RUNNING WRF ON LINUX USING PGI COMPILERS

Draft Version

Nash'at Ahmad

1. INSTALLING THE netCDF LIBRARY

netCDF is used in almost all the WRF components and therefore the first step is to make sure that its installed correctly on the system. Download the file from UNIDATA:

http://my.unidata.ucar.edu/content/software/netcdf/index.html

1. Run **./configure**

2. Edit **macros.make** and make the following changes under **#Compilation**

```
CC        = cc
CXX       = g++
FC        = pgf90
F90       = pgf90
CFLAGS    = -O -DNDEBUG -DpgiFortran
CXXFLAGS  = -DNDEBUG -DpgiFortran
```

3. Go in the directory **fortran/** and edit the file **cfortran.h.** Add the line:

#define pgiFortran
add this line before the Fortran compiler related stuff starts.

libnetcdf.a is created in **libsrc/**

4. Decide where you want to put all the netCDF stuff. Create a directory:

/path-to-netCDF/netcdf

5. Create three subdirectories underneath it:

/path-to-netCDF/netcdf/lib

/path-to-netCDF/netcdf/bin

/path-to-netCDF/netcdf/include

Copy **libnetcdf.a** to the **lib/** subdirectory.
Copy **netcdf.h** and **netcdf.inc** to the **include/** subdirectory.
Copy the binaries **ncdump** and **ncgen** to the **bin/** subdirectory.

6. Set the environment variable for netCDF

export NETCDF=/path-to-netCDF/netcdf

This should do it – The next step is installing the WRF model.

2. INSTALLING THE WRF MODEL ON LINUX WITH PORTLAND COMPILERS

1. Download the stuff:

http://www.mmm.ucar.edu/wrf/users/download/get_source2.html

2. Run **./configure** in the WRF source directory.

3. Check **configure.wrf** to make sure that ***netcdf*** shows up with the right path.

4. Run **./compile em_real**

This creates wrf.exe and real.exe in the WRF_SOURCE_CODE/main/ sub-directory.

There are other options with **./compile** also to create executables for idealized test runs.

5. Copy the executables, **wrf.exe** and **real.exe** to a directory which is in the **$PATH** (e.g., ~/bin).

3. INSTALLING THE TERRAIN, LANDUSE AND MISC. OTHER DATASETS

Download the datasets and place them where you want them in the system. The directory structure should look something like this:

```
[lnx352]~/wrf/geo_data> ls -ltr
total 72
drwxr-xr-x   2  wd20na   wd2      4096 Jun 14  2001  pctland_10m/
drwxr-xr-x   2  wd20na   wd2      4096 Jul 29  2002  greenfrac/
drwxr-xr-x   2  wd20na   wd2      4096 Jul 29  2002  albedo_ncep/
drwxr-xr-x   2  wd20na   wd2      4096 Feb 24  2003  islope/
drwxr-xr-x   2  wd20na   wd2     12288 Aug  4  12:44 landuse_30s/
drwxr-xr-x   2  wd20na   wd2      4096 Aug  4  12:57 maxsnowalb/
drwxr-xr-x   2  wd20na   wd2      4096 Aug  4  12:57 soiltemp_1deg/
drwxr-xr-x   2  wd20na   wd2     12288 Aug  4  13:01 soiltype_bot_30s/
drwxr-xr-x   2  wd20na   wd2     12288 Aug  9  16:03 topo_30s/
drwxr-xr-x   2  wd20na   wd2     12288 Aug  9  17:51 soiltype_top_30s/
```

The datasets are large. A **du -H** on **geo_data/** gives a total size of 4.6GB on the disk.

