Portland State University
PDXScholar

University Honors Theses University Honors College

2014

# WRF Validation in Portland, OR

Duc Tran
*Portland State University*

Follow this and additional works at: https://pdxscholar.library.pdx.edu/honorstheses
Let us know how access to this document benefits you.

Recommended Citation
Tran, Duc, "WRF Validation in Portland, OR" (2014). *University Honors Theses.* Paper 100.
https://doi.org/10.15760/honors.72

This Thesis is brought to you for free and open access. It has been accepted for inclusion in University Honors Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

# WRF Validation in Portland, OR

by

Duc Tran

An undergraduate honors thesis submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Science

in

University Honors

and

General Science

Thesis Adviser

Christopher Butenhoff

Portland State University

2014

**Introduction**

Global climate change is starting to destabilize earth's ecosystems. The earth's climate has dramatically changed many times throughout history, but recent events such as burning of fossil fuels, deforestation, and urbanization, have released carbon dioxide (and other greenhouse gases such as nitrous oxide, methane) into the atmosphere at a much faster rate than natural processes. Such rapid growth in greenhouses gases result in global climate change, which can cause a number of negative impacts on the earth, including higher temperatures, rising sea levels, habitat loss, species extinction, as well as social and economic problems.

Greenhouse gases such as carbon dioxide ($CO_2$), methane ($CH_4$), and nitrous oxide ($N_2O$) trap heat in the atmosphere via the greenhouse effect. The greenhouse effect occurs when light energy in the visible part of the electromagnetic spectrum from the sun is absorbed by earth's surface (i.e. rocks, soil, trees and water) and is reemitted as infrared radiation. This radiation is absorbed by greenhouse gases, warming the atmosphere which radiates part of the energy back down to earth's surface. $CO_2$ is the most abundant and important greenhouse gas, causing an estimated radiative forcing of 1.68 W $m^{-2}$ (IPCC 2013). Human activities, both urban and rural play a significant role in increasing $CO_2$ levels (Pataki, 2007). Carbon emissions from human activities have increased overall atmospheric $CO_2$ levels over 25% since preindustrial times (IPCC 2013). The main sources of carbon in urban areas include combustion from transportation (Bond, 2004), natural gas heating in buildings and homes (Glaeser, 2008), vegetation(Guenther, 2002), and industrial and land-use processes. Studies have shown that $CO_2$ concentrations are higher over urban areas than rural areas (Pataki, 2007), and that urbanization increases $CO_2$ emissions (Koerner, 2002).

Many cities have tried to reduce carbon emissions, and in recent years the City of Portland and Multnomah County enacted a Climate Action Plan to mitigate carbon emissions by as much as 40% from 1990 levels by 2030 and 80% by the year 2050.

The Climate Action Plan intends to reduce electricity consumption and energy used for travel, create local jobs and energy-efficient buildings, and reduce and recover 90% of wastes. Previous attempts to mitigate emissions have included the City of Portland's 1993 Carbon Dioxide Reduction Strategy and the Multnomah County-City of Portland Local Action Plan on Global Warming.

In order to verify that the strategies employed by the Climate Action Plan and other emission-reduction programs are effective in reducing carbon dioxide emissions in metropolitan Portland, we need to measure the resulting carbon dioxide above the city and use that data to estimate how much carbon dioxide is being emitted within the city and from what sources. This is known as "top-down" modeling and essentially the idea is that we can use information from the sink of carbon dioxide in the atmosphere to deduce and estimate how much carbon dioxide is emitted from specific sources such as vegetation, transportation, and buildings. Furthermore,

we can determine whether or not certain strategies, such as creating energy-efficient buildings, are effective based on this deductive approach.

To validate bottom-up inventories of emissions with data requires relating the spatial and temporal patterns of surface emissions to the spatial and temporal patterns of atmospheric measurements of $CO_2$. To do so we will build and run the Weather Research and Forecasting model (WRF) for the Portland metropolitan area.

WRF is a mesoscale meteorology model that produces high resolution simulations of wind fields and boundary layer dynamics required for this project. We will validate the output of the WRF model by using real measurements of wind speed and wind direction from DEQ sensors around Portland(obtained from Horizons website: http://www.horizons.pdx.edu/) to compare with the data generated from WRF which uses meteorological and static terrestrial data to simulate and produce corresponding wind direction and wind speed values. Hurricane Katrina tutorial case will also be run using WRF to validate a working model.

**WRF Background**

Using the Weather Research and Forecast Model (WRF), traditionally a tool used by meteorologists, it is possible to use atmospheric measurements of CO2 to infer carbon emissions at high spatial and temporal resolution. WRF uses as inputs, terrestrial geographical data such as surface heights, albedos, and land use, and boundary and initial condition data from reanalysis meteorology data. WRF can model daily meteorology such as temperature and atmospheric pressure. WRF provides the ability to produce high spatial and temporal resolution of $CO_2$ modeling, and offers of a multitude of physics options, including boundary layer physics. Physics processes in WRF include wind dynamics such as turbulence, vertical velocity and eddy currents, as well as long wave and short wave radiation. WRF also provides meteorology settings such as temperature, humidity, atmospheric pressure, and precipitation. Meteorological input helps to simulate photosynthesis and respiration in plants, which produce $CO_2$. For a complete list of physics and meteorology options refer to the WRF-ARW Online Tutorial (http://www2.mmm.ucar.edu/wrf/OnLineTutorial/) or the WRF-ARW user's guide (http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/ARWUsersGuideV3.pdf).

This model can be applied to studying urban carbon sources and sinks in the form of treating carbon as a tracer. A tracer is generally chemically unreactive in and its movement can be tracked. By studying the movement of carbon in WRF, it can be determined whether or not proposed mitigation strategies actually work in reducing atmospheric carbon, or if the carbon is simply moving from one location in the atmosphere to another. Past efforts to use WRF to model $CO_2$ emissions have been moderately successful. One study accurately models $CO_2$ flux and concentration (Ahmadov, 2007) and another study uses the Vegetation Photosynthesis and

Respiration model (VPRM), which can be coupled with WRF to measure the net ecosystem exchange (NEE) for several North American biomes (Pathmathevan, 2008). Total $CO_2$ concentration was modeled using WRF in Los Angeles, California as a top-down modeling technique, where WRF model data was compared with real data from CalNex to validate the efficiency of the model (Park, 2010). The first step in implementing this model is to use existing meteorological data to simulate daily conditions that have already occurred. In comparing data from the simulation versus actual data, validation that WRF works accurately will be accomplished. This method is known as top-down modeling, which in this case estimates $CO_2$ emissions by modeling atmospheric carbon dioxide, which is then used to compare with actual emissions for that given area.

**Methods**

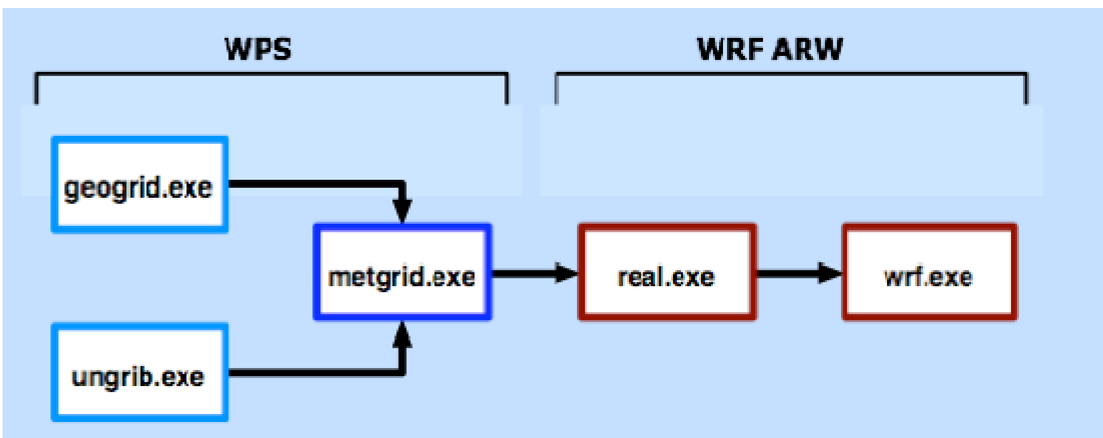


Figure 1. A flowchart for basic WRF processes. WPS includes geogrid.exe and ungrib.exe, which feeds into metgrid.exe where the data is interpolated horizontally and is fed into real.exe.

*Source code compilation*

Initially the source code for WPS and WRF was downloaded. WPS is the preprocessing step that feeds real terrestrial and meteorological information into WRF for the modeling of real cases. WRF and WPS source code were first unzipped and a directory WPS and WRFV3 were created. In the WRFV3 directory, WRF was configured using the Intel compiler with distributed memory parallelism(dmpar, #19). Dmpar was used to allow wrf.exe and real.exe to run on multiple nodes in a cluster, which speeds up processing (these processes will be explained shortly). Another option in configuration is the choice to model with nests. Basic nesting was used, which was option 1. Nesting yields high resolution models, and in this study 2-way nesting was performed, which means daughter nests feed information back into parent nests. Nesting

essentially places a smaller grid domain inside of a larger, coarser domain. The coarser domain provides boundary conditions for the smaller domain. Once all the configuration options were set, WRF was compiled for real data cases, and a directory em_real was created.

