the latest miniconda:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

- Run bash script, read terms, and set path:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

- Create conda environment with Python 3 (with some libraries for analysis), NCL, and NCO:

```
conda create -n python3_ncl_nco -c conda-forge -c oggm xarray salem xesmf numpy scipy␣
↪pandas matplotlib rasterio affine ncl nco wrf-python
```

- To activate/deactivate conda environment:

```
conda activate python3_ncl_nco
conda deactivate
```

- For more information on conda, visit.
- Create separate environments for downloading ECMWF data (requires Python 2) and ncview, which you can then load temporarily to execute these functions:

```
conda create -n python2_ecmwf -c conda-forge ecmwf-api-client
conda create -n ncview -c eumetsat -c conda-forge ncview libpng
```

Compile WPS, WRFMeteo, and WRFChem

- Modules:

```
conda deactivate
module unload conda
module unload openmpi
module load intel
module load intelmpi
module load netcdf
```

- Environment variables:

```
export FC=ifort
export NETCDF=$(nc-config --prefix)
export NETCDF_DIR=$NETCDF
export YACC='/usr/bin/yacc -d'
export FLEX_LIB_DIR='/nobackup/username/flex/lib'
export LD_LIBRARY_PATH=$FLEX_LIB_DIR:$LD_LIBRARY_PATH
export JASPERLIB=/usr/lib64
export JASPERINC=/usr/include

export WRF_EM_CORE=1 # selects the ARW core
export WRF_NMM_CORE=0 # ensures that the NMM core is deselected
export WRF_CHEM=1 # selects the WRFChem module
export WRF_KPP=1 # turns on Kinetic Pre-Processing (KPP)
export WRFIO_NCD_LARGE_FILE_SUPPORT=1 # supports large wrfout files
```

- WRFChem compilation:

```
cd /nobackup/username/WRFChem
./clean -a
./configure
```

- HPC option will be specific to your HPC architecture.
- ARC4 = 15 = INTEL (ifort/icc) (dmpar) e.g. Distributed-Memory Parallelism MPI.
- Compile for basic nesting: option 1.
- Compile real (as oppose to ideal simulations).
- Thousands of messages will appear. Compilation takes about 20-30 minutes.

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

- How do you know your compilation was successful?
  - If you have `main/*.exe`.
- Check the executables have all relevant linked libraries:

```
ldd main/wrf.exe
```

- WPS compilation (requires a successfully compiled WRF):

```
cd /nobackup/username/WPS
./clean -a
./configure
```

- HPC option will be specific to your HPC architecture.
- ARC4 = 17 = INTEL (ifort/icc) (serial).
- Sometimes configure.wps can assign the incorrect path to WRFChem, check and edit if required:

```
gedit configure.wps
WRF_DIR="/nobackup/${USER}/WRFChem"

./compile >& log.compile
```

- How do you know your compilation was successful?
  - If you have geogrid.exe, metgrid.exe, and ungrib.exe.
- Check the executables have all relevant linked libraries:

```
ldd geogrid.exe
```

- WRFMeteo compilation:
  - Deselect the WRFChem module

```
export WRF_CHEM=0

cd /nobackup/username/WRFMeteo
./clean -a
./configure
```

- HPC option will be specific to your HPC architecture.
- ARC4 = 15 = INTEL (ifort/icc) (dmpar).
- Compile for basic nesting: option 1.
- Compile real (as oppose to ideal simulations).
- Thousands of messages will appear. Compilation takes about 20-30 minutes.

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

- Check have `main/*.exe`.
- Check the executables have all relevant linked libraries:

```
ldd main/real.exe
```

- If make any changes to pre-processor settings then require a fresh re-compile.
- Also check if preprocessor requires a different module version that currently compiled with.
- Run above environment variables to get NetCDF.
- Add `-lnetcdff` to Makefile.
- Note for wes_coldens: FC hardcoded in `make_util`.
- Downloaded tools from here.
- If need JASPER:

```
wget http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/jasper-1.
↪900.1.tar.gz
tar xvfz jasper-1.900.1.tar.gz
./configure
make
make install
export JASPERLIB=/usr/lib64 # not installed need own jasper
export JASPERINC=/usr/include
```

- If need FLEX:

```
cd /nobackup/${USER}/flex/lib
./configure --prefix=$(pwd)/../flex
export FLEX_LIB_DIR='/nobackup/${USER}/flex/lib'
```

1.4 Automatic simulation

1.4.1 Running

WRFotron uses pre-built executables on ARC4 from CEMAC (for University of Leeds users). Everything required is loaded in `config.bash`, including Python, NCO, NCL, WPS, WRFMeteo, WRFChem, preprocessors, and ncview.

1. Log into ARC4, clone the WRFotron repo, and edit the `chainDir` path within `config.bash` if it is not `/nobackup/${USER}/WRFotron`:

```
git clone https://github.com/wrfchem-leeds/WRFotron.git
```

2. Load the availability of CEMAC modules. If have other modules loaded then unload them (`module purge`), and similarly deactivate conda (`conda deactivate`), as both of these can cause conflits.:

```
. /nobackup/cemac/cemac.sh
```

3. From within the WRFotron folder run `master.bash`:

```
. master.bash 2016 10 12 00 24 06
```

For users that require their own executables or that are from outside of the University of Leeds, you can manually compile them using the instructions here.

