NU-WRF Version 7
User's Guide

Updated for Fourth Patch Release
September 1, 2015

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

Contents

1 Introduction **1**
1.1 This Document . . . 1
1.2 Acknowledgments . . . 1

2 NU-WRF System **3**
2.1 Components . . . 3
2.2 Other Features . . . 6
2.3 History . . . 7

3 Obtaining Software **16**
3.1 Software Usage Agreement . . . 16
3.2 Tar File . . . 17
3.3 Subversion Repository . . . 17
3.4 Directory Structure . . . 17

4 Building Software **19**
4.1 Compilers . . . 19
4.2 External Libraries and Tools . . . 19
4.3 Build System . . . 20

5 Front-End Workflows **24**
5.1 Basic Workflow . . . 24
5.2 Land Surface Initialization and LIS Coupling . . . 25
5.3 Use of GEOS-5 Meteorological Data . . . 28
5.3.1 GEOS2WRF . . . 29
5.3.2 MERRA2WRF . . . 40
5.4 Use of New Erodible Soil Options . . . 44
5.5 Use of GOCART or MERRA Aerosol Data . . . 46
5.6 Use of CASA CO_2 Data . . . 50
5.7 Use of RSS Sea Surface Temperature Data . . . 51

6 Post-Processors **55**
6.1 G-SDSU . . . 55
6.2 RIP4 . . . 55
6.3 ARWPOST . . . 56
6.4 UPP . . . 56
6.5 MET . . . 57
6.6 LVT . . . 57

A Frequently Asked Questions **58**

B Porting NU-WRF **63**

References **68**

1 Introduction

1.1 This Document

This is the NASA Unified-Weather Research and Forecasting (NU-WRF) Version 7 User's Guide. This document provides an overview of NU-WRF; describes how to download, compile, and run the NU-WRF software; and provides guidance on porting the software to new platforms.

This document consists of six sections and two appendices:

- Section 1 is the present introductory section.
- Section 2 provides general information about the NU-WRF project, components, and development history.
- Section 3 provides information on obtaining a software usage agreement and the NU-WRF source code.
- Section 4 describes how to compile the NU-WRF software;
- Section 5 describes several front-end workflows that can be employed with the NU-WRF modeling system, ranging from basic weather simulation to aerosol coupling to coupling with LIS. This includes information on new pre-processors and changes to WRF and LIS.
- Section 6 describes several post-processors for visualization and/or verification.
- Finally, Appendix A answers Frequently Asked Questions about NU-WRF, while Appendix B provides guidance on porting NU-WRF to new platforms.

1.2 Acknowledgments

The development of NU-WRF has been funded by the NASA Modeling, Analysis, and Prediction Program. The Goddard microphysics, radiation, aerosol coupling modules, and G-SDSU are developed and maintained by the Mesoscale Atmospheric Processes Laboratory at NASA Goddard Space Flight Center (GSFC). The GOCART and NASA dust aerosol emission modules are developed and maintained by the GSFC Atmospheric Chemistry and Dynamics Laboratory. The LIS, LDT, and LVT components are developed and maintained by the GSFC Hydrological Sciences Laboratory. The CASA2WRF, GEOS2WRF, GOCART2WRF, LISCONFIG, MERRA2WRF, and SST2WRF components are developed and maintained by the GSFC Advanced Software and Technology Group; SST2WRF includes binary reader source code developed by Remote Sensing Systems.

Past and present contributors affiliated with the NU-WRF effort include: Kristi Arsenault, Clay Blankenship, Scott Braun, Rob Burns, Jon Case, Mian Chin, Tom Clune, David Considine, John David, Jim Geiger, Mei Han, Ken

Harrison, Arthur Hou, Takamichi Iguchi, Jossy Jacob, Randy Kawa, Eric Kemp, Dongchul Kim, Kyu-Myong Kim, Jules Kouatchou, Anil Kumar, Sujay Kumar, William Lau, David Liu, Yuqiong Liu, Toshi Matsui, Hamid Oloso, Christa Peters-Lidard, Chris Redder, Scott Rheingrover, Joe Santanello, Roger Shi, David Starr, Rahman Syed, Qian Tan, Wei-Kuo Tao, Zhining Tao, Eduardo Valente, Bruce Van Aartsen, Gary Wojcik, Di Wu, Benjamin Zaitchik, Brad Zavodsky, Sara Zhang, Shujia Zhou, and Milija Zupanski.

The mainstream community WRF, WPS, and ARWPOST components are developed and supported by the National Center for Atmospheric Research (NCAR), which is operated by the University Corporation for Atmospheric Research. The Kinetic Pre-Processor included with WRF-Chem was primarily developed by Dr. Adrian Sandu of the Virginia Polytechnic Institute and State University. The community RIP4 is maintained by NCAR and was developed primarily by Dr. Mark Stoelinga, formerly of the University of Washington. The community UPP is developed and maintained by the NOAA National Centers for Environmental Prediction. The community MET is developed and supported by the Developmental Testbed Center at NCAR. The mainstream community PREP_CHEM_SOURCES is primarily developed by CPTEC/INPE, Brazil.