Set an environmental (e.g., in **.bashrc** file) variable to point to these data sources:

export GEOG_DATAROOT=/path-to-datasets/geo_data

4. INSTALLING WRFSI

Define the following environment variables in the ***.bashrc*** or ***.cshrc*** file:

```
export INSTALLROOT=/path-to-wrfsi/wrfsi
export SOURCE_ROOT=/path-to-wrfsi/wrfsi
```

Decide where you would want to create domains and process the met data for different simulations. And define the following environment variables in the ***.bashrc*** or ***.cshrc*** file accordingly:

```
export DATAROOT=/path-to-wrf-case-setups/domains
export EXT_DATAROOT=/path-to-wrf-case-setups/extdata
export TEMPLATES=/path-to-wrf-case-setups/templates
```

For each case, WRFSI will create a directory under **$DATAROOT** and **$TEMPLATES**. Lets say that you have given a simulation the case name **east-us**, then WRFSI will create a directory named **east-us** under **$DATAROOT** and **$TEMPLATES**. We will come to this later.

Next go to the WRFSI source directory and run the installation script:

perl install_wrfsi.pl

This generates the binaries and scripts in the **etc/** and **bin/** subdirectories where the WRFSI source code was installed.

The next step is running the WRFSI. One can run WRFSI from any directory as long as the paths are pointing in the right direction. For convenience create a directory with the case name for your simulation:

mkdir east-us

cd east-us

Copy **wrfsi.nl** and **grib_prep.nl** to your case directory (in this example the directory **east-us**). Edit **wrfsi.nl** and **grib_prep.nl** according to simulation requirements:

1. In **wrfsi.nl,** check the simulation *start, end time* and *domain bounds, resolutions*, etc.
2. In **wrfsi.nl,** check paths to terrain/landuse datasets.
3. Finally at the end of the file (**wrfsi.nl**) make sure that the **si_paths** are pointing to **$EXT_DATAROOT/extprd**.
4. In **grib_prep.pl**, check the path pointing to gridded grib data from global models (e.g., AVN).

The details of all the namelist variables in **wrfsi.nl** can be found in the WRFSI documentation. In the **grib_prep.nl** one needs to make sure that the time information is correct and corresponds to the input in **wrfsi.nl**. The AVN data for initializing the WRF model for Jan 24-25, 2000 is available on the WRF website. The following files can be downloaded:

```
[lnx352]~/wrf/met_data/avn> ls -ltr
total 60884
-rw-r--r--    1 wd20na   wd2      20731540 Aug  6 10:41 fnl_2000-01-24_12
-rw-r--r--    1 wd20na   wd2      20813158 Aug  6 10:41 fnl_2000-01-24_18
-rw-r--r--    1 wd20na   wd2      20720264 Aug  6 10:41 fnl_2000-01-25_00
```

The **wrfsi.nl** and **grib_prep.nl** used with these AVN files are listed in the next two pages.