1.4.2 How it works

- WRFotron is used by calling the `master.bash` bash script. `master.bash` takes a starting date, a run time, a spinup time, and (optionally) a previous run's job ID on the queuing system as arguments.
- According to the run configuration in `config.bash`, WRFotron then prepares a run directory (typically on scratch or nobackup) with all necessary data, and submits 3 jobs to the queue:
  - Preprocessing script (`pre.bash`), containing calls to ungrib.exe, metgrid.exe, real.exe and the preprocessor tools for chemistry.
  - Main execution script (`main.bash`), which does the actual wrf.exe runs (spinup and chemistry run).
  - Postprocessing script (`post.bash`), which can be extended to do any kind of postprocessing on the wrfout files of the WRFChem run.
- Calling `master.bash` and giving it the job ID of another WRFotron `main.bash` process in the queue will tell the `main.bash` script to wait for this process to end before starting, thereby allowing you to submit several runs in a row at the same time, each of them restarting using the result of the previous run.
- The re-initialisation of meteorology works as follows:
  - After each successful WRFotron WRFChem run, wrfrst restart files are saved in a common directory. When a new run is called using `master.bash`, a meteo-only spinup run is made first, and a restart file is created at its end, now containing only "fresh" meteorology variables. It is then checked whether a restart file (with chemistry) from a previous run exists in the common restart directory. If this is the case,

only the chemistry variables are copied from the previous run's restart file to the meteo spinup restart file. Then, a WRFChem run is started using this combined restart file as initial conditions, thereby using "fresh" meteorology while carrying on chemistry variables across runs. In case no restart file is found, a "cold start" chemistry run is conducted, starting with MOZART global model forecast values as initial conditions.

- If `main.bash` breaks in the middle of the simulation, can restart using `main_restart.bash`:
  - Edited to not repeat the meteo spin up and carry on from where chem wrf.exe stopped.
  - Steps:
    - Go to the run folder where `main.bash` stopped
    - Copy the latest restart file with 00 hours over to the restart/base directory
    - Edit `main_restart.bash`:
      - Change newRunRstFile to this latest restart file
      - Change submission time length appropriately
      - Change lastRstFile to the final restart file date at the end of run
      - Change curDate to the first wrfout file date
    - Edit `namelist.input`:
      - Ensure `restart = .true.`
      - Change start date to match date of newRunRstFile
    - `qsub main_restart.bash`
    - When finished:
      - Manually copy over the final restart file to restart/base
      - Manually move the wrfout files to run/base/staging
      - Manually `qsub post.bash`
- Other files within WRFotron:
  - `postprocessing.py`.
    - Calculates AOD for 550nm through interpolations and just extracting for the surface.
    - Converts units of aerosols at a certain standard temperature and pressure by dividing by the inverse of density: μg/kg of dry air to μg/m^3 by dividing by m^3/kg.
  - WRFChem namelists (read `/WRFChem/run/README.namelist` or user guide for detailed information).
    - `namelist.chem`.
    - `namelist.wrf`.
    - `namelist.wps`.
  - `Vtable.ECMWF/GFS`.
    - Variable table for the intial and boundary meteorological conditions.
  - preprocessor input files (`emis_edgarhtap2_mozmos.inp`, `exo_coldens.inp`, `fire_emis.mozm.inp`, `mozbc.inp`, `megan_bio_emiss.inp`, `mozbc.inp.blueprint_201_mz4`, `mozbc.inp.blueprint_202_mz4`).

  - For files which depend on the aerosol / chemistry schemes (mozbc.inp, namelist.chem, and namelist.wrf), there are blueprints of each of these files for both the mozart_mosaic_4bin (`chem_opt = 201`) and the mozart_mosaic_4bin_aq (`chem_opt = 202`). See document.
    * Replace the contents of the namelist with the blueprint_201 / 202 version.

- Crontab script.
  - Not normally allowed, check with HPC staff first.
  - Touches all files in `/nobackup/${USER}` to update their date and stop them getting deleted.
  - Create a hidden file in home directory (`vi ~/.not_expire.sh`) and add to it triples of lines such as:
    * Touch -h makes sure symlinks don't expire too.
    * This script will change the last accessed date for all the specified directories and files underneath that path.
    * Change permissions 755 on .not_expire.sh (`chmod 755 ~/.not_expire.sh`).
    * Use the crontab command to edit the crontab file `crontab -e`
    * Then add a line: `0 4 4 * * ~/.not_expire.sh`
    * This has now set a cronjob to run that will automatically touch (and thus reset last accessed time) the files once a month at 0400 on the 4th of the month.
    * Runs on the login nodes

```
cd /nobackup/${USER}
find . -exec touch -ah {} \;
find . -exec touch -a {} \;
```

- Simulation folder layout automatically created by WRFChem:
  - Output/Base/ (NetCDF files for wrfout).
  - Restart/Base/ (Restart files for simulation runs).
  - Run/Base/Folder per simulation run/ (Everything gets created in here, specific to run).
  - Run/Base/Staging (wrfout files are stored for post-processing).
- Acquire meteorological NCEP GFS files.
  - Will have to change all scripts with dataDir locations to the correct ${USER}.

```
cd /nobackup/${USER}/download_and_find_gfs_mz4
get_GFS_analysis_2004-current.bash
get_GFS_analysis_parallel.bash
```

  - If these have a size of 0, use FNL analysis files at lower resolution.
    * The Globus Transfer Service (GridFTP) option to transfer the FNL files from the RDA.
    * The other option is to go to that link, click data access, click web file listing for either GRIB1 (pre 2007.12.06) or GRIB2 (post 2007.12.06), click complete file list, click on the year of interest within the group ID column and checkbox the timeframe you're interested in. Now either click csh download script and follow the instructions in the comments of the script (remembering to change your linux shell to csh), or click get as a tar file (though this is limited to 2GB), or again there is the option for globus.
    * To download for more than 1 day at a time. First changing the script to the time frame required, ensuring download for the spin-up timeframe too.