NU-WRF Version 7 "Arthur" is dedicated to the late Dr. Arthur Hou, who served as a Co-Principal Investigator, an inspiration, and a mentor.

2 NU-WRF System

NU-WRF has been developed at Goddard Space Flight Center (GSFC) as an observation-driven integrated modeling system representing aerosol, cloud, precipitation, and land processes at satellite-resolved scales (Peters-Lidard et al., 2015). NU-WRF is intended as a superset of the standard NCAR Advanced Research WRF [WRF-ARW; Skamarock et al. (2008)] and incorporates:

- The GSFC Land Information System [LIS; see Kumar et al. (2006) and Peters-Lidard et al. (2007)], coupled to WRF and also available as a standalone executable;
- The WRF-Chem enabled version of the Goddard Chemistry Aerosols Radiation Transport model [GOCART; Chin et al. (2002)];
- GSFC radiation and microphysics schemes including revised couplings to the aerosols (Shi et al., 2014); and
- The Goddard Satellite Data Simulator Unit [G-SDSU; see Matsui et al. (2014)].

In addition, multiple pre- and post-processors from the community and from GSFC have been collected with WRF and LIS. Taken together, the NU-WRF modeling system provides a framework for simulating aerosol and land use effects on such phenomena as floods, tropical cyclones, mesoscale convective systems, droughts, and monsoons (Peters-Lidard et al., 2015). Support also exists for treating CO_2 as a tracer, with plans to further refine into source components (anthropogenic versus biogenic). Finally, the software has been modified to use netCDF4 with HDF5 compression, reducing netCDF file sizes by up to 50%.

Work is underway to incorporate NU-WRF in a Maximum Likelihood Ensemble Filter-based atmospheric data assimilation system, with the capability of assimilating cloud and precipitation affected radiances. In addition, some secondary, rarely used elements of the community WRF modeling system that are not yet included with NU-WRF will be added in the future.

2.1 Components

The NU-WRF package contains the following components:

- The *WRFV3* component contains a modified copy of the core WRF version 3.5.1 modeling system [see Chapter 5 of NCAR (2014)], the WRF-Fire wildfire library [see Appendix A of NCAR (2014)], the WRF-Chem atmospheric chemistry library (ESRL, 2013), and several preprocessors (REAL, CONVERT_EMISS, TC, NDOWN, and NUP). These codes have been modified to add the 2014 Goddard radiation package (Matsui and Jacob, 2014), the new Goddard 4ICE microphysics scheme, the new Goddard 3ICE microphysics scheme (Shi et al., 2014), couplings between these

schemes and GOCART, and a CO_2 tracer parameterization. The community WRF versions of the Goddard microphysics and radiation are now separate options from the latest versions developed by GSFC. The component also includes LIS 7.1r (in *WRFV3/lis*) with code modifications to both LIS and WRF-ARW to facilitate on-line coupling between the atmosphere and land models, as well as land data assimilation (NASA, 2015b).

- The *WPS* component contains a modified copy of the WRF Preprocessing System version 3.5.1 [see Chapter 3 of NCAR (2014)]. This includes the GEOGRID, UNGRIB, and METGRID programs used to set up a WRF domain, to interpolate static and climatological terrestrial data, to extract fields from GRIB and GRIB2 files, and to horizontally interpolate the fields to the WRF grid.

- The *ldt* component contains version 7.1r of the NASA Land Data Toolkit (LDT) software (NASA, 2015a). This acts both as a preprocessor for LIS (including interpolation of terrestrial data to the LIS grid and separate preprocessing for data assimilation) and a postprocessor for LIS (merging dynamic fields from a LIS off-line "spin-up" simulation with static data for eventual input to WRF in LIS coupled mode).

- The *utils/lisconfig* component contains the NASA LISCONFIG software for customizing LDT and LIS ASCII input files so the domain(s) (grid size, resolution, and map projection) match that of predefined WRF grid(s). It uses output from the WPS GEOGRID program to determine the reference latitude and longitude.

- The *utils/geos2wrf_2* component contains version 2 of the NASA GEOS2WRF software, which extracts and/or derives atmospheric data from the Goddard Earth Observing System Model Version 5 (Rienecker et al., 2008) for input into WRF. It also contains MERRA2WRF, which can preprocess atmospheric fields from the Modern-Era Retrospective Analysis for Research and Applications (MERRA) (Rienecker et al., 2011) dataset hosted by the Goddard Earth Sciences Data Information Services Center, as well as preliminary MERRA2 reanalyses made available to select users by the NASA Global Modeling and Assimilation Office. These programs essentially take the place of UNGRIB in WPS, as that program cannot read the netCDF, HDF4, or HDFEOS formats used with GEOS-5, MERRA, and MERRA2.

- The *utils/sst2wrf* component contains the NASA SST2WRF preprocessor, which reads sea surface temperature (SST) analyses produced by Remote Sensing Systems (http://www.remss.com) and converts into a format readable by the WPS program METGRID. This essentially takes the place of UNGRIB as the SST data are in a non-GRIB binary format.