In the WPS directory, WPS was configured with the Intel compiler dmpar (#8) as well. Once WPS was compiled, ungrib.exe, geogrid.exe and metgrid.exe files were produced.

To run a real case in WRF, it is necessary to feed real.exe meteorological and terrestrial data in the form of ungrib.exe and geogrid.exe, respectively. Geogrid.exe and ungrib.exe feeds into metgrid.exe, which interpolates the data horizontally across the specified domain in namelist.wps. Namelist.wps is a text document in the WPS directory where one can specify the start date and end date of the interval of their model run, as well as the location and size of all of their nests and domains (this step should be done first before anything is executed in WPS). The data from metgrid.exe feeds into real.exe. To do the WPS preprocessing step -- once terrestrial and meteorological data was downloaded – it was linked in the WPS directory along with Vtables which chose the desired variables. The command "./geogrid.exe" and "./ungrib.exe" was executed before "./metgrid.exe." Metgrid.exe produced the input files to be used for real.exe, which was executed with the command "./real.exe." It was at this step and when executing wrf.exe where it was specified to run on 14 nodes in the CsAR computer cluster.

The option of a nested model was implemented, each with its individual input file created from real.exe to feed into wrf.exe. 2-way nesting was performed, which means daughter nests feed information back into parent nests. Wrf.exe was run on 14 nodes using dmpar on Portland State University's Center for Climate and Aerosol Research's CsAR computer cluster.

A 12-hour case for August 28, 2005 for Hurricane Katrina was performed initially to test and validate the WRF model that was configured. Hurricane Katrina case was run with 4 nested domains with 2-way nesting. The coarsest domain had a 98 x 70 grid. Each successive domain, with each domain placed inside the previous domain, had 118 x 103 grid.

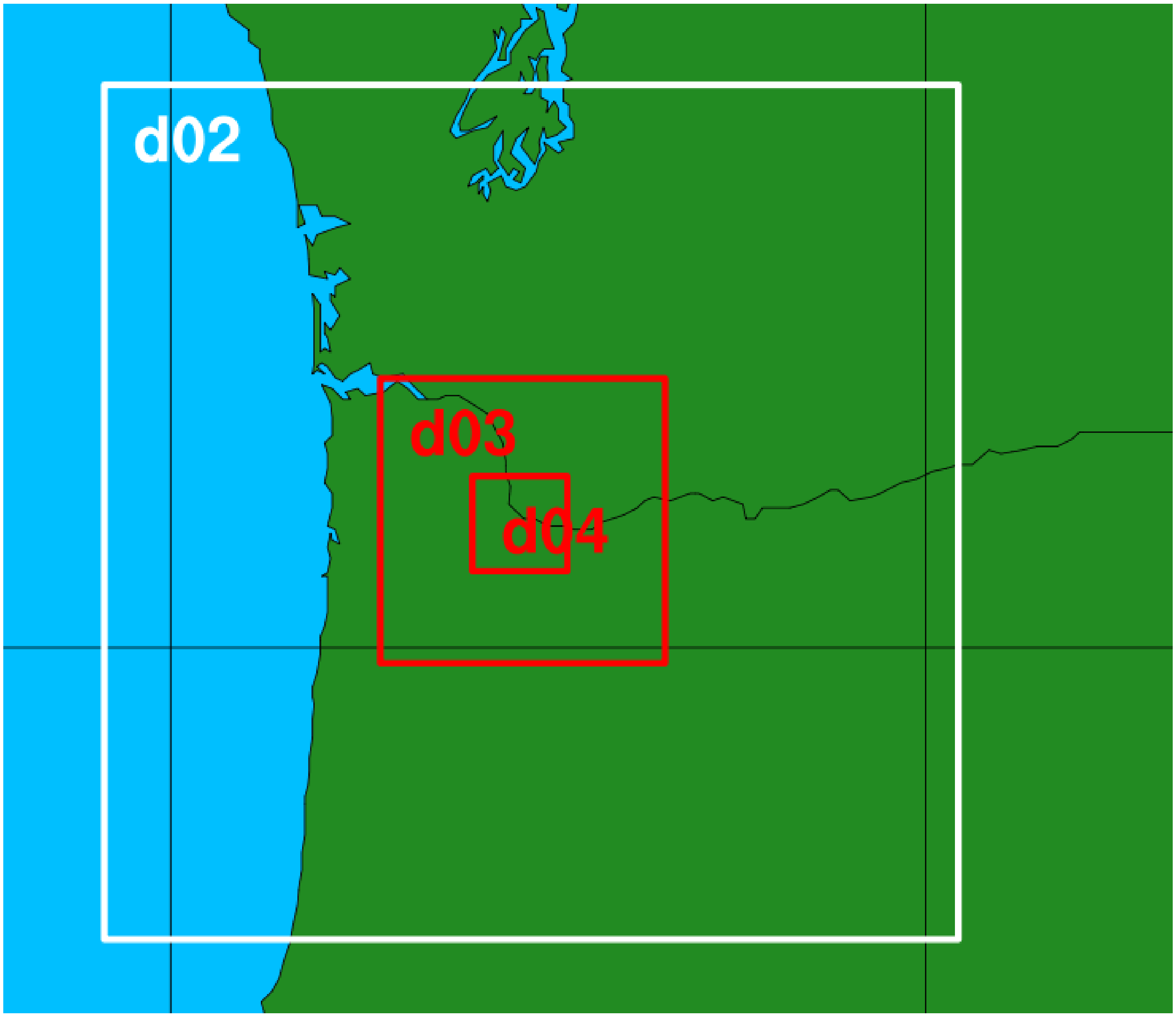


Figure 2: The figure above shows the four nested domains used for the July 1, 2013 case study in Portland, OR.

A case study for July 1, 2013 was also produced using WRF with 4 nested domains for Portland, OR. All 4 nests had a 100 x 100 grid, where the daughter nest was always 1/3 the size of the parent nest. Once output from wrf.exe was produced for a July 1, 2013 case study, post processing with the Ncar Command Language(NCL) was used. NCL scripts for wind speed, wind direction, and air pressure were created using sample NCL scripts as a template. Paths to the WRF output we produced was specified along with the addition of real data from DEQ stations in Portland in order to simultaneously plot WRF and real data at the same time, with an overlay for each real case with WRF. All NCL scripts used are found in Appendix D through J.

For wind speed validation, the DEQ station data from Sauvie Island (DEQSVI) was used to compare to WRF output. For air pressure validation, Portland State University Science Building

2 (PS2) (now called Science Research and Teaching Center) and Forest Grove (FRG) stations were used. For wind direction validation, DEQ stations from DEQSVI and Sprangler Road (DEQSPR) was used.