5. WRFSI NAMELIST FILE (wrfsi.nl)

```
&project_id
 SIMULATION_NAME = 'WRF Model Simulation'
 USER_DESC = 'EAST-US'
/
&filetimespec
 START_YEAR = 2000,
 START_MONTH = 01,
 START_DAY = 24,
 START_HOUR = 12,
 START_MINUTE = 00,
 START_SECOND = 00,
 END_YEAR = 2000,
 END_MONTH = 01,
 END_DAY = 25,
 END_HOUR = 00,
 END_MINUTE = 00,
 END_SECOND = 00,
 INTERVAL = 21600,
/
&hgridspec
 NUM_DOMAINS = 1
 XDIM = 74,
 YDIM = 61,
 PARENT_ID = 1,
 RATIO_TO_PARENT = 1,
 DOMAIN_ORIGIN_LLI = 1,
 DOMAIN_ORIGIN_LLJ = 1,
 DOMAIN_ORIGIN_URI = 1,
 DOMAIN_ORIGIN_URJ = 1,
 MAP_PROJ_NAME = 'lambert',
 MOAD_KNOWN_LAT = 34.726,
 MOAD_KNOWN_LON = -81.226,
 MOAD_STAND_LATS = 30.0, 60.0,
 MOAD_STAND_LONS = -98.
 MOAD_DELTA_X = 30000.
 MOAD_DELTA_Y = 30000.
 SILAVWT_PARM_WRF = 1.
 TOPTWVL_PARM_WRF = 2.
/
&sfcfiles
 TOPO_30S = '/usr1/wd20na/wrf/geo_data/topo_30s',
 LANDUSE_30S = '/usr1/wd20na/wrf/geo_data/landuse_30s',
 SOILTYPE_TOP_30S = '/usr1/wd20na/wrf/geo_data/soiltype_top_30s',
 SOILTYPE_BOT_30S = '/usr1/wd20na/wrf/geo_data/soiltype_bot_30s',
 GREENFRAC = '/usr1/wd20na/wrf/geo_data/greenfrac',
 SOILTEMP_1DEG = '/usr1/wd20na/wrf/geo_data/soiltemp_1deg',
 ALBEDO_NCEP = '/usr1/wd20na/wrf/geo_data/albedo_ncep',
 MAXSNOWALB = '/usr1/wd20na/wrf/geo_data/maxsnowalb',
 ISLOPE = '/usr1/wd20na/wrf/geo_data/islope',
/
```

```
&interp_control
 NUM_ACTIVE_SUBNESTS = 0,
 ACTIVE_SUBNESTS = 2,3,4,
 PTOP_PA = 5000,
 HINTERP_METHOD = 1,
 LSM_HINTERP_METHOD = 1,
 NUM_INIT_TIMES = 1,
 INIT_ROOT = 'AVN',
 LBC_ROOT = 'AVN',
 LSM_ROOT = '',
 CONSTANTS_FULL_NAME = '',
 VERBOSE_LOG = .false.,
 OUTPUT_COORD = 'ETAP',
 LEVELS = 1.000 , 0.990 , 0.978 , 0.964 , 0.946 , 0.922 ,
          0.894 , 0.860 , 0.817 , 0.766 , 0.707 , 0.644 ,
          0.576 , 0.507 , 0.444 , 0.380 , 0.324 , 0.273 ,
          0.228 , 0.188 , 0.152 , 0.121 , 0.093 , 0.069 ,
          0.048 , 0.029 , 0.014 , 0.000
/
&si_paths
 ANALPATH = /usr1/wd20na/wrf/cases/extdata/extprd
 LBCPATH = /usr1/wd20na/wrf/cases/extdata/extprd
 LSMPATH = /usr1/wd20na/wrf/cases/extdata/extprd
 CONSTANTS_PATH = /usr1/wd20na/wrf/cases/extdata/extprd
/
```

6. WRFSI INPUT FILE (grib_prep.pl)

```
&filetimespec
 START_YEAR = 2000
 START_MONTH = 01
 START_DAY = 24
 START_HOUR = 12
 START_MINUTE = 00
 START_SECOND = 00
 END_YEAR = 2000
 END_MONTH = 01
 END_DAY = 25
 END_HOUR = 00
 END_MINUTE = 00
 END_SECOND = 00
 INTERVAL = 21600
/
&gpinput_defs
 SRCNAME = 'AVN'
 SRCVTAB = 'AVN'
 SRCPATH = '/usr1/wd20na/wrf/met_data/avn'
 SRCCYCLE = 6, 6, 6, 6, 12, 12, 24
 SRCDELAY = 3, 4, 4, 3, 0, 0, 36
/
```

7. WRFSI STEPS

EINS – SETTING UP CASE NAME/DIRECTORY

Decide what you are going to call your case, let's say **east-us**.

```
export CASE_DIR=east-us
```

Create the environment variable **$MOAD_DATAROOT**

```
export MOAD_DATAROOT=$DATAROOT$CASE_DIR
```

```
mkdir $TEMPLATES$CASE_DIR
```

At this point it's not a bad idea to be reminded that **$DATAROOT**, **$TEMPLATES** and **$EXT_DATAROOT** are the environment variables created at the beginning of WRFSI installation and they refer to the three directories where WRFSI related data will be processed.