- The *utils/prep_chem_sources* component contains a modified copy version of the community PREP_CHEM_SOURCES version 1.3.2 preprocessor.

This program prepares anthropogenic, biogenic, wildfire, and volcanic emissions data for further preprocessing by the WRF-Chem preprocessor CONVERT_EMISS, before finally being put into netCDF4 format for input to WRF. The NU-WRF version uses the WPS map projection software to ensure consistency in interpolation, and adds support for GFEDv3.1 biomass burning emissions [see van der Werf et al. (2010) and Randerson et al. (2013)], NASA QFED wildfire emissions [see Darmenov and da Silva (2013)], support for new 72-level GOCART background fields, improved interpolation of the GOCART background fields when the WRF grid is at a relatively finer resolution, and output of data for plotting with the NASA PLOT_CHEM program.

- The *utils/plot_chem* component stores a simple NCAR Graphics based PLOT_CHEM program for visualizing the output from PREP_CHEM_SOURCES. This program is only intended for manual review and sanity checking, not for publication quality plots.

- The *utils/gocart2wrf_2* component stores version 2 of GOCART2WRF, a NASA program for reading GEOS-5 GOCART aerosol data and MERRAero (Kishcha et al., 2014) data, interpolating to the WRF grid, and adding the data to the netCDF4 initial condition and lateral boundary condition (IC/LBC) files for WRF.

- The *utils/casa2wrf* component contains the NASA CASA2WRF preprocessor and related software to read CO_2 emissions and concentrations from the CASA biosphere model, and interpolate and append the data to the WRF netCDF4 IC/LBC files (Tao et al., 2014).

- The *RIP4* component contains a modified copy of the NCAR Graphics-based Read/Interpolate/Plot (Stoelinga, 2006) graphical postprocessing software, version 4.6.5. Modifications include support for UMD land use data (from LIS 7) and WRF-Chem output.

- The *ARWpost* component contains version 3.1 of the GrADS-compatible ARWpost program for visualization of output [see Chapter 9 of NCAR (2014)].

- The *UPP* component contains a modified copy of the NCEP Unified Post Processor version 2.1. This software can derive fields from WRF netCDF output and write in GRIB format [see Chapter 7 of DTC (2014)].

- The *MET* component contains version 4.1 of the Meteorological Evaluation Tools (DTC, 2013b) software produced by the Developmental Testbed Center. This can be used to evaluate WRF atmospheric fields converted to GRIB via UPP against observations and gridded analyses.

- The *lvt* component contains version 7.1r of the NASA Land Verification Toolkit (NASA, 2015c) for verifying LIS land and near-surface fields against observations and gridded analyses.

- The *GSDSU* component contains version 3.3.3 of the Goddard Satellite Data Simulation Unit (Matsui and Kemp, 2014), which can be used to simulate satellite imagery, radar, and lidar data for comparison against actual remote-sensing observations.

In addition, NU-WRF includes a unified build system written in the Bash scripting language to ease compilation of the different NU-WRF components, and to automatically resolve several dependencies between the components (e.g., WPS requires WRFV3 to be compiled first). The build system is discussed in section 4.3.

2.2 Other Features

While NU-WRF aims to be a superset of the official WRF modeling system, there are currently some WRF components that are not supported. These are:

- WRF-Hydro, a hydrologic library for WRF (Gochis et al., 2013). The source code for WRF-Hydro is located in *WRFV3/hydro*, and there are plans to add compilation support and coupling between WRF-Hydro and LIS.
- WRF-DA, the data assimilation library for WRF [see Barker et al. (2012) and Chapter 6 of NCAR (2014)]. The source code for WRF-DA is located in *WRFV3/var*, and there are plans to add compilation support in the near future.
- OBSGRID, an objective analysis program for WRF [see Chapter 7 of NCAR (2014)]. The source code for this program is not included with NU-WRF.
- PROC_OML, a preprocessor for interpolating HYCOM (Bleck et al., 2002) 3D ocean model temperature data and writing for further processing by the WPS METGRID program. See Chapter 10 of NCAR (2014). The source code for this program is not included with NU-WRF.
- WRF-NMM, an alternative dynamic core developed by NOAA/NCEP (DTC, 2014). The NMM source code is located in *WRFV3/dyn_nmm* but there is no support to compile it with the NU-WRF build system. While it may be possible to run WRF-NMM with the physics packages and coupling added by NASA, it has never been tested and is not recommended.
- HWRF, the Hurricane WRF modeling system developed by NOAA/NCEP with on-line coupling between WRF-NMM and the Princeton Ocean Model for Tropical Cyclones [Tallapragada et al. (2013); see also

  http://www.dtcenter.org/HurrWRF/users/index.php]. This is a standalone software package provided by the Developmental Testbed Center, and is not included in NU-WRF.

- READ_WRF_NC, a utility similar to the netCDF **ncdump** program for examining WRF netCDF output [see Chapter 10 of NCAR (2014)]. The source code is not included in NU-WRF.
- IOWRF, a utility for manipulating WRF-ARW netCDF data [see Chapter 10 of NCAR (2014)]. The source code is not included in NU-WRF.
- P_INTERP, a utility to interpolate WRF netCDF output to user-specified isobaric levels [see Chapter 10 of NCAR (2014)]. The source code is not included in NU-WRF.
- V_INTERP, a utility to add vertical levels to a WRF-ARW netCDF file [see Chapter 10 of NCAR (2014)]. The source code is not included in NU-WRF.