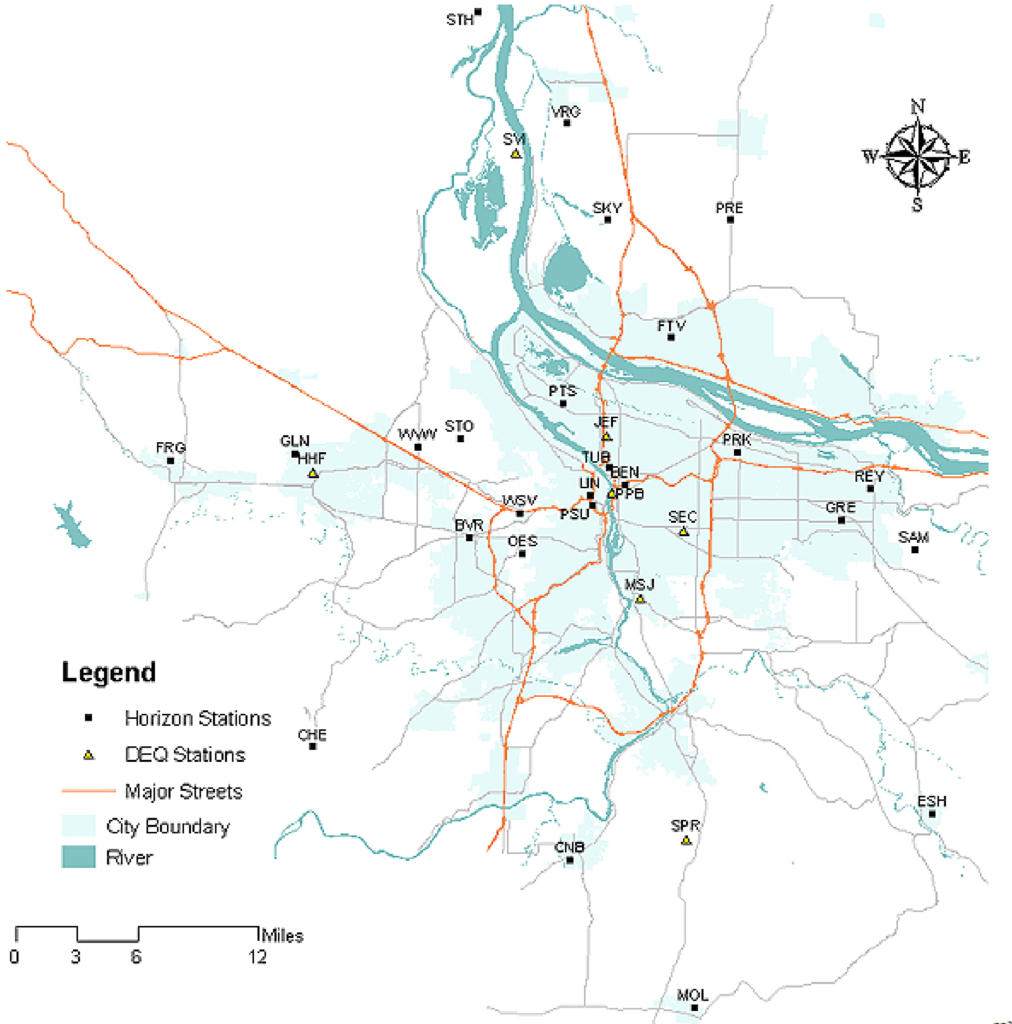


Figure 3: The map above indicates the DEQ station locations used in real data comparisons as triangles.

**Results**

Figure 4 shows a plot overlay of wind speed at Sauvie Island (45.7164° N, 122.8011° W) for WRF(blue) vs measured data(red)

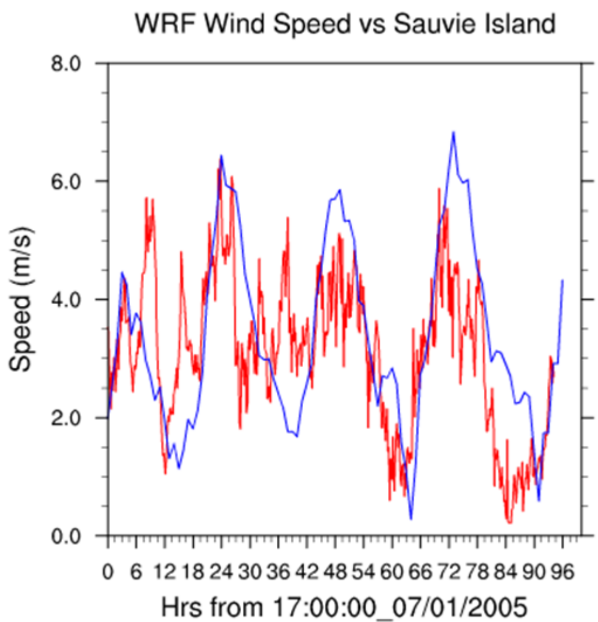


*Figure 41* Plot overlay of wind speed in meters per second vs time for WRF and Horizons data at SVI. SVI data is given by the red line, and WRF data by the blue line.

Figure 5 and 6 shows plot overlays of air pressure(PSFC) at Forest Grove (45.5197° N, 123.1106° W) and Portland State University (45.5128° N, 122.6853° W) for WRF vs measured data.

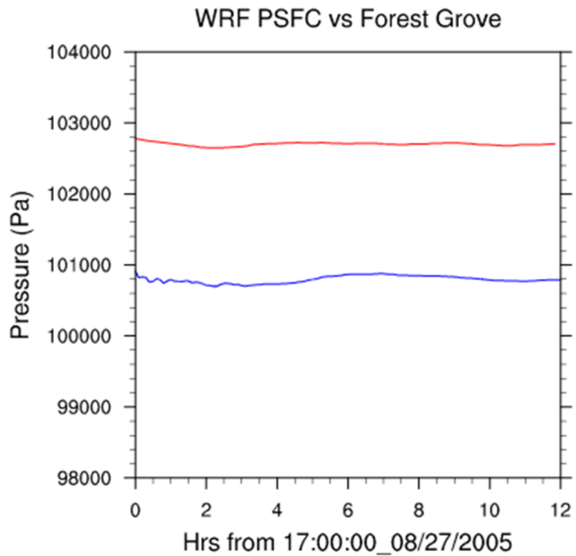


*Figure 5* Plot overlay of air pressure vs time for WRF and Horizons data at Forest Grove. Pressure is units of pascals (Pa). Measured data is given by the red line, and WRF data by the blue line.

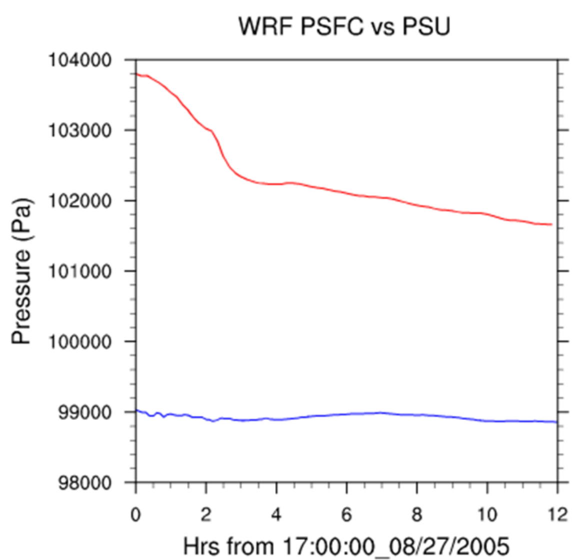


*Figure 6* Plot overlay of air pressure vs time for WRF and Horizons data at PSU. Pressure is units of pascals (Pa). Measured data is given by the red line, and WRF data by the blue line.

Each pair of plots in figures 7.1 & .2 and 8.1 & 8.2 compares WRF wind direction and magnitude withmeasured data (obtained from Horizons website) from Sauvie Island and Sprangler Road (45.2593° N, 122.588° W ), respectively. The length of each line indicates the percentage of the total wind coming from that direction, and the colors indicate the magnitude of the speed of that wind. Wind direction can be treated like a compass where north points vertically upwards from the center of the circle.

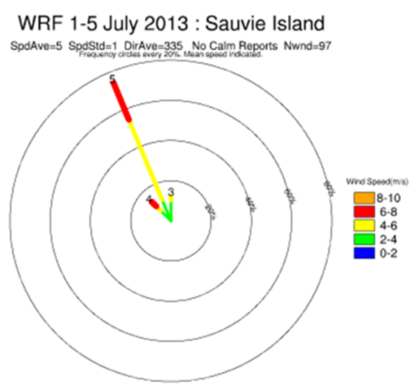


*Figure 7.1* Wind rose plot for WRF data at SVI. Plots show the average wind speed at each direction.

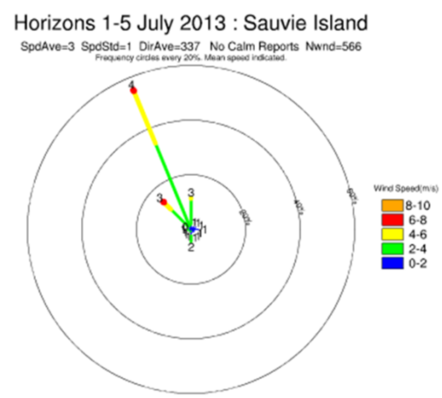


*Figure 7.2* Wind rose plot for Horizons data at SVI. Plots show the average wind speed at each direction.

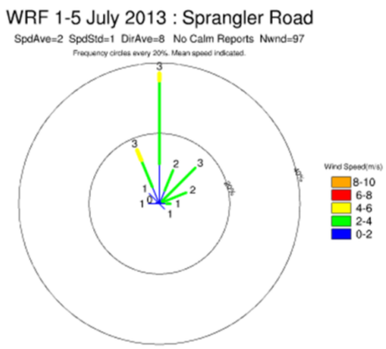


*Figure 8.1* Wind rose plot for WRF data at SPR.

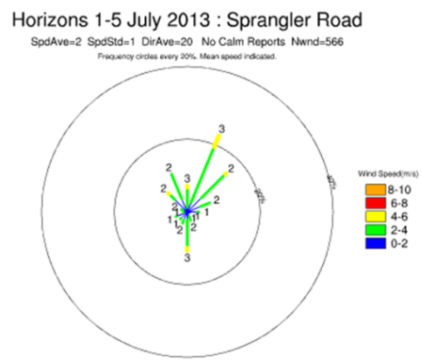


*Figure 8.2* Wind rose plot for Horizons data at SPR.

**Analysis/Conclusion**

WRF was configured, compiled and run successfully. Output files from the Hurricane Katrina default case were produced as well as the case for July 1 – July 5, 2013. Hurricane Katrina case output was viewed using the program Ncview, and July 2013 case plots were produced using NCL scripts.