  - Go over GFS folder to check have 8 files per day for each day of simulation.

```
.find_missing_GFS.bash
qsub find_missing_GFS_parallel.bash
```

  - Rename FNL files to original GFS naming convention and copy for 3 hourly interval midpoints.

- Acquire MOZART (MZ4) files for chemical initial and boundary conditions.
  - Pre-2018:
    * Download MZ4.
    * Download CAM-Chem.
  - Post-2018:
    * Download WACCM.
  - Ensure for a month have day either side of time frame of interest, and go for global domain.
- Emissions.
  - Choose anthropogenic input namelist setting in config.bash.

```
cd /nobackup/${USER}/WRFotron
vi emis_edgarhtap2_mozmos.inp
```

  - Fire emissions (FINN).
    * Update `fire_emis.mozm.inp` to have to correct filename for the year of simulation careful to update file for the correct chemical mechanism.
- `config.bash`.
  - Check all directories are correct.
  - Change where WRFChem will run.
  - Keep the same name for synchronous runs.
  - Or if a new simulation, change.
    * workDir / achiveRootDir / restartRootDir.
- Check `pre.bash`.
  - Check the linked MZ4 files are for timeframe required e.g. 2015.
  - If using daily files, use this portion of code and comment out the monthly section.
  - Vice versa for if using monthly files.
- `namelist.wps.blueprint`.
  - Change domain, resolution, map projection, and map area.
  - Edit `namelist.wps.domain_test` to try out different domain settings.
  - Create domain plot `ncl plotgrids.ncl`.
  - View the PDF of the domain `evince wps_show_dom.pdf`.
  - When decided update setting in `namelist.wps.blueprint`.
- `namelist.wrf.blueprint`.
  - Change domain, resolution, and number of levels.

- `namelist.chem.blueprint`.
  - Change chemistry options.
  - See WRFChem User Guide.
- `master.bash`.
  - Calling master.bash without arguments gives you usage instructions:

```
. master.bash

$ Call with arguments <year (YYYY)> <month (MM)> <day (DD)> <hour (hh)> ..
↪.
$                  or <year (YYYY)> <month (MM)> <day (DD)> <hour (hh)> ...
$ possible options (have to go before arguments):
$                   -e <experiment name>
$                   -c <chain directory (submission from CRON)>
```

  - Master.bash submits `pre.bash`, `main.bash`, and `post.bash`.
  - Creates output, restart, and run directories on /nobackup/${USER}.
    - /run/base/startdate_enddate.
  - In this folder is all the files copied over with the settings updated in all the bash scripts (master, pre, main, post, config).
  - Test run for 24 hours.

```
. master.bash 2016 10 05 00 24 06
```

  - Start year / start month / start day / start hour (UTC time) / simulation length / spin up length.
  - Spin-up runs from 2016-10-04_18:00:00 to 2016-10-05_00:00:00.
  - Simulation runs from 2016-10-05_00:00:00 to 2016-10-06_00:00:00.
  - Check linked files were for this ${USER}.
  - Now make another run starting when the first one finishes, which will use the output of the previous run for chemistry initial conditions (rather than MOZART chemical boundary conditions), while re-initialising meteorology (from GFS/ECMWF data):

```
. master.bash 2016 10 05 00 24 06 999999
```

  - The 999999 is the job id for the `main.bash` from the previous run. This is used in the syntax to tell the HPC machine to wait until this job has finished before starting the new run. This is because the new run uses the files created from the first run. This allows you to submit several runs in a row at the same time, each of them restarting using the result of the previous run.
  - Four-dimension data assimilation (FDDA, i.e. re-initialisation of meteorology).

```
vi master.bash
S/GRIDFDDA/0/g # to turn it off
S/GRIDFDDA/1/g # to turn it on
```

  - Nudges horizontal and vertical wind, potential temperature and water vapor mixing ratio to analyses. It doesn't take the analyses fields for its values like some other models do. It uses them as initial conditions and then uses the primitive atmospheric equations. This is not for chemistry directly, though affects chemicals through transport.

  - After each successful WRFotron run, wrfrst restart files are saved in the restart directory. When a new run is called using `master.bash`, a meteo-only spinup run is made first, and a restart file is created at its end, now containing only "fresh" meteorology variables. It is then checked whether a restart file (with chemistry) from a previous run exists in the common restart directory. If this is the case, only the chemistry variables are copied from the previous run's restart file to the meteo spinup restart file. Then, a WRFChem run is started using this combined restart file as initial conditions, thereby using "fresh" meteorology while carrying on chemistry variables across runs. In case no restart file is found, a "cold start" chemistry run is conducted, starting with MOZART global model forecast values as initial conditions.

- If need to re-submit any parts of the simulation, from within the folder, make changes to the relevant bash script and then:

```
qsub pre.bash
qsub main.bash
qsub post.bash
```

- Approximate job run times and HPC requirements:
  - 1 day simulation takes 1 hour wall clock time approximately.
  - 1 month simulation takes 2 days wall clock time approximately.
  - 1 year simulations takes 1 month wall clock time approximately.
  - `pre.bash` = 2 hours, 1 core, 12GB/process (run in serial).
  - `main.bash` = 48 hours, 64 cores, 2GB/process (run in parallel).
  - `post.bash` = 48 hours, 4 cores, 12GB/process (run in parallel).
- Analyse output using Python. For those new to Python, I have a introductory course.

1.5 Manual Simulation

Independently run a 24 hour simulation for India from 2016 10 05.