ZWEI – DEFINING DOMAIN BOUNDS A.K.A. DOMAIN LOCALIZATION

Input: **wrfsi.nl** in **$TEMPALTES$CASE_DIR**

Copy your **wrfsi.nl** to **$TEMPLATES$CASE_DIR**

Run the following `perl` script:

```
perl $INSTALLROOT/etc/window_domain_rt.pl -t $TEMPLATES$CASE_DIR
                                          -w wrfsi -c
```

Take a look at WRFSI documentation for a description of command line arguments.

Output: Creates topo, landuse, etc. files in **$DATAROOT$CASE_DIR/static**

DREI – PROCESS THE GRIB DATA

Input: **grib_prep.nl** in **$EXT_DATAROOT/static**

Copy your **grib_prep.nl** file to **$EXT_DATAROOT/static**

Run the following `perl` script:

```
perl $INSTALLROOT/etc/grib_prep.pl -l 12 -s 2000012412 AVN
```

WRFSI documentation gives a detailed description of command line arguments.

Output: Creates files for the script **wrfprep.pl** in **$EXT_DATAROOT/extprd**

For this particular case the following files were generated:

```
[lnx352]~/wrf/cases/extdata/extprd> ls -ltr
-rw-rw-rw-    1 wd20na   wd2      37824120 Aug 11 18:41 AVN:2000-01-24_12
-rw-rw-rw-    1 wd20na   wd2      37824120 Aug 11 18:41 AVN:2000-01-24_18
-rw-rw-rw-    1 wd20na   wd2      37824120 Aug 11 18:41 AVN:2000-01-25_00
```

VIER – GENERATE INITIAL AND BOUNDARY CONDITIONS

Input: **`wrfsi.nl`** file in **`$DATAROOT$CASE_DIR/static`**

Copy your **`wrfsi.nl`** file to **`$DATAROOT$CASE_DIR/static`**

Run the following `perl` script:

```
perl $INSTALLROOT/etc/wrfprep.pl -d ${MOAD_DATAROOT} -s 2000012412
                                 -f 12 -t 6
```

Output: Creates files in **`$DATAROOT$CASE_DIR/siprd`**

In this case the following files were generated:

```
[lnx352]~/wrf/cases/domains/east-us/siprd> ls -ltr
hinterp.d01.2000-01-24_12:00:00
hinterp.d01.2000-01-24_18:00:00
hinterp.global.metadata
hinterp.d01.2000-01-25_00:00:00
wrf_real_input_em.d01.2000-01-24_12:00:00
wrf_real_input_em.d01.2000-01-25_00:00:00
wrf_real_input_em.d01.2000-01-24_18:00:00
CYCLE.0002412000012
```

The files **`wrf_real_input*`** are inputs for the WRF program **`real.exe`**.

8. RUNNING THE WRF MODEL

Inputs:
Files generated by WRFSI (**`wrf_real_input*`**)
Namelist file name: **`namelist.input`**
And the following files (you can copy these from `WRF_SOURCE_DIRECTORY/run`):

`LANDUSE.TBL`	**`GENPARM.TBL`**
`ETAMPNEW_DATA`	**`VEGPARM.TBL`**
`tr49t67`	**`SOILPARM.TBL`**
`RRTM_DATA`	**`tr67t85`**
`tr49t85`	

STEP 1: Run real.exe

It takes the WRFSI files (wrf_real_input*) and consolidates them into single initial and boundary conditions files for each of the domain defined.