In addition, the NU-WRF build system (discussed in Section 4.3) does not currently support compilation of any "ideal" data case [described in Chapter 4 of NCAR (2014)], nor compilation with OpenMP or hybrid OpenMP-MPI. The latter decision is due to a lack of OpenMP support in some of the new physics packages (this is being addressed and may be supported in Version 8).

2.3 History

- Version 7-3.5.1-p4 (Official "Arthur" Patch Release 4)
  - Upgraded LIS to 7.1r. This includes:
    - Noah 3.6, NoahMP 3.6, and CABLE 1.4b LSMs (only Noah 3.6 tested with NU-WRF).
    - Flood and drip irrigation.
    - VIIRS Daily GVF data
    - TRMM 3B42 V7 real time precipitation
    - Gaussian T1534 GFS met forcing data
    - MERRA-2 met forcing data (data only available to select users).
    - Downscaling precipitation (PRISM) (NLDAS-2 only)
    - Should compile now with gfortran.
  - Upgraded LDT to 7.1r. Updates include:
    - Fixes to several metforcing readers that process the forcing terrain-height or elevation fields when processing model parameters.
    - Corrected conserved (budget-bilinear) interp routines. Users may notice slight differences in results.
    - MetTimeDScale and Metforcproc runmodes.
    - Crop, CLM2, Flake, Mosaic, Noah, SiB2, and VIC parameters.
    - TRMM 3B42 V7 real time precipitation.
    - Aquarius L2, GCOMW AMSR2 L3, and SMOS L2 soil moisture observations.

    * Simulated GRACE products.
  - Upgraded LVT to 7.1r. Updates include:
    * Fixed bug affecting output of summary and final metric files.
    * GCOMW, ASCAT, SMOS, and SMOS L1 Tb observations.
    * MODIS LST and Great Lakes Hydro data.
    * Time lagged computations.
  - Upgraded compiler version to Intel 13.1.3.192.

- Version 7-3.5.1-p3 (Official "Arthur" Bug Fix Release 3)
  - Updated *pleiades.cfg* to use new generic SGI MPT module on Pleiades. Older versions of SGI MPT are scheduled for removal on 6 July 2015, and users are advised to switch to the generic module to prevent future breakage when SGI MPT is upgraded and older versions are removed.
  - Fixed netCDF4 chunking and deflation (compression) in GOCART2WRF when writing new chemistry fields to wrfinput and wrfbdy. Also removed unused variables from source code, and reorganized Makefiles.
  - Added new temporalInterpolation GEOS2WRF tool, plus *run_temporalInterpolation_merra2_3hr.discover.sh* script to automate and interpolate 6-hourly MERRA2 reanalyses to 3-hourly.
  - Upgraded LIS to 7.0rp3.
  - Added options to suppress dynamic dust emission and to spread dust across multiple near-surface levels.
  - Updated sample batch scripts to check if SGI MPT, Intel MPI, or MVAPICH2 is used and invoke the corresponding appropriate run command when starting a MPI program (**mpirun**, **mpiexec**, **mpiexec_mpt**, or **mpiexec.hydra**).
  - Modified ARWPOST to read wrfout files from WRF-Chem.
  - Modified PREP_CHEM_SOURCES to support NASA QFED fire emissions.
  - Modified *Run_MERRA2.csh* to allow user to override base path of MERRA2 data via environment variable.
  - Added build system option to compile WRF with Allinea MAP profiling libraries (only works on Discover).

- Version 7-3.5.1-p2 (Official "Arthur" Bug Fix Release 2)
  - Fixed WRF-Chem GOCART cloud fraction code to run when not using Grell cumulus scheme.
  - Updates to LIS MERRA-Land reader code – better handling of missing input.

- Updated *Run_MERRA2.csh* script to reflect current stream status and to use consistent indentation.
- Added MERRA2 support in LIS.
- Added *run_geos2wrf_merra2_assim.discover.sh* script to run GEOS2WRF to process 3-hourly MERRA2 assimilation data (GEOS-5 output as the model is adjusting towards the MERRA2 6-hr analyses via IAU). The collection for this data is a bit different than the 6-hr analyses, and it was easier to use GEOS2WRF rather than edit MERRA2WRF. Script only works on Discover, and assumes the user has access to the MERRA2 GMAO directory on Discover.
- Added bug fix to GEOS2WPS to handle 3-hourly MERRA2 assimilation data.
- Bug fix for MERRA-Land in LIS (fixes downward shortwave radiation index).
- Allow WRF to run with Noah LSM selection if REAL processed LIS data. Allows user to switch from WRF-LIS coupling to WRF-Noah-initialized-from-LIS without rerunning LIS to produce GRIB files.
- Updated LIS/LDT/LVT Makefiles to reduce number of include directories to search through (cuts down on build time).
- Fixed loops in WRF-Chem convective transport code to skip nv=1 (which is not a valid chemical species), and to loop through all chemical species instead of just gas (back ported from WRF-Chem 3.6.1).
- Added value for variable DELZ_THRESOLD in WRF-Chem plume model. Plume model with stop when plume top changes less than 100 meters over ten minutes.
- Bug fix to PREP_CHEM_SOURCES, resolving conflicting symbols imported from two different modules.