For wind speed comparison of WRF vs. Sauvie Island data, WRF was able simulated wind speed and direction that followed closely with the measured data. This is important since accurate simulated wind fields are required to properly relate surface CO2 fluxes to atmospheric concentrations. For air pressure plots, the Forest Grove location showed that WRF produced a plot that was similar in its trend but had a difference of a constant 20,000 pascals. Similarly, PSU location had WRF producing data that followed a similar trend but showed a difference of 50,000 pascal. We do not know currently the reason for this discrepancy. Future work is required to diagnose the cause of these differences.

Wind rose comparisons between WRF and measured data at Sauvie Island showed that WRF modeled wind direction very accurately. However, the magnitude of the wind speed showed that WRF modeled wind speed slightly higher in all directions and was not able to reproduce the light wind variability. For wind rose plots at PSU, WRF was able to show that it can model the variability of wind direction, but the percentage of how much wind from each direction was not accurate.

The WRF model appears to be able to model wind speed, air pressure, and wind direction in this research. Some plots showed high accuracy compared to real data in this method of top-down modeling, while others do not. Further validation needs to be done on the WRF model, but this research shows that it has potential to be used as a model for atmospheric research and possibly simulate $CO_2$ emissions spatially and temporally.

**Acknowledgements**

James Powell for technical help in maintaining programs that assisted in building WRF and script writing.

**References**

Ahmadov, R., C. Gerbig, R. Kretschmer, S. Koerner, B. Neininger, A. J. Dolman, and C. Sarrat (2007). Mesoscale covariance of transport and CO2 fluxes: Evidence from observations and simulations using the WRF-VPRM coupled atmosphere-biosphere model, J. Geophys. Res., 112, doi:10.1029/2007JD008552

Alex Guenther (2002), The contribution of reactive carbon emissions from vegetation to the carbon balance of terrestrial ecosystems , Chemosphere Volume 49, 837–844

Bond, C. Tami, David G. Streets, Kristen F. Yarber, Sibyl M. Nelson, Jung-Hun Woo4 and, Zbigniew Klimont (2004), A technology-based global inventory of black and organic carbon emissions from combustion, Journal of Geophysical Research: Atmospheres 109, doi: 10.1029/2003JD003697

Glaeser, L. Edward, Matthew E. Kahn (2010), The greenness of cities: Carbon dioxide emissions and urban development, J. of Urban Econ., 67, 404–418

IPCC, 2013: Summary for Policymakers. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Koerner, B., J. Klopatek (2002), Anthropogenic and natural CO2 emission sources in an arid urban environment, Envir. Pol., 116, S45–S51

Pataki, D. E., T. Xu, Y. Q. Luo, J. R. Ehleringer (2007), Inferring biogenic and anthropogenic carbon dioxide sources across an urban to rural gradient, Oecologia, 152, 307–322

Pathmathevan M., Steven C. Wofsy, Daniel M. Matross, Xiangming Xiao, Allison L. Dunn, John C. Lin, Christoph Gerbig, J. William Munger1, Victoria Y. Chow, Elaine W. Gottlieb (2008), A satellite-based biosphere parameterization for net ecosystem CO2 exchange: Vegetation Photosynthesis and Respiration Model (VPRM), Global Biogeochemical Cycles, 22, doi: 10.1029/2006GB002735

**Appendix A -- WRF & WPS Configuration and Compilation Script**

```
#!/bin/bash -e
# file: duc-thesis.sh
#
# This shell script executes the software DT uses for his thesis.
# The output of a good run of this script is composed of every single
# figure, table, statistical result that DT requires for the thesis.
#
echo "requirements:"
echo "- installed netcdf 4.1.3"
echo "- copies of"
echo "  - WRF-GHGV3.2.tar"
echo "      md5sum 6db252c771034fc8ba4ed3e9c36699c6  WRF-GHGV3.2.tar"
echo "  - wp-scripts-2013-04-30-package-for-DT.zip"
echo "      md5sum dd83b93b6426eeb6f4cca234b7e4d171  wp-scripts-2013-04-30-package-for-DT.zip"
echo "  - WPSV3.5.1.TAR"

echo "      md5sum 332a596fd0179fe5b379ca48777cc9c8  WPSV3.5.1.TAR"

source /act/Modules/3.2.6/init/bash

# Install the environment for the step called "Build and run WRF:"

source /opt/intel/Compiler/11.1/046/bin/iccvars.sh intel64

source /opt/intel/Compiler/11.1/046/bin/ifortvars.sh intel64

module load netcdf/3.6.3/intel11 openmpi/1.2.8/intel11

# BEGIN load sge

# load sge per CB instructions, see /opt/wrf/scripts/cb-scripts-2013-10-28/run_real_script.

module load sge

echo "Setting up SGE info"

#$ -N wrf_real

#$ -S /bin/csh

#$ -pe openmpi 8

#$ -cwd

#$ -o /home/christop/wrf/WRFCHEM/test/em_real/output
```

```
#$ -j y

# END load sge

export WRFIO_NCD_LARGE_FILE_SUPPORT=1

export DM_FC=mpif90

export DM_CC=mpicc

export LD_LIBRARY_PATH=/opt/openmpi/1.2.8/intel11/lib/openmpi/:$LD_LIBRARY_PATH

# Lay in our directory structure.

mkdir -p ~/basemaps

mkdir -p ~/build

mkdir -p ~/tmp

# Do the step called "Build and run WRF:"

# cd ~/WRFGHG/WRFV3

cd ~/build/WRFV3

# enter 17 for WRFV3 platform called

#  17.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (serial)

#echo "17\n0\n" | sh configure

#./compile em_hill2d_x >& compile.log

# end of the step called "Build and run WRF:"

echo "Finished successfully with duc-thesis.sh."

echo "You are now prepared to do the tutorial at
http://www.mmm.ucar.edu/wrf/OnLineTutorial/Compile/arw_compile2.old.htm"

ulimit -s unlimited

# Pick a mode (one mode) from this list:

#

# MODE="em_quarter_ss"

mode="em_real"

wrf_use_debug="yes"

# Build WRF

if [ -f ~/build/WRFV3/main/real.exe ]; then

  echo "WRF (real) has already been built, skipping to the next step."
```

```
elif [ -f ~/build/WRFV3/main/ideal.exe ]; then
  echo "WRF (ideal) has already been built, skipping to the next step."
else
  cd ~/build/WRFV3
  ./clean
  # 19 is intel compiler, dmpar
  # echo "19" | sh configure
  # 15.  Linux x86_64, PGI accelerator compiler with gcc  (dmpar)
  echo "19" | sh configure
  if [ wrf_use_debug="yes" ]; then
    ./compile -d $mode >& compile.log
  else
    ./compile $mode >& compile.log
  fi
  # TODO insert a test that reliably can tell if the compile worked.
  echo "We assume that the WRF compile worked."
  bash
fi
# Build WPS
# Do the step called "Build WPS"
if [ -d ~/build/WPS ]; then
    echo "~/build/WPS is present."
else
    echo "~/build/WPS was not found.  Expanding the WPV3.5.1 tarball into ~/build."
    cd ~/build
    tar xf ~/WPSV3.5.1.TAR
fi
if [ -f ~/build/WPS/compile.log ]; then
  echo "WPS has already been built, skipping to the next step."
```

```
else
  cd ~/build/WPS
  ./clean
  # 19.  Linux x86_64, Intel compiler    (dmpar)
  echo "19" | sh configure
  ./compile >& compile.log
  echo "We assume that the WPS compile worked.  compile.log does not report on success or failure so this is the best we can do."
fi
# end of the step called "Build WPS"
# begin - download geog data
if [ -d ~/basemaps/geog ]; then
   echo "geography has already been expanded."
else
  if [ -f ~/tmp/geog_v3.4.tar.gz ]; then
     echo "geography has already been downloaded."
  else
     cd ~/tmp
     echo "Downloading geography."
     wget http://www.mmm.ucar.edu/wrf/src/wps_files/geog_v3.4.tar.gz
  fi
   echo "expanding geography."
   tar xpzf ~/tmp/geog_v3.4.tar.gz -C /home/duc2/basemaps
fi
# end - download geog data
# begin edit namelist.wps
echo "Editing namelist.wps."
cd ~/build/WPS
# if there's an "original" version (which you would have made by hand
# earlier), restore that to improve the chances that the following sed
```

```
# command (which is a search-and-replace) will work.
# (commented out cos it could easily overwrite our own careful work on namelist.wps)
# cp -f /home/duc2/build/WPS/namelist.wps.bak ~/build/WPS/namelist.wps
sed -i 's,/mmm/users/wrfhelp/WPS_GEOG,/home/duc2/basemaps/geog,g' ~/build/WPS/namelist.wps
echo "namelist.wps has been edited.  You're ready to run geogrid.exe"
# end edit namelist.wps
# begin run geogrid.exe
./geogrid.exe
# end run geogrid.exe


echo "We just built $mode."
# Prepare NCL for execution.
module load ncl
# We're running quarter_ss because, from
#
# http://www.mmm.ucar.edu/wrf/OnLineTutorial/CASES/Ideal/index.html
#
# (accessed 2013-10-24), "- ideal.exe cannot generally be run in
# parallel. For parallel compiles, run this on a single processor.  -
# The exception is the quarter_ss case, which can now be run with MPI."
if [ $mode == "em_quarter"  ]; then
		# Run the em_quarter_ss test case.  Supposed to be the only ideal case
		# that runs in parallel.
		cd ~/build/WRFV3/test/em_quarter_ss
		echo "Doing run_me_first."
		csh ./run_me_first.csh
		echo "Running ideal.exe"
		time mpirun ./ideal.exe
		echo "Running wrf.exe."
```

```
                    time mpirun ./wrf.exe
fi
if [ $mode == "em_real"  ]; then
  echo "geogrid.exe has been run.  Continue please."
  # echo "Please edit /home/duc2/build/WPS/namelist.wps."
  # we need some data as detailed at
  # http://www.mmm.ucar.edu/wrf/OnLineTutorial/CASES/JAN00/ungrib.htm
  if [ -s ~/build/DATA ]; then
          echo "DATA soft link already exists."
  else
          echo "Making DATA soft link."
          cd ~/build
          ln -s /data/raid02/wrf/ ./DATA
  fi
  cd ~/build/WPS

                    # Run em_real
          #         cd ~/build/WRFV3/test/em_real
                    #echo "Doing run_me_first."
                    #csh ./run_me_first.csh
                    #echo "Running ideal.exe"
                    #time mpirun ./ideal.exe
                    #echo "Running wrf.exe."
                    #time mpirun ./wrf.exe
fi

bash
```