- Check you have the GFS data you need for the dates required to initialise and force meteorological conditions (1 file per 3 hours, 8 files per day, none are too small):

```
cd /nobackup/${USER}
mkdir initial_boundary_meteo_gfs
cd initial_boundary_meteo_gfs
cp /nobackup/WRFChem/initial_boundary_meteo_gfs/GF201610{04..07}* .
```

- If require more GFS data, can copy more over from `/nobackup/WRFChem/initial_boundary_meteo_gfs` or can use the download scripts `get_GFS_analysis_2004-current.bash` and `get_GFS_analysis_parallel.bash` within `cd /nobackup/WRFChem/download_and_find_gfs_mz4`.
- Create a test run folder for the manual run of WRF:

```
cd /nobackup/${USER}/
mkdir testrun
```

- Copy `link_grib.csh` to the new folder.

```
cd /nobackup/${USER}/testrun
cp /nobackup/WRFChem/testrun_files/link_grib.csh .
```

- Link the required GFS data via `link_grib.csh` in to the new simulation folder.

```
./link_grib.csh /nobackup/${USER}/initial_boundary_meteo_gfs/GF201610*
```

- Copy over the ungrib, geogrid and metgrid folders.

```
cp -r /nobackup/WRFChem/testrun_files/ungrib .
cp -r /nobackup/WRFChem/testrun_files/geogrid .
cp -r /nobackup/WRFChem/testrun_files/metgrid .
```

- Link the ungrib, geogrid and metgrid executables from the folders that are now copied over.

```
ln -sf metgrid/src/metgrid.exe
ln -sf geogrid/src/geogrid.exe
ln -sf ungrib/src/ungrib.exe
```

- Copy over the WPS and input namelists.

```
cp /nobackup/WRFChem/testrun_files/namelist.wps .
cp /nobackup/WRFChem/testrun_files/namelist.input .
```

- Link to the variables table.
- If post-2015 simulation, use new variable table:

```
ln -sf /nobackup/WRFChem/Vtable.GFS_new Vtable
```

- If pre-2015 simulation, use old variable table:

```
ln -sf /nobackup/WRFChem/Vtable.GFS Vtable
```

- Copy over the WRF and real executables, and the WRF and real bash scripts for job submission.

```
cp /nobackup/WRFChem/testrun_files/real.exe .
cp /nobackup/WRFChem/testrun_files/real.bash .
cp /nobackup/WRFChem/testrun_files/wrf.exe .
cp /nobackup/WRFChem/testrun_files/wrf.bash .
```

- Edit the time for the run on the WPS namelist according to the new requirements for the simulation. Be careful for leap years, and any changes made in the WPS namelist have to mirrored if the same variables are present in the input namelist.
  - start_date = '2016-10-05_00:00:00'.
  - end_date = '2016-10-06_00:00:00'.
  - number of domains (use 1).
  - spatial resolution (dx and dy).
  - map projection (i.e. Lambert conformal, Mercator, polar stereographic, or Regular latitude-longitude also known as cylindrical equidistant).
  - If lambert, dx and dy are in metres.
  - Uses projection parameters: truelat1, truelat2, stand_lon.
  - See page 37 of WRF User Guide.
  - Update and edit the namelist.input.

  - make sure the run_hours, start date, end date, timestep, e_we, e_sn, dx, dy are the same here as they are in the namelist.wps.
  - time step for integration seconds (recommended 6*dx in km for a typical case).

- Load the netCDF module.

```
module load netcdf
export NETCDF=$(nc-config --prefix)
export NETCDF_DIR=$NETCDF
```

- Run geogrid.

```
./geogrid.exe
```

- Configures the horizontal domain, interpolating static geographical data.
  - Creates geography (`geo_em.d01.nc`) for each domain.
  - Progress logged in geogrid.log.
- Run ungrib.

```
./ungrib.exe
```

- Reads, reformats, and extracts meteo input data.
  - Creates meteorology by ungribbing the GFS grb2 files.
  - Intermediate files for every time step.
  - Progress logged in ungrib.log.
- Run metgrid.

```
./metgrid.exe
```

- Ingests and interpolates input data creating initial and boundary meteorological conditions.
  - Creates `met_em.d01.2016-02-25_00:00:00.nc` for every 6 hour time step, for both domains.
  - Also metgrid.log.
- Copy the anthro_emiss, wesely, exo_coldens, megan_bio_emiss, mozbc executables.

```
cp /nobackup/WRFChem/testrun_files/anthro_emis .
cp /nobackup/WRFChem/testrun_files/wesely .
cp /nobackup/WRFChem/testrun_files/exo_coldens .
cp /nobackup/WRFChem/testrun_files/megan_bio_emiss .
cp /nobackup/WRFChem/testrun_files/mozbc .
```

- Copy the input files for these executables.

```
cp /nobackup/WRFChem/testrun_files/emis_edgarhtap2_mozmos.inp .
cp /nobackup/WRFChem/testrun_files/wesely.inp .
cp /nobackup/WRFChem/testrun_files/exo_coldens.inp .
cp /nobackup/WRFChem/testrun_files/megan_bio_emiss.inp .
cp /nobackup/WRFChem/testrun_files/mozbc.inp .
```

- Copy over the run subdirectory from WRF.

```
cp -r /nobackup/${USER}/WRFChem/run/* .
```

- Remove the testrun version of real.exe and wrf.exe and copy the freshly compiled versions.

```
rm real.exe
rm wrf.exe
cp /nobackup/${USER}/WRFChem/main/real.exe .
cp /nobackup/${USER}/WRFChem/main/wrf.exe .
```