• Version 7-3.5.1-p1 (Official "Arthur" Bug Fix Release)

- Modified build config files for Discover to check operating system version on computer used for compilation. New default configuration requires SLES 11.3 (same as the new Haswell nodes) and Intel MPI 5. A separate config file *discover_intel13_sgimpt_sp3.cfg* is also added to allow use of SGI MPT. Older Intel MPI 4 and MVAPICH2 configurations are moved to *discover_intel13_impi4_sp1.cfg* and *discover_intel13_mvapich2_sp1.cfg*, respectively, and require the older SLES 11.1 operating system.
- Fixed build config file *pleiades_intel13_impi.cfg* to compile on Pleiades with Intel MPI (default configuration uses SGI MPT).
- Merged in LIS 7.0rp2. Bug fixes: Do not reset Zh and Zm for processes with zero tiles; several VIC LSM related changes, and updates for CMAP.

  - Merged in LDT 7.0rp2. Bug fixes to mapping between LIS fine Lambert Conformal grid and coarser Lat-Lon parameter grid extents. Also fixes domain extent checks for NLDAS-1 and NLDAS-2 forcing.
  - Bug fixes to Goddard 2011 and 2014 radiation schemes. Transmission functions in the CO2, O3, and three water vapor bands with strong absorption are now computed using table look-up. Significantly improves accuracy for pressures less than 10 mb.
  - Modified WRF and LIS configure templates to add -xCORE-AVX2 compiler flag options. However, early tests show slower run times compared to -xSSE4.2, so the build cfg files do not currently use them.
  - Merged in Weile Wang's (NASA ARC) modifications to the WRF spectral nudging code. FFT runs significantly faster with Weile's changes.
  - Modified WRF code to fix -DBENCH instrumentation for most parts of the WRF solver.
  - Added Python scripts to calculate summary metrics from WRF RSL files for benchmarking.
  - Modified MERRA2WRF to add preliminary support for MERRA2 data files released to select users by the GMAO. "Official" support will not occur until after MERRA2 is released to the general public.
  - Build system fixes for handling CASA preprocessors.
  - Updated sample Discover batch scripts for Haswell nodes.

- Version 7-3.5.1 "Arthur" (Official Release)
  - Merged WRF 3.5.1, WPS 3.5.1, PREP_CHEM_SOURCES 1.3.2, RIP 4.6.5, UPP 2.1, and MET 4.1.
  - Merged LIS 7.0rp1, LDT 7rp1, and LVT 7r, and updated LISCONFIG.
  - Merged GSDSU 3.3.3.
  - Modified build system to use netCDF4 with HDF5 compression for *all* programs that rely on netCDF. Also removed **lisreal** build option, turning all LIS related compile-time code changes into run-time changes.
  - Added 2014 Goddard radiation package, new Goddard 4ICE microphysics, and new Goddard 3ICE microphysics. Community WRF versions of Goddard microphysics and radiation are now separate options.
  - Added new soil erodibility options to WRF-Chem: MDB, DYN_CLIMO, and DYN.
  - Skin temperature bug fix for restarts when using time-varying sea ice.

  - Fixed processing of GFEDv3.1 biomass burning emissions.
  - Improved interpolation of GOCART background fields.
  - Added support for MERRAero data in addition to the GEOS-5 GOCART data to use in WRF-Chem using GOCART2WRF utility.
  - Updated Vtable.LIS for optional UNGRIB processing of LIS GRIB2 files.
  - Revised WRF diagnostic mean and standard deviation calculations to use Welford algorithm (less sensitive to roundoff errors, avoids NaNs when fields vary little with time).
  - Added on-line diagnostics: precipitable water, liquid water path, ice water path, cloud liquid water path, cloud ice water path, rain water path, frozen precipitation water path, time-averaged integrated water vapor transport vector, and freezing level.
  - Added spatial subsetting option for GEOS2WPS program in GEOS2WRF.
  - Added support for CASA climatological CO_2 tracers in WRF-Chem, including preprocessors.
  - Removed SZIP library build dependencies.
  - Updated sample batch scripts for Pleiades and Discover.
  - Temporarily dropped support for gfortran compiler.
  - Upgraded to SGI MPT 2.11r13 on Pleiades with Intel compilers.
  - Added MVAPICH2 support on Discover with Intel compilers.

- Version 6-3.4.1-p2 (Official Bug Patch Release)
  - Merged in LIS 6.1rp7 updates, including fixes to latitude of NARR forcing, and bug fixes to WRFout reader.
  - Bug fixes to PREP_CHEM_SOURCES (memory allocation and namelist initialization).
  - Improved error checking for NaNs.
  - Source code updates to allow compilation on OS X with gfortran and OpenMPI.
  - Bug fix to *MET/pcp_combine* ensuring closure of files.
  - Fixed WRF bug for changing restart dump interval when simulation is itself restarted.
  - Build system improvements for LIS, UPP, and GSDSU.
  - Reorganized sample batch scripts into Discover/SLURM and Pleiades/PBS versions.