**WRF Namelist Scripts**

**Appendix B -- "namelist.wps" Script Used in WPS Preprocessing Step**

```
&share
 wrf_core = 'ARW',
 max_dom = 4,
 start_date = '2005-08-28_00:00:00','2005-08-28_00:00:00','2005-08-28_00:00:00','2005-08-28_00:00:00'
 end_date   = '2005-08-29_00:00:00','2005-08-28_00:00:00','2005-08-28_00:00:00','2005-08-28_00:00:00'
 interval_seconds = 21600
 io_form_geogrid = 2,
/

&geogrid
 parent_id         =   1,   1, 2,3
 parent_grid_ratio =   1,   3, 3,3
 i_parent_start    =   1,  35, 35,35
 j_parent_start    =   1,  23, 23,23
 e_we              =  98, 118, 118,118
 e_sn              =  70,  103, 103,103
 geog_data_res     = '10m','2m', '30s', '30s'
 dx = 30000,
 dy = 30000,
 map_proj = 'mercator',
 ref_lat   =  45.52,
 ref_lon   = -123.682,
 truelat1  =  0.0,
 truelat2  =  0.0,
 stand_lon = -122.682,
 geog_data_path = '/data/raid02/wrf/geog'
/
```

```
&ungrib
 out_format = 'WPS',
 prefix = 'SST',
/

&metgrid
 fg_name = 'FILE'
 io_form_metgrid = 2,
/
```

**Appendix C -- "namelist.input" Script Used in "real.exe" to interpolate vertical layers**

```
&time_control
 run_days                 = 0,
 run_hours                = 12,
 run_minutes              = 0,
 run_seconds              = 0,
 start_year               = 2005, 2005, 2005, 2005
 start_month              = 08,   08,   08, 08
 start_day                = 28,   28,   28, 28
 start_hour               = 00,   00,   00, 00
 start_minute             = 00,   00,   00, 00
 start_second             = 00,   00,   00, 00
 end_year                 = 2005, 2005, 2005, 2005
 end_month                = 08,   08,   08, 08
 end_day                  = 29,   29,   29, 29
 end_hour                 = 00   00,   00, 00
 end_minute               = 00,   00,   00, 00
 end_second               = 00,   00,   00, 00
 interval_seconds         = 21600
```

```
input_from_file                    = .true.,.true.,.true.,.true.
history_interval                   = 180,  60,   20, 6
frames_per_outfile                 = 1000, 1000, 1000, 1000
restart                            = .false.,
restart_interval                   = 720,
io_form_history                    = 2
io_form_restart                    = 2
io_form_input                      = 2
io_form_boundary                   = 2
debug_level                        = 0
/

&domains
time_step                          = 140,
time_step_fract_num                = 0,
time_step_fract_den                = 1,
max_dom                            = 4,
e_we                               = 98,    118,   118, 118
e_sn                               = 70,    103,    103, 103
e_vert                             = 40,    40,    40, 40
p_top_requested                    = 5000,
num_metgrid_levels                 = 27,
num_metgrid_soil_levels            = 4,
dx                                 = 30000, 10000,  3333.33, 1111.1
dy                                 = 30000, 10000,  3333.33, 1111.1
grid_id                            = 1,     2,     3, 4
parent_id                          = 1,     1,     2, 3
i_parent_start                     = 1,     35,    35, 35
j_parent_start                     = 1,     23,    23, 23
```

```
parent_grid_ratio                = 1,     3,     3, 3
parent_time_step_ratio           = 1,     3,     3, 3
feedback                         = 1,
smooth_option                    = 0
/

&physics
mp_physics                       = 3,     3,     3, 3
ra_lw_physics                    = 1,     1,     1, 1
ra_sw_physics                    = 1,     1,     1, 1
radt                             = 30,    30,    30, 30
sf_sfclay_physics                = 1,     1,     1, 1
sf_surface_physics               = 2,     2,     2, 2
bl_pbl_physics                   = 1,     1,     1, 1
bldt                             = 0,     0,     0, 0
cu_physics                       = 1,     1,     0, 0
cudt                             = 5,     5,     5, 5
isfflx                           = 1,
ifsnow                           = 1,
icloud                           = 1,
surface_input_source             = 1,
num_soil_layers                  = 4,
sf_urban_physics                 = 0,     0,     0, 0
/

&fdda
/

&dynamics
```

```
w_damping                  = 0,
diff_opt                   = 1,
km_opt                     = 4,
diff_6th_opt               = 0,      0,      0, 0
diff_6th_factor            = 0.12,   0.12,   0.12, 0.12
base_temp                  = 290.
damp_opt                   = 0,
zdamp                      = 5000.,  5000.,  5000., 5000.
dampcoef                   = 0.2,    0.2,    0.2, 0.2
khdif                      = 0,      0,      0, 0
kvdif                      = 0,      0,      0, 0
non_hydrostatic            = .true., .true., .true., .true.
moist_adv_opt              = 1,      1,      1,  1
scalar_adv_opt             = 1,      1,      1,   1
/

&bdy_control
spec_bdy_width             = 5,
spec_zone                  = 1,
relax_zone                 = 4,
specified                  = .true., .false.,.false., .false.
nested                     = .false., .true., .true., .true.
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
```

```
 nio_groups = 1,
 /
```

**NCL scripts**

**Appendix D -- Script to Construct Windrose Plot for WRF Data at Sauvie Island**

```
;**************************************
; rose_5.ncl
;
; Concepts illustrated:
;   - Drawing wind rose and adding a label bar.
;
;***********************************************
;################################################################
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/wind_rose.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"

begin

; Enter latitude and longitude, south lats/west lons are negative
; clat = 45.7685 clon = -122.7721 ; DEQ Sauvie Island
; clat = 45.4966 clon = -122.6029 ; DEQ SE Lafayette
; clat = 45.2593 clon = -122.588 ; DEQ SE Sprangler Rd
; clat = 45.5128 clon = -122.6853 ; PSU
; clat = 45.5197 clon = -123.1106 ; Forest Grove
; clat = 44.8100 clon = -122.9141 ; Turner Cascade
; clat = 45.5604 clon = -122.6721 ; Jefferson High School
; clat = 45.2900 clon = -122.3344 ; Estacada
```

```
;****************************
; specify WRF output filename
;****************************
 dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
 cdf_filename = "wrfout_d04_20130702000000.4dayUCM.nc"
 f = addfile(dir + cdf_filename,"r")

;****************************
; read in u, v, (m/s) wind data
;****************************
 uvm10 = wrf_user_getvar(f,"uvmet10",-1)
 u10 = uvm10(0,:,:,:)
 v10 = uvm10(1,:,:,:)


;****************************
; specify lat,lon of location
; ***************************
 lon = -122.7721
 lat = 45.7685
 loc = wrf_user_ll_to_ij(f, lon, lat, True)
 mx = loc(0) - 1
 ny = loc(1) - 1

 wspd = ndtooned( sqrt(u10(:,ny,mx)^2 + v10(:,ny,mx)^2) )
 wdir = ndtooned( atan2(u10(:,ny,mx),v10(:,ny,mx))/0.01745329 +180. )

;*****************************
```

```
; specify plot arguments
;******************************
  numPetals =  16                    ; N, NE, E, SE, S, SW, W, NW
  circFr    = 20.
  spdBounds         = (/ 2, 4, 6, 8, 10 /)*1.0

;******************************
; generate color plot
;******************************
  wks = gsn_open_wks("x11","rose")
  res              = True
  res@tiMainString  = "WRF 1-5 July 2013 : Sauvie Island"

  colorBounds       = (/ "blue", "green", "yellow", "red", "orange" /)

  res@gsnFrame      = False         ; do not advance the frame

  wrColor           = WindRoseColor (wks,wspd,wdir,numPetals, \
                              circFr,spdBounds,colorBounds,res)



; Set up resources for the labelbar.

  lbres                    = True          ; labelbar only resources
  lbres@lbAutoManage       = False         ; Necessary to control sizes
  lbres@vpWidthF           = 0.05          ; labelbar width
  lbres@vpHeightF          = 0.15          ; labelbar height
  lbres@vpXF               = 0.80          ; labelbar position in x direction
  lbres@vpYF               = 0.58          ; labelbar position in y direction
```

```
  lbres@lbBoxMajorExtentF  = 0.80         ; puts space between color boxes
  lbres@lbFillColors       = colorBounds   ; labelbar colors
  lbres@lbMonoFillPattern  = True         ; Solid fill pattern
  lbres@lbLabelFontHeightF = 0.015        ; font height. default is small
  lbres@lbPerimOn         = False
  lbres@lbTitleString      ="Wind Speed(m/s)"
  lbres@lbTitleFontHeightF = 0.01

; Create labels

  nbar   = dimsizes(spdBounds)
  labels = new(nbar,string)
  labels(0) = 0 + "-" + spdBounds(0)
  do i=1,nbar-1
    labels(i) = spdBounds(i-1) + "-" +spdBounds(i)
  end do

  lbid = gsn_create_labelbar(wks,nbar,labels,lbres)

  draw(lbid)
  frame(wks)

end
```