- Link the required MOZART chemical boundary condition files (need previous day too for spin up).

```
cd /nobackup/

cd /nobackup/${USER}
mkdir initial_boundary_chem_mz4
cd initial_boundary_chem_mz4
cp /nobackup/${USER}/initial_boundary_chem_mz4/MZ2016oct .
cd /nobackup/${USER}/testrun
ln -sf /nobackup/${USER}/initial_boundary_chem_mz4/MZ2016oct moz0000.nc
```

- Pre-2018:
  - Download MZ4.
  - Download CAM-Chem.
- Post-2018:
  - Download WACCM.
  - Note the directory needs to change in config.bash (`MOZARTdir`).
- Can access individual days using the script.

```
cd /nobackup/WRFChem/download_and_find_gfs_mz4
. get_MZ4_fcst.bash YYYY MM DD
```

- Edit bash script for real.exe.

```
vi real.bash
```

- This has all the requirements for time, nodes, cores, processors.
  - 1 core required, with h_vmem 6GB.
  - May need to change/remove the project code.
  - Before running real.exe, may need to comment out (with a ! in Fortran) in namelist.input aux_input_6 for megan_bio_emiss (3 lines which relates to this).
- Check namelists, run real, and check progress.

```
qsub real.bash
```

- Interpolates between the intermediate files to create the time domain data at the prescribed time intervals.

```
qstat
```

- When complete, creates:
  - `real.bash.o3502300`.
    - Output from the job submission script (MPI output from job id 3502300).
  - `real.bash.e3502300`.

        * Error from the job submission script (MPI output from job id 3502300).
    - `namelist.output`.
        * `wrfinput_d01` (for initial conditions).
        * `wrfinput_d02` (for initial conditions).
        * `wrfbdy_d01` (for boundary conditions).
    - Check `rsl.error*` that the run was successful.
        * If it fails, the wrfinput and wrfbdy won't be created.
    - Check in `rsl.error*` and `rsl.out*` files for each core.
- Edit namelist for biogenic emissions.

```
vi megan_bio_emiss.inp
```

- Run MEGAN.

```
./megan_bio_emiss < megan_bio_emiss.inp
```

- Creates for both domains (`wrfbiochemi_d*`).
- Edit and run mozbc.

```
vi mozbc.inp
```

- Domain 1, `do_ic = .true.`
    - Updates `wrfinput_d01` (NetCDF) with initial conditions.
- Domain 1, `do_bc = .true.`
    - Updates `wrfbdy_d01` (NetCDF) with boundary conditions.
- If ncview wrfbdy_d01, then can see the 2D curtains in space of the boundary conditions (think of box walls), i.e. T is transect or not, X or Y domain, E east or S south.
    - Domain 2, `do_ic = .true.`
- Updates `wrfinput_d02` (netCDF) with initial conditions for the nested domain, as gets its boundary conditions from the outer domain.

```
./mozbc < mozbc.inp
```

- Run wesely.
    - Reads, reformats, and extracts input data for dry deposition.
    - Copy over the `season_wes_usgs.nc` file.
    - Creates `wrf_season_wes_usgs_d01.nc` and `wrf_season_wes_usgs_d02.nc`.

```
cp /nobackup/WRFChem/wes-coldens/season_wes_usgs.nc .
./wesely < wesely.inp
```

- Run EXO COLDENS.
    - Reads, reformats, and extracts input data.
    - Copy over the `exo_coldens.nc` file.
    - Creates `exo_coldens_d01` and `exo_coldens_d02`.

```
cp /nobackup/WRFChem/wes-coldens/exo_coldens.nc .
./exo_coldens < exo_coldens.inp
```

- Edit anthropogenic namelist (check the NO/NO2 ratio from NOX is correct for your domain).

```
vi emis_edgarhtap2_mozmos.inp
```

- Run anthro_emis.

```
./anthro_emis < emis_edgarhtap2_mozmos.inp
```

- Run for both domain 1 and 2 separately.
  - Change the `start_output_time` and `stop_output_time`.
  - Creates `wrfchemi`.
- Before running `wrf.exe`, may need to comment back in (removing the !) in namelist.input aux_input_6 for megan_bio_emiss (3 lines which relates to this).
- Create bash script for wrf.exe.

```
vi wrf.bash
```

- This has all the requirements for time, nodes, cores, processors.
  - 32 cores required.
- Run wrf.exe.

```
qsub wrf.bash
```

- Can follow the progress by tailing the `rsl.error.0000` file.

```
tail rsl.error.0000
```

- Can also check jobs running on HPC through.

```
qstat
```

- Creates:
  - wrfout files per hour.
  - rsl.out.* (for each core).
  - rsl.error.* (for each core).
- Check linked files were for this ${USER}.
- Post-processing.
  - Not doing in the test run.
- To view wrfout files (without the post-processing).

```
conda activate ncview
ncview wrfout*
```

1.6 Frequently Asked Questions

1.6.1 Recommendations

- Submit runs individually and quality control.
- Check all steps in the process have run correctly.
- Check main.bash.o has "substituting initial chemistry from restart" if do not want a cold start.
- Manage space requirements, as run and output folders can get very large.
- Make use of CPU affinity to have dedicated input/output processors, as these are not scalable in WRF-Chem:
  - Within namelist.wrf.blueprint:
    - &namelist_quilt
      - nio_tasks_per_group = 5 ! number of processors used for IO quilting per IO group.
      - nio_tasks_per_group = 2 ! number of quilting groups
      - So the number of cores = nproc_x * nproc_y + nio_groups * nio_tasks_per_group
      - For example, 42 = 4 * 8 + 2 * 5
- LC and BS have created a selection of data science scripts using Python.