- Version 6-3.4.1-p1 (Official Bug Patch Release)
  - LIS bug fix (for directory creation) to allow use with SGI MPT.

  - Updated optimization flags, targeting Westmere or newer Intel hardware.
  - Upgraded to Intel MPI 4.0.3.008 on Discover, and to SGI MPT 2.08r7 on Pleiades.
  - Build system tweaks for WRF and LIS to reduce compile times.
  - Build system fix for cleaning GSDSU.

- Version 6-3.4.1 (Official Release)
  - Merged in WRF 3.4.1, WPS 3.4.1, UPP 2.0, and multiple MET 4.0 bug patches.
  - Merged in LIS 6.1rp6.
  - Merged in GSDSU 3.0.
  - Overhauled WRF-LIS coupling. Added **lisreal** build option to compile WRF and REAL with special logic supporting LIS. Added &lis block to *namelist.input* to allow REAL to process LIS netCDF output file and update *wrfinput* accordingly.
  - Added support for compiling LIS as standalone executable. Added wrfout plugin to use WRF netCDF files as forcing for LIS. Added deep soil lapse rate option to adjust deep soil temperature in high terrain. Added dynamic deep soil temperature option to change as function of time-lagged skin temperature.
  - Updated Goddard microphysics to add variables rainncv_sepa and rainnc_sepa, and added multiple bug fixes.
  - Bug fixes to UPP for lightning threat product.
  - Several bug fixes to build system to compile GSDSU, LIS, and WRF on Pleiades. Also better handles netCDF and ESMF paths for LVT.
  - Added script support for plotting composite reflectivity, surface reflectivity, and skin temperature with RIP4.
  - Upgraded to Intel 13 compilers on Discover and Pleiades.

- Version 5-3.4 (Internal Beta Release)
  - Merged WRF 3.4, WPS 3.4, RIP 4.6.3, UPP 1.1, MET 4.0 (with patches through 29 June 2012), and PREP_CHEM_SOURCES 1.2_10apr2012.
  - Merged in GEOS2WRF 2.0.
  - Bug fix to CONVERT_EMISS to be compatible with PREP_CHEM_SOURCES.
  - Bug fixes to GOCART2WRF 2.0.
  - Added user defined tuning factors for GOCART dust emissions. Bug fixes for aerosols and AFWA GOCART dust emissions.
  - LIS bug fix for porosity: values from Noah and CLM2 LSMs now passed back to WRF and used in GOCART dust emissions.

  - Overhauled LISCONFIG program to process GEOGRID output instead of METGRID output.
  - Upgraded compilers to Intel 12.1 on Discover and Pleiades.

- Version 4-3.3.1 (Official Release)
  - Merged in WRF 3.3.1, WPS 3.3.1, ARWpost 3.1, RIP 4.6.2, MET 3.1, UPP 1.0, and PREP_CHEM_SOURCES v2_04aug2011 updates. Removed WPP.
  - Added GSDSU V3BETA.
  - Added WPS map projection support to PREP_CHEM_SOURCES, and added PLOT_CHEM utility to display output. Modified CONVERT_EMISS to support new 72-level GOCART background files with improved vertical positioning. Upgraded GOCART2WRF to version 2.0 (supporting GEOS-5 netCDF4 files), and removed support for old GEOS-4 netCDF3 files. Added porosity calculation as function of USGS land use for GOCART dust emissions.
  - Updated GEOS2WRF to read specific humidity from GEOS-5 HDF4 file and convert to relative humidity. Added bug fixes for LANDSEA and 2-meter relative humidity. Added scripting to ease use.
  - Added MERRA2WRF 2.0 to process HDF4 and netCDF MERRA files from the NASA Goddard Earth Sciences Data Information Services Center (GES DISC).
  - Fixed bug in RIP4 preventing processing of wrfout files from WRF-Chem.
  - Modified build system to optionally compile WRF-Chem with KPP.
  - Added sample batch scripts and input files for running GEOGRID, UNGRIB, METGRID, REAL, WRF, and RIP on Discover.

- Version 3-3.2.1 (Official Release)
  - Merged in WRF 3.2.1 and WPS 3.2.1. Added PREP_CHEM_SOURCES.
  - Merged in LIS 6.1rp1. Added LISCONFIG tool to customized lis.config file based on WRF domain and map projection settings. Added LVT.
  - Fixed GOCART2WRF bug triggered when WRF and GOCART pressures were identical, and added support for inner-nest domains. Added fromGEOS5_to_GEOS4 utility to convert on-line GEOS-5 GOCART data to off-line GEOS-4 GOCART format for GOCART2WRF.
  - Updated Goddard radiation and aerosol coupling component with bug fixes to prevent negative effective radii and division-by-zero, and handle spurious negative mixing ratios.
  - Added Goddard Microphysics-GOCART aerosol coupling.