**Appendix E -- Script to Construct Wind Rose Plot for Horizons Data at Sauvie Island**

```
;*************************************
```

```
; rose_5.ncl

;

; Concepts illustrated:

;   - Drawing wind rose and adding a label bar.

;

;***********************************************

;############################################################

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/wind_rose.ncl"

load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"


begin


; **********************************************

; Set path and file name to ASCII data

ascii_dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"

ascii_wind_file = "horizons_wind_speed_data.csv"

ascii_dir_file = "Horizons_wind_direction_data.csv"

windf = ascii_dir + ascii_wind_file

dirf = ascii_dir + ascii_dir_file

; ************************************************





; ************************************************

; Read in ascii wind speed data

nrow = numAsciiRow(windf)

ncol = numAsciiCol(windf)
```

```
horizon_wind = readAsciiTable(windf,ncol,"float",1)

; **********************************************


; ************************************************

; Read in ascii wind dir data

nrow = numAsciiRow(dirf)

ncol = numAsciiCol(dirf)

horizon_dir = readAsciiTable(dirf,ncol,"float",1)

; **********************************************


; ************************************************

; Specify which column/Horizons station to use

wspd = horizon_wind(:,1)

wdir = horizon_dir(:,1)

; ************************************************


;******************************

; specify plot arguments

;******************************

  numPetals =  16                  ; N, NE, E, SE, S, SW, W, NW

  circFr    = 20.

  spdBounds        = (/ 2, 4, 6, 8, 10 /)*1.0


;******************************

; generate color plot

;******************************

  wks = gsn_open_wks("x11","rose")

  res             = True
```

```
  res@tiMainString  = "Horizons 1-5 July 2013 : Sauvie Island"

  colorBounds      = (/ "blue", "green", "yellow", "red", "orange" /)

  res@gsnFrame     = False        ; do not advance the frame

  wrColor          = WindRoseColor (wks,wspd,wdir,numPetals, \
                            circFr,spdBounds,colorBounds,res)



; Set up resources for the labelbar.

  lbres                   = True         ; labelbar only resources
  lbres@lbAutoManage      = False        ; Necessary to control sizes
  lbres@vpWidthF          = 0.05         ; labelbar width
  lbres@vpHeightF         = 0.15         ; labelbar height
  lbres@vpXF              = 0.80         ; labelbar position in x direction
  lbres@vpYF              = 0.58         ; labelbar position in y direction
  lbres@lbBoxMajorExtentF  = 0.80         ; puts space between color boxes
  lbres@lbFillColors      = colorBounds   ; labelbar colors
  lbres@lbMonoFillPattern  = True         ; Solid fill pattern
  lbres@lbLabelFontHeightF = 0.015        ; font height. default is small
  lbres@lbPerimOn         = False
  lbres@lbTitleString     ="Wind Speed(m/s)"
  lbres@lbTitleFontHeightF = 0.01

; Create labels

  nbar   = dimsizes(spdBounds)
```

```
  labels = new(nbar,string)

  labels(0) = 0 + "-" + spdBounds(0)

  do i=1,nbar-1

    labels(i) = spdBounds(i-1) + "-" +spdBounds(i)

  end do


  lbid = gsn_create_labelbar(wks,nbar,labels,lbres)


  draw(lbid)

  frame(wks)


end
```

**Appendix F -- Script to Construct Wind Rose Plot for WRF Data at Sprangler Road**

```
;***************************************
; rose_5.ncl
;
; Concepts illustrated:
;   - Drawing wind rose and adding a label bar.
;
;************************************************
;##############################################################
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/wind_rose.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
```

```
begin

; Enter latitude and longitude, south lats/west lons are negative
; clat = 45.7685 clon = -122.7721 ; DEQ Sauvie Island
; clat = 45.4966 clon = -122.6029 ; DEQ SE Lafayette
; clat = 45.2593 clon = -122.588 ; DEQ SE Sprangler Rd
; clat = 45.5128 clon = -122.6853 ; PSU
; clat = 45.5197 clon = -123.1106 ; Forest Grove
; clat = 44.8100 clon = -122.9141 ; Turner Cascade
; clat = 45.5604 clon = -122.6721 ; Jefferson High School
; clat = 45.2900 clon = -122.3344 ; Estacada

;*****************************
; specify WRF output filename
;*****************************
 dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
 cdf_filename = "wrfout_d03_20130702000000.4dayUCM.nc"
 f = addfile(dir + cdf_filename,"r")

;*****************************
; read in u, v, (m/s) wind data
;*****************************
 uvm10 = wrf_user_getvar(f,"uvmet10",-1)
 u10 = uvm10(0,:,:,:)
 v10 = uvm10(1,:,:,:)


;*****************************
```

```
; specify lat,lon of location
; ***************************
  lon = -122.588
  lat = 45.2593
  loc = wrf_user_ll_to_ij(f, lon, lat, True)
  mx = loc(0) - 1
  ny = loc(1) - 1

  wspd = ndtooned( sqrt(u10(:,ny,mx)^2 + v10(:,ny,mx)^2) )
  wdir = ndtooned( atan2(u10(:,ny,mx),v10(:,ny,mx))/0.01745329 +180. )

;******************************
; specify plot arguments
;******************************
  numPetals =  16                   ; N, NE, E, SE, S, SW, W, NW
  circFr    = 20.
  spdBounds        = (/ 2, 4, 6, 8, 10 /)*1.0

;******************************
; generate color plot
;******************************
  wks = gsn_open_wks("x11","rose")
  res             = True
  res@tiMainString  = "WRF 1-5 July 2013 : Sprangler Road"

  colorBounds       = (/ "blue", "green", "yellow", "red", "orange" /)

  res@gsnFrame      = False        ; do not advance the frame
```

```
  wrColor          = WindRoseColor (wks,wspd,wdir,numPetals, \
                               circFr,spdBounds,colorBounds,res)



; Set up resources for the labelbar.

  lbres                  = True         ; labelbar only resources
  lbres@lbAutoManage       = False        ; Necessary to control sizes
  lbres@vpWidthF          = 0.05         ; labelbar width
  lbres@vpHeightF         = 0.15         ; labelbar height
  lbres@vpXF              = 0.80         ; labelbar position in x direction
  lbres@vpYF              = 0.58         ; labelbar position in y direction
  lbres@lbBoxMajorExtentF  = 0.80         ; puts space between color boxes
  lbres@lbFillColors       = colorBounds   ; labelbar colors
  lbres@lbMonoFillPattern  = True         ; Solid fill pattern
  lbres@lbLabelFontHeightF = 0.015        ; font height. default is small
  lbres@lbPerimOn         = False
  lbres@lbTitleString      ="Wind Speed(m/s)"
  lbres@lbTitleFontHeightF = 0.01

; Create labels

  nbar   = dimsizes(spdBounds)
  labels = new(nbar,string)
  labels(0) = 0 + "-" + spdBounds(0)
  do i=1,nbar-1
    labels(i) = spdBounds(i-1) + "-" +spdBounds(i)
  end do
```

```
  lbid = gsn_create_labelbar(wks,nbar,labels,lbres)


  draw(lbid)
  frame(wks)




end
```