1.6.2 Troubleshooting and errors

- Find the errors' first occurance, checking rsl.error, rsl.out, .e, .o, and .log files within the run folder.
- You need to make sure all programs are compiled and useable, and that the paths in your config.bash point to the correct locations.
- You need to ensure that your data is all available for the period you want to simulate (including meteo spin-up).
- You need to ensure your namelists are correct.
- If the error if related to a FORTRAN run-time error.
- Check WRF FAQ's.
- Check WRF forums.
- Check Google groups for:
  - WRFChem.
  - fire_emiss.
  - anthro_emis.
  - Runs.
- Try increasing the debug level.
- Timesteps for meteorology (time_step in namelist.wrf), chemistry (chemdt in namelist.chem), and biogenics (bioemdt in namelist.chem) need to match (careful of units).
- If no error message given at the bottom of rsl.error. file:
  - Potentially a violation of the CFL criterion:
    - Try reducing the timestep.

        * Try turning w_damping off.
    - Potentially a memory error:
        * Increase the number of cores.
        * Increase the memory per core.

1.6.3 Download ECMWF meteorlogy files

- Create an account with ECMWF.
- Follow the steps.
- Login.
- Retrieve your key.
- Copy the information to ~/.ecmwfapirc
- Create a python2 environment for ecmwf-api-client (this library has not yet been updated for python 3).

```
conda create -n python2_ecwmf -c conda-forge ecmwf-api-client
```

- Go to the folder initial_boundary_meteo_ecmwf.
- Edit the python scripts:
    - Both surface and pressurelevels.
    - Only need to change the date and target name.
- Qsub the .bash scripts
- Edit pre.bash to comment out the GFS and comment in the ECMWF files
- Ensure the date and target name correspond to those you want to run with
- Change the number of meteorological vertical levels from 27 (GFS) to 38 (ECMWF)
- Also, the number of meteorological soil levels from 4 (GFS):
    - To 3 for ECMWF with WRFChem4
    - To 4 for ECMWF with WRFChem3

1.6.4 To run with a nest

- Offline nests:
    - See step-by-step guide to run with a nest document from Carly Reddington.
    - Uses ndown.exe for one-way nesting
    - Feedback = 0
    - Parent and nest domain may drift apart
- Online nests:
    - Turn off urban physics (i.e. sf_urban_physics = 0, 0, 0) in physics subsection of namelist.wrf.
    - Requires a large amount of cores, as memory intensive
    - Uses wrf.exe for two-way nesting

- Feedback = 1
- Have an odd number for the parent_grid_ratio.
- For nest 2, (e_we-s_we+1) must be one greater than an integer multiple of the parent_grid_ratio (3 or 5).
- WRF will decompose each domain in the exact same way, so ensure all the domains are similar shapes (i.e. don't have a square domain within a rectangular domain, or even a rectangular domain which is longer in the x-direction within another domain which is longer in the y-direction).
- Check all namelist settings and check all required nest parameters are set (use registry to check which parameters need to be set for every domain).
- All variables with dimension = max_domains or (max_dom) need to be set for the nests
- Careful the domains are not too big, otherwise wrfinput won't be created
- Use same physics options and physics calling options e.g. radt/cudt
  - An exception is cumulus scheme. One may need to turn it off for a nest that has grid distance of a few kilometers or less.
- For nest, e_we and e_sn for a parent_grid_ratio of 3 must be return a whole number when minus 1 and divide by parent_grid_ratio (3)
- Decrease restart_interval to 720 (2/day) from 360 (4/day)
- Tests with less than 24 hours break the coarsest domains mozbc
- Add diurnal cycle for all domains:
  - Within MAIN_emission_processing.ncl, change to all domains.
- Timestep:
  - Decrease propotionally
- Radiation timestep should coincide with the finest domain resolution (1 minute per km dx), but it usually is not necessary to go below 5 minutes. All domains should use the same value, so that radiation forcing is applied at the same time for all domains.
- Other namelist.wrf settings specific for domains < 3km res
  - &domains
    - smooth_option = 0
  - &physics
    - cugd_avedx = 3
    - smooth_option = 0
    - cu_rad_feedback = .false.
    - cu_diag = 0
    - slope_rad = 1
    - topo_shading = 1
  - &dynamics
    - non_hydrostatic = .false.

1.6.5 To run with cumulus parameterisation off

- Namelist.chem
  - cldchem_onoff = 1
  - chem_conv_tr = 0 (subgrid convective transport)
  - conv_tr_wetscav = 0 (subgrid convective wet scavenging)
  - conv_tr_aqchem = 0 (subgrid convective aqueous chemistry)
- Namelist.wps
  - Resolution < 5 km
- Namelist.wrf
  - Resolution < 5 km
  - cu_physics = 0 (cumulus parameterization off)
  - cugd_avedx = 3 (number of grid boxes over which subsidence is spread)
- Master.bash, turn off nudging
  - s/GRIDFDDA/0/g

1.6.6 Changes for WRFChem4

- Select updates for WRFChem4
  - Bug fixes:
    - NOAH land surface scheme
    - Thompson microphysics scheme
    - Boundary layer and surface schemes from MYNN.
    - Chemical reaction rate constant for reaction: SO_2 + OH -> SO_4
    - Dust < 0.46 microns contribution to AOD.
    - Dust and salt bin contributions to AOD.
    - Urban physics.
    - Fires (module_mosaic_addemiss.F).
    - glysoa not needed as no longer uses to_toa variable which has the summation error (module_mosaic_driver.F).
    - GEOGRID.TBL within WPS4/geogrid is a hard copy of the GEOGRID.TBL.ARW_CHEM including erod.
  - New defaults
    - Hybrid sigma-pressure vertical coordinate.
    - Temperature variable is now moist theta.
    - Method to compute vertical levels, smooth variation of dz.
  - Various improved options available:

    * RRTMK (ra_sw_physics=14, ra_lw_physics=14) improves RRTMG

1.6.7 To run with WRFChem3.7.1 or WRFChem4.2

- Within `config.bash`:
  - Replace all instances of 4.2 with 3.7.1, or vice-versa.
  - Use the appropriate geography files, being either `/nobackup/WRFChem/WPSGeog3` or `/nobackup/WRFChem/WPSGeog4`.
- Within `namelist.wps.blueprint`:
  - For the `geog_data_res` variable (within `&geogrid`), use `'modis_30s+30s'` for WRFChem3.7.1 and use `'default'` for WRFChem4.2.
- Within `namelist.wrf.blueprint`:
  - Remove the `force_use_old_data` variable (within `&time_control`) for WRFChem3.7.1 and have it set to `T` for WRFChem4.2.
  - For the `num_metgrid_soil_layers` variable (within `&domains`), use `4` for WRFChem3.7.1 and `3` for WRFChem4.2.
  - For the `num_soil_layers` variable (within `&physics`), use `4` for WRFChem3.7.1 and `3` for WRFChem4.2.
  - For the `num_land_cat` variable (within `&physics`), use `20` for WRFChem3.7.1 and `21` for WRFChem4.2.

1.6.8 To run with a diurnal cycle

- Choosing the diurnal cycle:
  - There are several different diurnal cycles in WRF_UoM_EMIT.
  - They are contained in the emission_script_data*.ncl files. Whichever of these files is named emission_script_data.ncl will be the diurnal cycle that is read by MAIN_emission_processing.ncl. The current emission_script_data.ncl is a copy of emission_script_data_EU.ncl.
    * EU = European diurnal cycles based on Olivier et al 2003
    * EX = Exaggerated diurnal cycle with 99% of emissions during daytime
    * QH = Qinghua diurnal cycle
  - Change settings in MAIN_emission_processing.ncl
    * time_offset
    * oc_om_scale
- To check if the diurnal cycle application was successful, run the following python script, which should be within WRF_UoM_EMIT, and is automatically linked to your run folder during pre.bash. Take a copy of the file from the following location if you don't have it:

```
python plot_wrfchemi.py
```

1.6.9 To run with NAEI emissions

- Follow the guide created by Ailish Graham.

1.6.10 To add (or remove) variables to wrfout files

- First, check whether the variable is in the Registry. If it isn't, then add it using the steps here.
- Then, if you're running with chemistry edit the file `iofields.chem`, otherwise edit the file `iofields.met`, which are both in WRFotron.
- There are lines of text such as:

```
+:h:0:ccn1,ccn2,ccn3,ccn4,ccn5,ccn6
```

- The + is to add or - is to remove a variable.
- The h is for the history (wrfout) stream. Can have history, restarts, or both.
- The 0 is for the stream number. Generally, stream numbers of 10-24 are okay, and avoid 22-23.
- Then list the variables.

1.6.11 To include upper boundary conditions

- Turn on the `have_bcs_upper` boolean within `namelist.chem.blueprint`.
- Set the lowest pressure level where the upper boundary concentrations are overwritten: `fixed_ubc_press` variable, default is 50 (hPa).
- Provide 2 data files: a climatology for tropopause levels (`clim_p_trop.nc`) and an input file with upper boundary conditions for gas species (`fixed_ubc_inname`).
  - Climatologies for 4 different time periods derived from WACCM RCP simulations are here. A direct download link is here. Within here is the `clim_p_trop.nc` file, along with the 4 different climatology time periods: `ubvals_b40.20th.track1_1950-1959.nc`, `ubvals_b40.20th.track1_1980-1989.nc`, `ubvals_b40.20th.track1_1996-2005.nc`, and `ubvals_rcp4_5.2deg_2020-2029.nc` where the years used to produce the climatology are specified in the file names.
- Copy the climatology files over to each run folder by adding the following to the bottom of `pre.bash`:

```
msg "bringing over upper boundary condition files"
cp /nobackup/${USER}/where_you_place_these_files/{clim_p_trop.nc,ubvals_b40.20th.
↪track1_1996-2005.nc} .
```

- More information is here and here within Chapter 2 here.

1.7 Versions

1.7.1 2.2.0 - 10/11/2020

- Summary of changes:
  - Changing default settings to turn on heterogeneous uptake of N2O5 onto aerosol particles. Within `namelist.chem.blueprint`, `n2o5_hetchem = 1`.

  - Added introductory Python course for analysing WRFChem output.
  - Added blueprints for CEMAC/manual compilation runs.
  - Added Python postprocessing script.
  - Changed default cores to 64.

1.7.2 2.1.0 - 23/06/2020

- Changes relative to version 2.0.0:
  - WRF-Chem4.2
    - Fixes the performance interval issue of WRFChem4.0.3.
  - If still use WRFChem3.7.1, then add aqueous chemistry in stratocumulus clouds in WRFChem3.7.1.
  - Refactored the GitHub repository:
    - Converted user guide to readthedocs.io documentation.
    - Focus on CEMAC WRFotron.
    - Removed old WRFotron code but kept a reference to the settings they used in the user guide.
    - Removed superfluous files.
  - Added no binding for MPI executions `-bind-to none` in `main.bash` and `main_restart.bash`.
  - Added NCO and chemistry variables list version checks to `main.bash` and `main_restart.bash`.
  - Changed the default memory per core to 2G for `main.bash` and `main_restart.bash`.