  - WRF-Chem updates: Added capability of estimating SOA from biogenic terpene emissions, including three new variables: e_terp, e_api, e_lim. Linked MEGAN2 biogenic emissions scheme, GOCART dry deposition scheme, and various optical property schemes to GOCART aerosols. Linked RADM2 chemistry to GOCARTRADM2 option. Added namelist options for radiation-aerosol and microphysics-aerosol coupling (Goddard and GOCART). Added bio_emiss_soa namelist option to toggle emission conversion to SOA. Removed a number of chemistry variables from wrfout file.
  - GSDSU updates: Added GOCART input options to WRF input, GCE-SBM 3D option, SBM moment output, GrADS control file output, and Morrison two-moment support, plus bug fixes.
  - Build system updates: Set incremental building as the default option (instead of a performing a complete rebuild). Updated Discover configuration to use Intel MPI 4.0.1.007-beta, and added support for Pleiades. Added automatic detection of default configuration file on Discover and Pleiades. Add compilation of CONVERT_EMISS when 'chem' target is selected.

- Version 2-3.2.1 (Official Release)
  - Merged in MET bug fixes through 15 Feb 2011.
  - Updated LIS to version 6.0rp6.
  - WRF-LIS coupling: LIS export data not longer overwrites WRF data at water-points. Run-time checks for surface physics scheme setting in wrfinput disabled. REAL now built in special mode to generate consistent initial conditions for coupled WRF-LIS runs.
  - Goddard radiation and GOCART are coupled when running with WRF-Chem.
  - Severe weather diagnostics added to WPP for applicable physics schemes. Diagnostics include: Maximum 10-meter wind speed, column mean vertical velocity, max column integrated graupel, maximum lightning threat, derived radar reflectivity, precipitation accumulation for a given time window, and convective precipitation accumulation.
  - Changed MERRA2WRF to output relative humidity rather than specific humidity due to bug in REAL. Added improved error checking when calling the HDF4 library. Also, batch script for running on Discover changed to gracefully handle back PBS charge codes.
  - Changed GOCART2WRF to calculate correct tendencies at the final time level, and adds error checking when calling the netCDF library.
  - Wrote unified build system to compile all components of NU-WRF, targeting NASA's NCCS Discover system.

- Version 1-3.1.1 (Official Release)
  - Includes WRF 3.1.1, WPS 3.1.1, ARWpost 2.1, RIP 4.5, MET 2.0, WPP.
  - Includes LIS 6.0rp1, SDSU, SST2WRF 1.0, GEOS2WRF 1.0, MERRA2WRF 1.0
  - Includes simple build script for WRF on Discover.
  - Contains new NASA microphysics and radiation.
  - LIS integrated as WRF component.
  - Added severe weather diagnostics from NASA MSFC SPoRT.

3 Obtaining Software

3.1 Software Usage Agreement

The release of NU-WRF software is subject to NASA legal review, and requires users to sign a Software Usage Agreement. Toshi Matsui (toshihisa.matsui-1@nasa.gov) and Eric Kemp (eric.kemp@nasa.gov) are the points of contact for discussing and processing requests for the NU-WRF software.

There are three broad categories for software release:

1. **US Government – Interagency Release.** A representative of a US government agency should initiate contact and provide the following information:
   (a) The name and division of the government agency
   (b) The name of the Recipient of the NU-WRF source code
   (c) The Recipient's title/position
   (d) The Recipient's address
   (e) The Recipient's phone and FAX number
   (f) The Recipient's e-mail address
2. **US Government – Project Release under a Contract.** If a group working under contract or grant for a US government agency requires the NU-WRF source code for the performance of said contract or grant, then a representative should initiate contact and provide a *copy of the grant or contract cover page*. Information should include the following:
   (a) The name and division of the government agency
   (b) The name of the Recipient of the NU-WRF source code
   (c) The Recipient's title/position
   (d) The Recipient's address
   (e) The Recipient's phone and FAX number
   (f) The Recipient's e-mail address
   (g) The contract or grant number
   (h) The name of the Contracting Officer
   (i) The Contracting Officer's phone number
   (j) The Contracting Officer's e-mail address
3. **All Others.** Those who do not fall under the above two categories but who wish to use NU-WRF software should initiate contact to discuss possibilities for collaborating. Note, however, that NASA cannot accept all requests due to legal constraints.

3.2 Tar File

The Recipient will be provided a compressed tar file containing the entire NU-WRF source code distribution. (NU-WRF project members have access to tar files pre-staged on the NASA Discover and Pleaides supercomputers.) Two variants are available: gzip compression (*nu-wrf_v7lis7-3.5.1-p4.tgz*) and bzip2 compression (*nu-wrf_v7lis7-3.5.1-p4.tar.bz2*). Bzip2 is slightly more efficient but can take considerably longer to decompress.

To untar, type either **tar -zxvf nu-wrf_v7lis7-3.5.1-p4.tgz** or **bunzip2 nu-wrf_v7lis7-3.5.1-p4.tar.bz2 ; tar -xvf nu-wrf_v7lis7-3.5.1-p4.tar**. A *nuwrf_v7lis7-3.5.1-p4* directory should be created.