**Appendix G -- Script to Construct Wind Rose Plot for Horizons Data at Sprangler Road**

```
;***************************************
; rose_5.ncl
;
; Concepts illustrated:
;   - Drawing wind rose and adding a label bar.
;
;************************************************
;############################################################
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/wind_rose.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"

begin


; **********************************************
; Set path and file name to ASCII data
ascii_dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
ascii_wind_file = "horizons_wind_speed_data.csv"
```

```
ascii_dir_file = "Horizons_wind_direction_data.csv"

windf = ascii_dir + ascii_wind_file

dirf = ascii_dir + ascii_dir_file

; ************************************************




; ************************************************

; Read in ascii wind speed data

nrow = numAsciiRow(windf)

ncol = numAsciiCol(windf)

horizon_wind = readAsciiTable(windf,ncol,"float",1)

; **********************************************



; ************************************************

; Read in ascii wind dir data

nrow = numAsciiRow(dirf)

ncol = numAsciiCol(dirf)

horizon_dir = readAsciiTable(dirf,ncol,"float",1)

; **********************************************


; ************************************************

; Specify which column/Horizons station to use

wspd = horizon_wind(:,2)

wdir = horizon_dir(:,2)

; ************************************************


;******************************
```

```
; specify plot arguments
;******************************
  numPetals =  16                  ; N, NE, E, SE, S, SW, W, NW
  circFr    = 20.
  spdBounds        = (/ 2, 4, 6, 8, 10 /)*1.0

;******************************
; generate color plot
;******************************
  wks = gsn_open_wks("x11","rose")
  res             = True
  res@tiMainString  = "Horizons 1-5 July 2013 : Sprangler Road"

  colorBounds       = (/ "blue", "green", "yellow", "red", "orange" /)

  res@gsnFrame      = False        ; do not advance the frame

  wrColor          = WindRoseColor (wks,wspd,wdir,numPetals, \
                              circFr,spdBounds,colorBounds,res)



; Set up resources for the labelbar.

  lbres                  = True         ; labelbar only resources
  lbres@lbAutoManage       = False        ; Necessary to control sizes
  lbres@vpWidthF          = 0.05         ; labelbar width
  lbres@vpHeightF         = 0.15         ; labelbar height
  lbres@vpXF             = 0.80         ; labelbar position in x direction
  lbres@vpYF             = 0.58         ; labelbar position in y direction
```

```
  lbres@lbBoxMajorExtentF  = 0.80         ; puts space between color boxes

  lbres@lbFillColors       = colorBounds   ; labelbar colors

  lbres@lbMonoFillPattern  = True         ; Solid fill pattern

  lbres@lbLabelFontHeightF = 0.015        ; font height. default is small

  lbres@lbPerimOn          = False

  lbres@lbTitleString      ="Wind Speed(m/s)"

  lbres@lbTitleFontHeightF = 0.01


; Create labels


  nbar   = dimsizes(spdBounds)

  labels = new(nbar,string)

  labels(0) = 0 + "-" + spdBounds(0)

  do i=1,nbar-1

    labels(i) = spdBounds(i-1) + "-" +spdBounds(i)

  end do


  lbid = gsn_create_labelbar(wks,nbar,labels,lbres)


  draw(lbid)

  frame(wks)





end
```

**Appendix H -- Script to Construct Air Pressure Plot Overlay for WRF vs. Forest Grove**

```
; Example series of plotting meteograms with WRF ARW model data
; First let's just get and plot t2 at a point
; Add some into to the plot
```

```
;**********************************************
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRF_contributed.ncl"
;**********************************************
begin
;**********************************************

; *********************************************
; Set path and file name to WRF OUTPUT
dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
cdf_filename = "wrfout_d04_20130702000000.4dayUCM.nc"
a = addfile(dir + cdf_filename,"r")
; *********************************************

; *********************************************
; Set path and file name to ASCII data
ascii_dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
ascii_filename = "pressure_2005.csv"
b = ascii_dir + ascii_filename
; ***********************************************

; ***********************************************
; Read in ascii data
nrow = numAsciiRow(b)
ncol = numAsciiCol(b)
ascii_data = readAsciiTable(b,ncol,"float",1)
; *********************************************

; ***********************************************
; Convert Horizons pressure from mb to pa
ascii_data(:,1) = ascii_data(:,1) * 100.0
ascii_data(:,2) = ascii_data(:,2) * 100.0
ascii_data(:,3) = ascii_data(:,3) * 100.0
; **********************************************


print (ascii_data)

; *********************************************
; Specify the variable name to be extracted
; Use the function wrf_user_getvar to extract data
; "-1" means to extract all data
t2    = wrf_user_getvar(a,"PSFC",-1)    ; get t2 for all times
; *********************************************

; *********************************************
; Extract a times series at a specified lat/lon
; Enter latitude and longitude, south lats/west lons are negative
; clat = 45.7685 clon = -122.7721 ; DEQ Sauvie Island
```

```
; clat = 45.4966 clon = -122.6029 ; DEQ SE Lafeyette
; clat = 45.2593 clon = -122.588 ; DEQ SE Sprangler Rd
; clat = 45.5128 clon = -122.6853 ; PSU
; clat = 45.5197 clon = -123.1106 ; Forest Grove
clat = 45.5197
clon = -123.1106

loc  = wrf_user_ll_to_ij(a,clon,clat,True)
i = loc(0) - 1  ; need to subtract one if used as array index
j = loc(1) - 1  ; need to subtract one if used as array index
t2_point  = t2(:,i,j)             ; extract a time series at a point
; *********************************************

; *********************************************
; Convert temp
t2_point = t2_point -  273.15 ; from K to C
; t2_point = (9./5.)*t2_point + 32.; from C to F
; *********************************************

; *********************************************
; Read timestrings into 2D array (#times, #chars in time_string)
; dims(0) is the number of times
times_in_file = a->Times
dims = dimsizes(times_in_file)
; *********************************************

; *********************************************
; Convert time_string into new format
; New array is 1D
Time_0 = wrf_times_c(times_in_file, 0 )      ; "hours since" initial time on file   (double)
; *********************************************




 wks = gsn_open_wks("x11","meteo2")     ; open a workstation

 t2_res = True
 ;t2_res@tmXBMode       = "Explicit"    ; Define own tick mark labels.
 ;t2_res@tmXBValues     = plot_taus         ; location of explicit labels
 ;t2_res@tmXBLabels     = times        ; labels are the locations

 ; Used to set min/max/spacing of major x-axis tick marks
 t2_res@tmXBMode       = "Manual"
 t2_res@tmXBTickStartF  = 0
 t2_res@tmXBTickEndF    = 12
 t2_res@tmXBTickSpacingF = 2

 t2_res@trYMinF      = 100600
```

```
  t2_res@trYMaxF  = 103000

; Used to set min/max/spacing of major Y-axis tick marks
  ;t2_res@tmYLMode       = "Manual"
  ;t2_res@tmYLTickStartF  = 10
  ;t2_res@tmYLTickEndF    = 28
  ;t2_res@tmYLTickSpacingF = 2


  t2_res@tmXTOn         = False         ; turn off the top tick marks
  t2_res@xyLineThicknesses = 2         ; increase line thickness
  t2_res@xyLineColor    =  "blue"       ; set line color
  ; t2_res@tmXBLabelAngleF  = 90         ; sets CCW rotation of x-axis bottom labels

 ; titles
  ; t2_res@tiMainString       = "WRF t2 vs DEQ Sauvie Island"
  ; t2_res@tiMainString       = "WRF t2 vs DEQ SE Lafeyette"
  t2_res@tiMainString       = "WRF PSFC vs Forest Grove"
  t2_res@tiXAxisString       = "Hrs from 17:00:00_08/27/2005"
  t2_res@tiYAxisString       = "Pressure (Pa)"
  plot0  = gsn_csm_xy(wks,Time_0,t2_point,t2_res)

  t2_res@xyLineColor    =  "red"       ; set line color
  plot1  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,1),t2_res)
plot2  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,2),t2_res)
plot3  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,3),t2_res)

  overlay (plot2, plot0)
  draw (plot2)
  frame(wks)



end
```