1.7.3 2.0.0 - 01/02/2019

- Changes relative to version 1.0.0:
  - WRFChem4.0.3.
  - With aqueous chemistry in stratocumulus clouds (`cldchem_onoff = 1`).
    - Works with WRFChem4.0.3.
  - Biomass burning plume rise throughout the boundary layer (`bbinjectscheme = 2`).
    - The original option 2 was 50% at the surface and 50% evenly throughout the BL.
    - The new option 2 has all BB emissions evenly distributed throughout the BL.
    - To add the bbinjectscheme to any new version of WRFChem.
  - Diurnal cycle from Olivier et al., (2003).
  - Aerosol optical properties approximated by Maxwell-Garnett.
  - Updated TUV scheme for photolysis (`phot_opt = 4`).
    - Download the additional data files to your `WRFChem/run` folder.
    - Extract the data directories `DATAE1` and `DATAJ1`, and the `wrf_tuv_xsqy.nc` file from downloaded file using `tar xvf TUV.phot.bz2`.
  - Initial and boundary conditions for chemistry from WACCM for post 2018 or CAM-Chem for pre 2018.

- Fixed the bug where nudging would stop after 312 hours (i.e. after day 13 of a simulation) i.e. changed `gfdda_end_h` to 10,000.
- Nudge above the boundary layer. To do this, go into `namelist.wrf.blueprint`, and within the FDDA section change:

```
if_no_pbl_nudging_uv                = 1, 1, 1,                                  ␣
↪         ! nudging of u and v in the PBL, 0 = yes, 1 = no
if_no_pbl_nudging_t                 = 1, 1, 1,                                  ␣
↪         ! nudging of t in the PBL, 0 = yes, 1 = no
if_no_pbl_nudging_q                 = 1, 1, 1,                                  ␣
↪         ! nudging of q in the PBL, 0 = yes, 1 = no
```

- Hard-coded NCL and NCO commands.
- Fixed the bug where within the anthro_emiss namelist for EDGAR-HTAP2, NH_3 was incorrectly set as an aerosol i.e. removed (a) in the emis_map.
- Fixed the bug in plume rise where extra biomass burning mass was added aloft when the thickness of the vertical grid (dz) increases by altitude.
  - Within `chem/module_chem_plumerise_scalar.F`:
    - `dz_flam=zzcon(k2)-zzcon(k1-1) ! original version.`
    - `dz_flam=zzcon(k2+1)-zzcon(k1) ! fixed version.`
- Corrected the `metInc` within config.bash for ECMWF to be 6 (3 was for GFS).
- Added the faster version of post.bash from Helen Burns in CEMAC.
  - Hard coded NCL and NCO commands in.
  - Also, removed the deletion of pre-processed and temporary wrfout files from the staging directory, as these are often needed for error diagnosis.

1.7.4 1.0.0 - 01/06/2018

- Changes relative to version 0.0.0:
  - MOZART-MOSAIC 4 bin, with aqueous chemistry and VBS SOA (`chem_opt = 202`).
  - Without aqueous chemistry in stratocumulus clouds (*cldchem_onoff = 0*).
    - Does not work with WRF-Chem version 3.7.1.
  - Morrison microphysics (`mp_physics = 10`).
  - Initial and boundary conditions for meteorology from ECMWF.
  - 38 meteoroglogical levels.
  - 3 meteorological soil levels for WRFChem4.0.3 and 4 for WRFChem3.7.1.
  - Consistent timestep for chemistry and biogenics with meteorology.

1.7.5 0.0.0 - 15/10/2015

- WRFChem3.7.1.
- Single domain.

- Continuous nudged meteorology each timestep (with target fields on a 3-hour update freq) with chemical restarts.
- Initial and boundary conditions for meteorology from GFS.
- Initial and boundary conditions for chemistry from MOZART.
- MOZART-MOSAIC 4 bin, without aqueous chemistry and simple SOA (`chem_opt = 201`).
- Horizontal spatial resolution of 30 km spatial resolution.
- 33 vertical levels.
- 27 meteoroglogical levels.
- 180 second timestep for meteorology.
- Thompson microphysics scheme (`mp_physics = 8`).
- Radiation from RRTMG for both long and short-wave.
- Boundary layer scheme from Mellor-Yamada Nakanishi and Niino-2.5.
- Noah Land Surface Model.
- Convective parameterisation from Grell 3-D ensemble.
- Photolysis scheme from Madronich fTUV.
- Emissions.
  - Anthropogenic from EDGAR-HTAPv2.2.
  - Fire from FINN.
  - Biogenic from MEGAN.
  - Dust from GOCART with AFWA.

1.8 License

WRFotron is licensed under the GPLv3 License, free of charge for non-commercial use. If you intend to publish something based on WRFChem simulations made using the WRFotron scripts and think this contributed substantially to you research, please consider offering co-authorship and referencing:

CHAPTER 2

Quick start

WRFotron uses pre-built executables on ARC4 from CEMAC (for University of Leeds users). Everything required is loaded in `config.bash`, including Python, NCO, NCL, WPS, WRFMeteo, WRFChem, preprocessors, and ncview.

1. Log into ARC4, clone the WRFotron repo, and edit the `chainDir` path within `config.bash` if it is not `/nobackup/${USER}/WRFotron`:

```
git clone https://github.com/wrfchem-leeds/WRFotron.git
```

2. Load the availability of CEMAC modules. If have other modules loaded then unload them (`module purge`), and similarly deactivate conda (`conda deactivate`), as both of these can cause conflits.:

```
. /nobackup/cemac/cemac.sh
```

3. From within the WRFotron folder run `master.bash`:

```
. master.bash 2016 10 12 00 24 06
```

For users that require their own executables or that are from outside of the University of Leeds, you can manually compile them using the instructions here.