Output:

`wrfinput_d01`	(initial conditions for domain 1)
`wrfbdy_d01`	(boundary conditions for domain 1)

STEP 2: Run wrf.exe

`wrf.exe` requires the files mentioned above and uses **`wrfinput_d01`** and **`wrfbdy_d01`** for the run. The output is of the form:

```
wrfout_d01_2000-01-24_12:00:00
wrfout_d01_2000-01-24_18:00:00
wrfout_d01_2000-01-25_00:00:00
```

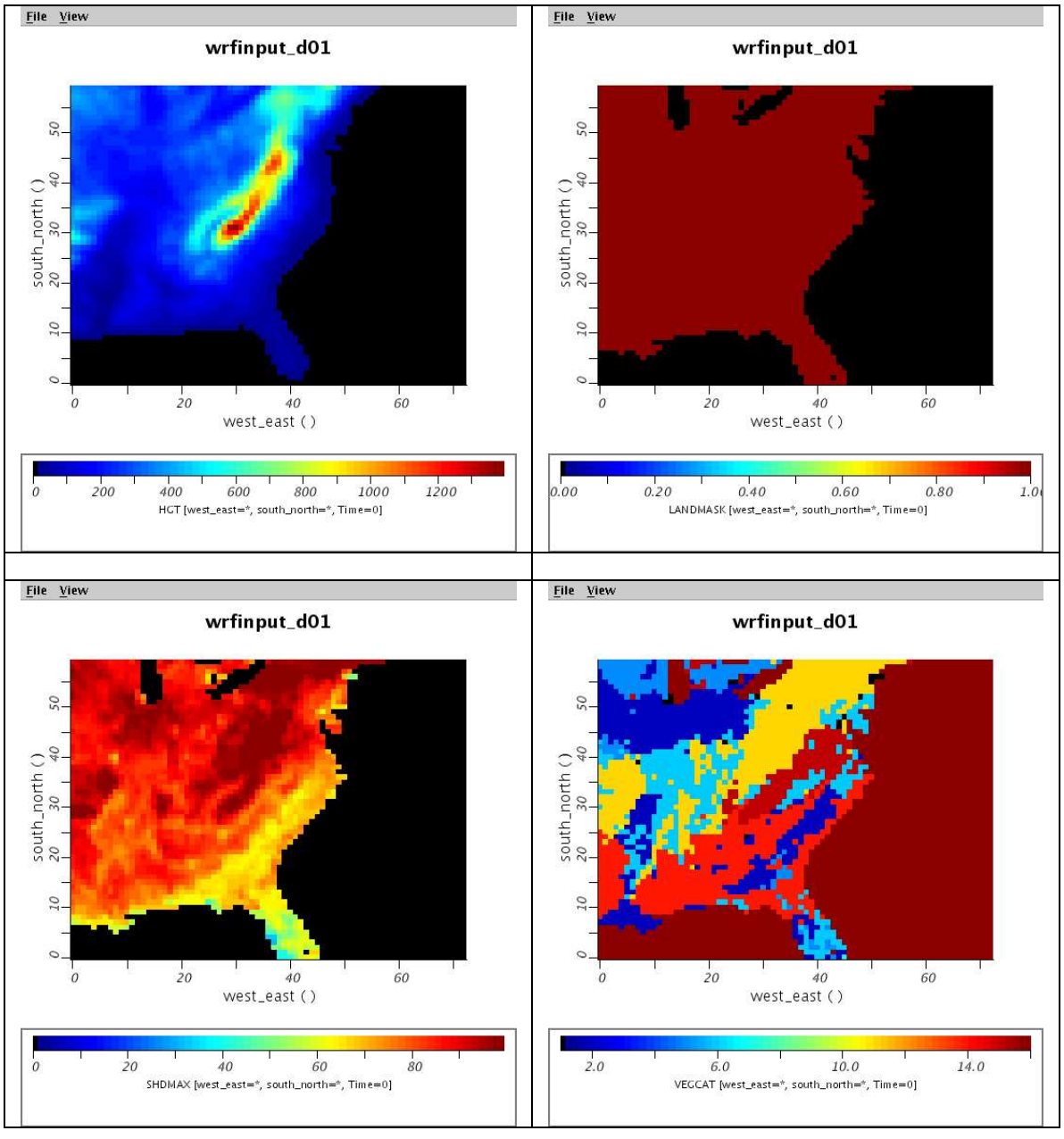
9. SAMPLE SCRIPT WHICH GOES THROUGH WRFSI STEPS DESCRIBED ABOVE

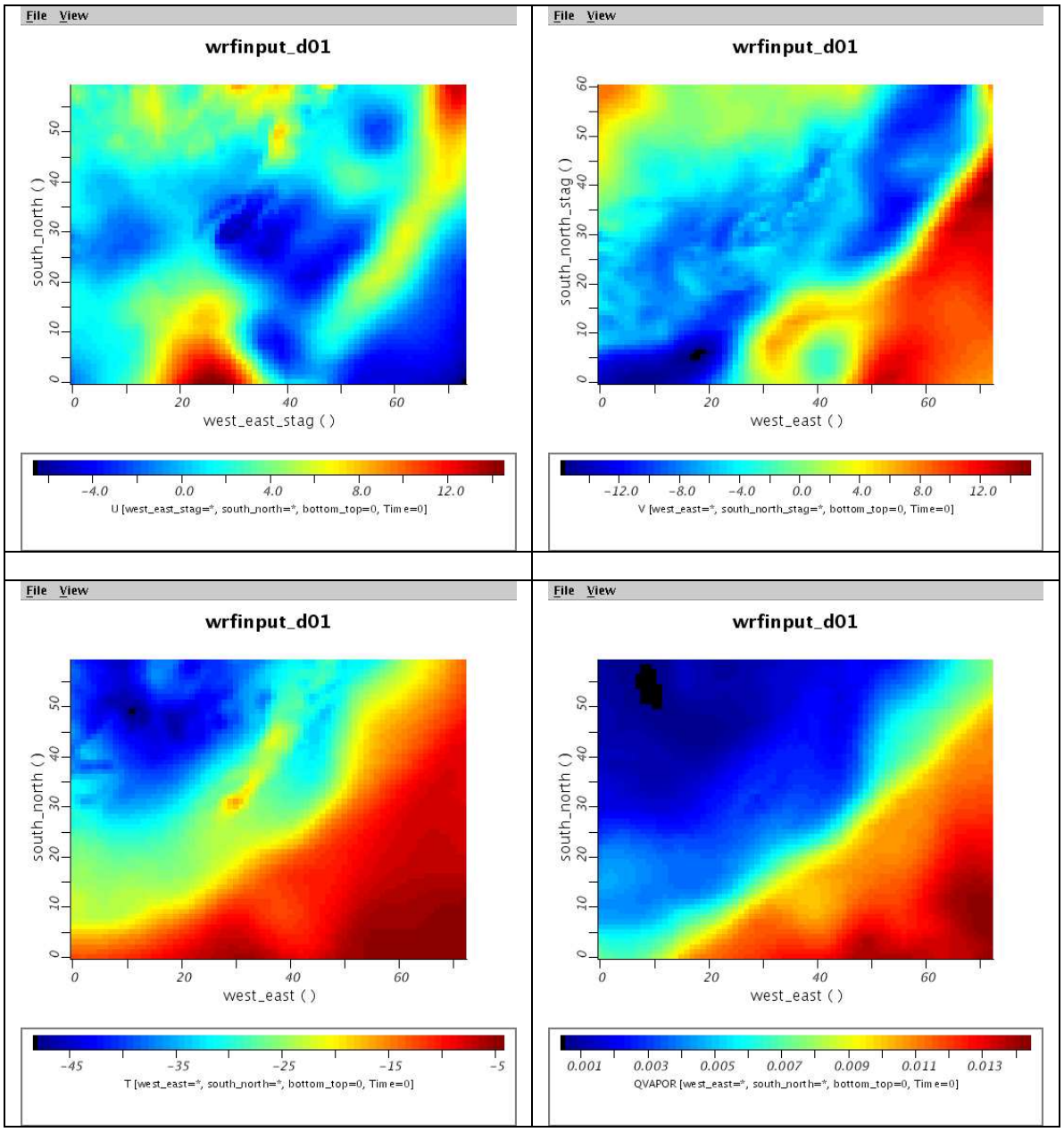
```
#! /bin/csh -f

# ******************************************************************************************
# * WRFSI Tutorial                                                                         *
# * --------------                                                                         *
# *                                                                                        *
# * It is assumed that the files wrfsi.nl and grib_prep.nl are sitting in the directory*
# * from where this script is executed.                                                    *
# *                                                                                        *
# * Edit the CASE_DIR environment variable for a different case.                           *
# *                                                                                        *
# * Nash'at Ahmad                                                                          *
# ******************************************************************************************

# STEP 1: DEFINE CASE (GENERATE THE wrfsi.nl IN $TEMPLATES$CASE_DIR)
# ------------------------------------------------------------------
# - create case directory (case-dir)
setenv CASE_DIR /east-us
mkdir $TEMPLATES$CASE_DIR

# - define wrfsi.nl for the domain parameters and copy it to $TEMPLATES/case-dir
cp ./wrfsi.nl $TEMPLATES$CASE_DIR

# - set MOAD_DATAROOT variable - This is the case directory - This environment variable
# - is needed for subsequent steps.
setenv MOAD_DATAROOT $DATAROOT$CASE_DIR
echo $MOAD_DATAROOT

# =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

# STEP 2: DOMAIN LOCALIZATION
# ---------------------------
# INPUT:  wrfsi.nl in $TEMPLATES$CASE_DIR     (check the paths!!!)

echo ' '
echo 'DOMAIN LOCALIZATION'
echo ' '

perl $INSTALLROOT/etc/window_domain_rt.pl -t $TEMPLATES$CASE_DIR -w wrfsi -c

# OUTPUT: creates topo/landuse files in $DATAROOT$CASE_DIR/static

echo ' '
echo 'OUTPUT IN: '$DATAROOT$CASE_DIR/static
echo ' '

# =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