**Appendix I -- Script to Construct Air Pressure Plot Overlay for WRF vs. PSU**

```
; Example series of plotting meteograms with WRF ARW model data
; First let's just get and plot t2 at a point
; Add some into to the plot


;***********************************************
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRF_contributed.ncl"
;***********************************************
begin
;***********************************************
```

```
; ***********************************************
; Set path and file name to WRF OUTPUT
dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
cdf_filename = "wrfout_d04_20130702000000.4dayUCM.nc"
a = addfile(dir + cdf_filename,"r")
; ***********************************************

; ***********************************************
; Set path and file name to ASCII data
ascii_dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
ascii_filename = "pressure_2005.csv"
b = ascii_dir + ascii_filename
; ************************************************

; ************************************************
; Read in ascii data
nrow = numAsciiRow(b)
ncol = numAsciiCol(b)
ascii_data = readAsciiTable(b,ncol,"float",1)
; ***********************************************

; ************************************************
; Convert Horizons pressure from mb to pa
ascii_data(:,1) = ascii_data(:,1) * 100.0
ascii_data(:,2) = ascii_data(:,2) * 100.0
ascii_data(:,3) = ascii_data(:,3) * 100.0
; ***********************************************


print (ascii_data)

; ***********************************************
; Specify the variable name to be extracted
; Use the function wrf_user_getvar to extract data
; "-1" means to extract all data
t2    = wrf_user_getvar(a,"PSFC",-1)    ; get t2 for all times
; ***********************************************

; ***********************************************
; Extract a times series at a specified lat/lon
; Enter latitude and longitude, south lats/west lons are negative
; clat = 45.7685 clon = -122.7721 ; DEQ Sauvie Island
; clat = 45.4966 clon = -122.6029 ; DEQ SE Lafeyette
; clat = 45.2593 clon = -122.588 ; DEQ SE Sprangler Rd
; clat = 45.5128 clon = -122.6853 ; PSU
; clat = 45.5197 clon = -123.1106 ; Forest Grove
clat = 45.5197
clon = -123.1106

loc  = wrf_user_ll_to_ij(a,clon,clat,True)
i = loc(0) - 1  ; need to subtract one if used as array index
j = loc(1) - 1  ; need to subtract one if used as array index
t2_point  = t2(:,i,j)            ; extract a time series at a point
```

```
; *********************************************

; *********************************************
; Convert temp
t2_point = t2_point -  273.15 ; from K to C
; t2_point = (9./5.)*t2_point + 32.; from C to F
; *********************************************

; *********************************************
; Read timestrings into 2D array (#times, #chars in time_string)
; dims(0) is the number of times
times_in_file = a->Times
dims = dimsizes(times_in_file)
; *********************************************

; *********************************************
; Convert time_string into new format
; New array is 1D
Time_0 = wrf_times_c(times_in_file, 0 )      ; "hours since" initial time on file   (double)
; *********************************************





 wks = gsn_open_wks("x11","meteo2")     ; open a workstation

 t2_res = True
 ;t2_res@tmXBMode       = "Explicit"    ; Define own tick mark labels.
 ;t2_res@tmXBValues     = plot_taus         ; location of explicit labels
 ;t2_res@tmXBLabels     = times        ; labels are the locations

 ; Used to set min/max/spacing of major x-axis tick marks
 t2_res@tmXBMode       = "Manual"
 t2_res@tmXBTickStartF  = 0
 t2_res@tmXBTickEndF    = 12
 t2_res@tmXBTickSpacingF = 2

 t2_res@trYMinF      = 100600
 t2_res@trYMaxF  = 103000

; Used to set min/max/spacing of major Y-axis tick marks
 ;t2_res@tmYLMode       = "Manual"
 ;t2_res@tmYLTickStartF  = 10
 ;t2_res@tmYLTickEndF    = 28
 ;t2_res@tmYLTickSpacingF = 2


 t2_res@tmXTOn         = False        ; turn off the top tick marks
 t2_res@xyLineThicknesses = 2         ; increase line thickness
```

```
 t2_res@xyLineColor    =  "blue"      ; set line color
 ; t2_res@tmXBLabelAngleF  = 90        ; sets CCW rotation of x-axis bottom labels

; titles
 ; t2_res@tiMainString       = "WRF t2 vs DEQ Sauvie Island"
 ; t2_res@tiMainString       = "WRF t2 vs DEQ SE Lafeyette"
 t2_res@tiMainString       = "WRF PSFC vs Forest Grove"
 t2_res@tiXAxisString      = "Hrs from 17:00:00_08/27/2005"
 t2_res@tiYAxisString      = "Pressure (Pa)"
 plot0  = gsn_csm_xy(wks,Time_0,t2_point,t2_res)

 t2_res@xyLineColor    =  "red"      ; set line color
 plot1  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,1),t2_res)
plot2  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,2),t2_res)
plot3  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,3),t2_res)

 overlay (plot3, plot0)
 draw (plot3)
 frame(wks)



end
```

**Appendix J -- Script to Construct Wind Speed Plot Overlay for WRF vs. Sauvie Island**

```
; Example series of plotting meteograms with WRF ARW model data
; First let's just get and plot t2 at a point
; Add some into to the plot


;************************************************
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRF_contributed.ncl"
;************************************************
begin
;************************************************

; ***********************************************
; Set path and file name to WRF OUTPUT
dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
cdf_filename = "wrfout_d04_20130702000000.4dayUCM.nc"
a = addfile(dir + cdf_filename,"r")
; ***********************************************

; ***********************************************
; Set path and file name to ASCII data
ascii_dir = "/home/duc2/build/WRFV3/test/em_real/Portland/"
ascii_filename = "horizons_wind_speed_data.csv"
```

```
b = ascii_dir + ascii_filename
; ***********************************************

; ***********************************************
; Read in ascii data
nrow = numAsciiRow(b)
ncol = numAsciiCol(b)
ascii_data = readAsciiTable(b,ncol,"float",1)
; *********************************************

; ***********************************************
; Convert Horizons pressure from mb to pa
;ascii_data(:,1) = ascii_data(:,1) * 100.0
;ascii_data(:,2) = ascii_data(:,2) * 100.0
;ascii_data(:,3) = ascii_data(:,3) * 100.0
; **********************************************


; print (ascii_data)

; *********************************************
; Specify the variable name to be extracted
; Use the function wrf_user_getvar to extract data
; "-1" means to extract all data
   u10 = wrf_user_getvar(a,"U10",-1)    ; u at 10 m, mass point
   v10 = wrf_user_getvar(a,"V10",-1)    ; v at 10 m, mass point


speed = sqrt((u10*u10) + (v10*v10))
;atan(u10/v10)



; *********************************************

; *********************************************
; Extract a times series at a specified lat/lon
; Enter latitude and longitude, south lats/west lons are negative
; clat = 45.7685 clon = -122.7721 ; DEQ Sauvie Island
; clat = 45.4966 clon = -122.6029 ; DEQ SE Lafeyette
; clat = 45.2593 clon = -122.588 ; DEQ SE Sprangler Rd
; clat = 45.5128 clon = -122.6853 ; PSU
; clat = 45.5197 clon = -123.1106 ; Forest Grove
clat = 45.7685
clon = -122.7721

loc  = wrf_user_ll_to_ij(a,clon,clat,True)
i = loc(0) - 1  ; need to subtract one if used as array index
j = loc(1) - 1  ; need to subtract one if used as array index
; t2_point  = t2(:,i,j)             ; extract a time series at a point
speed_point  = speed(:,i,j)             ; extract a time series at a point
; *********************************************
```

```
; ********************************************
; Convert temp
; t2_point = t2_point -  273.15 ; from K to C
; t2_point = (9./5.)*t2_point + 32.; from C to F
; ********************************************

; ********************************************
; Read timestrings into 2D array (#times, #chars in time_string)
; dims(0) is the number of times
times_in_file = a->Times
dims = dimsizes(times_in_file)
; ********************************************

; ********************************************
; Convert time_string into new format
; New array is 1D
Time_0 = wrf_times_c(times_in_file, 0 )      ; "hours since" initial time on file   (double)
; ********************************************




 wks = gsn_open_wks("x11","meteo2")     ; open a workstation

 t2_res = True
 ;t2_res@tmXBMode       = "Explicit"    ; Define own tick mark labels.
 ;t2_res@tmXBValues     = plot_taus         ; location of explicit labels
 ;t2_res@tmXBLabels     = times         ; labels are the locations

 ; Used to set min/max/spacing of major x-axis tick marks
 t2_res@tmXBMode       = "Manual"
 ;t2_res@tmXBTickStartF  = 0
 ;t2_res@tmXBTickEndF    = 96
 t2_res@tmXBTickSpacingF = 6

 t2_res@trYMinF      = 0
 t2_res@trYMaxF  = 8

; Used to set min/max/spacing of major Y-axis tick marks
 ;t2_res@tmYLMode       = "Manual"
 ;t2_res@tmYLTickStartF  = 10
 ;t2_res@tmYLTickEndF    = 28
 ;t2_res@tmYLTickSpacingF = 2


 t2_res@tmXTOn         = False         ; turn off the top tick marks
 t2_res@xyLineThicknesses = 2         ; increase line thickness
 t2_res@xyLineColor    =  "blue"       ; set line color
 ; t2_res@tmXBLabelAngleF  = 90         ; sets CCW rotation of x-axis bottom labels
```

```
 ; titles
  ; t2_res@tiMainString       = "WRF t2 vs DEQ Sauvie Island"
  ; t2_res@tiMainString       = "WRF t2 vs DEQ SE Lafeyette"
  t2_res@tiMainString       = "WRF Wind Speed vs Sauvie Island"
  t2_res@tiXAxisString       = "Hrs from 17:00:00_07/01/2013"
  t2_res@tiYAxisString       = "Speed (m/s)"



plot0  = gsn_csm_xy(wks,Time_0,speed_point,t2_res)    ; plot WRF data

  t2_res@xyLineColor    =  "red"       ; set line color



plot1  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,1),t2_res) ; plot Horizons data for col1
 plot2  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,2),t2_res)  ; plot Horizons data for col2
 plot3  = gsn_csm_xy(wks,ascii_data(:,0),ascii_data(:,3),t2_res)  ; plot Horizons data for col3

  overlay (plot1, plot0)   ; overlay plot3 on plot0
  draw (plot1)            ; draw plot
  frame(wks)



end
```