```
# STEP 3: PROCESS THE GRIDDED GRIB DATA
# ----------------------------------------
# INPUT:  grib_prep.nl in $EXT_DATAROOT/static  (check the paths!!!)

echo ' '
echo 'RUNNING GRIB_PREP'
echo ' '

# - define the parameters in grib_prep.nl file
cp ./grib_prep.nl $EXT_DATAROOT/static

perl $INSTALLROOT/etc/grib_prep.pl -l 12 -s 2000012412 AVN

# OUTPUT: creates files for wrfprep.pl in $EXT_DATAROOT/extprd

echo ' '
echo 'OUTPUT IN: '$EXT_DATAROOT/extprd
echo ' '

# =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

# STEP 4: GENERATE IC's AND BC's
# -----------------------------
# INPUT:  wrfsi.nl                            (check the paths!!!)

echo ' '
echo 'RUNNING WRFPREP'
echo ' '

cp ./wrfsi.nl $DATAROOT$CASE_DIR/static

perl $INSTALLROOT/etc/wrfprep.pl -d ${MOAD_DATAROOT} -s 2000012412 -f 12 -t 6

# OUTPUT: creates files for real.exe in $DATAROOT$CASE_DIR/siprd (wrf_real_input_*)

echo ' '
echo 'OUTPUT IN: '$DATAROOT$CASE_DIR/siprd
echo ' '

# =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

exit(0)

# ***************************************************************************************
```

10. LOOKING AT WRF RESULTS

The final step would be to look at some of the output for a sanity-check. As mentioned before, most of the WRF setup is geared for I/O in the `netCDF`. There are quite a few, freely available software on the net for looking at `netCDF` data files. **`ncBrowse`** is a Java application, which is easy to install and use. Open the file **`wrfinput_d01`** to view the data. The following images are screen captures of **`ncBrowse`** displays:





p.s. Not really sure about those black spots in the images. It is probably a problem with the display from **`ncBrowse`**.

p.p.s If there are any omissions or mistakes please let me know. Thanks.

THAT'S ALL FOLKS