3.3 Subversion Repository

NU-WRF developers have the alternative of pulling code directly from the Subversion (SVN) repository. This approach requires several set-up steps to comply with NASA security requirements.

First, the developer will require an account on the NASA Center for Climate Simulations (NCCS) Discover supercomputer. The developer should refer to the NCCS website for details (http://www.nccs.nasa.gov/account_info.html#useradminforms).

Second, the developer should contact repository manager Eric Kemp (eric.kemp@nasa.gov) and provide (1) the NCCS username, and (2) the project being worked on. Confirmation from the NU-WRF Principal Investigator may be required before access is granted to the repository.

Third, the developer should create a SSH public key unless they have already created a key on Discover. To create a key, run **ssh-keygen -t rsa** on Discover. Note that RSA encryption is required.

Fourth, the developer should upload the ssh public key to the NCCS PROGRESS repository server (see https://progress.nccs.nasa.gov/keyupload). Note that it will take a few minutes for the uploaded public key to be recognized by the server.

Fifth, the developer should add a virtual host entry on Discover. Open or create the file *$HOME/.ssh/config* and add the following entry:

Host progressdirect
Hostname progress.nccs.nasa.gov
Port 22223

Once set-up, the developer can export the source code using the following command on Discover:

svn export svn+ssh://progressdirect/svn/nu-wrf/code/tags/releases/v7-3.5.1-p4

3.4 Directory Structure

The source code directory structure is as follows:

- The *ARWpost/*, *GSDSU/*, *ldt/*, *lvt/*, *MET/*, *RIP4/*, *UPP/*, *utils/*, *WPS/*, and *WRFV3/* folders contain the source codes of the components summarized in Section 2.1.
- The *docs/* folder contains documentation on the different NU-WRF components.
- The *oldcfg/* folder contains retired build config files for different platforms, compilers, and libraries. These are included to aid users in porting NU-WRF to a non-supported configuration.
- The *scripts/* folder contains sample batch scripts for running a number of NU-WRF component programs on the NASA Discover and Pleiades supercomputers (in the *discover/* and *pleiades/* subfolders, respectively). Sample input files for *RIP4* are also stored in the *rip/* subfolder. A script for creating new tags from the main development branch in the SVN repository is also included in the *devel/* subfolder.

The main directory also includes Bash scripts for the unified NU-WRF build system (discussed in Section 4.3) and a *CHANGELOG.TXT* file summarizing the changes to the overall modeling system.

4 Building Software

4.1 Compilers

The NU-WRF source code requires Fortran 90/2003, C, and C++ compilers. The current release officially supports Intel compilers 13.1.3.192 (ifort, icc, and icpc) on Discover and Pleiades. Previous NU-WRF versions have also been compiled using Portland Group (pgf90/pgf77, pgcc, and pgCC) and GNU (gfortran, gcc, and g++) compilers. (Note that GNU compilers do not currently build LDT or LVT, a situation we are addressing.)

4.2 External Libraries and Tools

A large number of third party libraries must be installed before building NU-WRF. Except as noted, the libraries must be compiled using the same compilers as NU-WRF, and it is highly recommended that *static* library files be created and linked rather than shared object. The list is as follows:

- A MPI library. (NU-WRF currently supports Intel MPI 5.0.2.044 and SGI MPT 2.11r13 on Discover nodes running SLES 11.3; Intel MPI 4.0.3.008 and MVAPICH2-2.0a2 on Discover nodes running SLES 11.1; and SGI MPT and Intel MPI 4.0.3.008 on Pleiades.)
- BUFRLIB 10.2.3.
- ESMF 5.2.0rp3 compiled with and without MPI support.
- FLEX (can use precompiled system binary on Linux).
- G2CLIB 1.4.0.
- GRIB_API 1.12.3.
- GSL 1.16.
- HDF4 4.2.6.
- HDF5 1.8.7.
- HDFEOS 2.18v1.
- JASPER 1.900.1.
- JPEG 6b.
- LIBPNG 1.2.50. (The newer 1.5.* versions *cannot* be used due to incompatibilities with WPS.)
- NCAR Graphics 6.0.0.
- NETCDF4 4.1.3 built with HDF5 compression. (Newer versions of netCDF *cannot* be used due to incompatibilities with MET 4.1.)

- YACC (can use preinstalled version on Linux).
- ZLIB 1.2.5.

In addition to the above libraries, NU-WRF requires Perl, Python, Bash, tcsh, GNU Make, Sed, Awk, M4, and the UNIX **compile** uname command to be available on the computer.

4.3 Build System

Each component of the NU-WRF modeling system has a unique compilation mechanism, ranging from simple Makefiles to sophisticated Perl and shell scripts. To make it easier for the user to create desired executables and to more easily resolve dependencies between components, NU-WRF includes a set of high level "wrapper" scripts for compilation. Each wrapper script is designed to be used by a common master script, and is customized to directly manage the component-specific build mechanism and inject appropriate configuration settings into that build mechanism. For example, the wrapper script for *WPS* will modify the *configure.wps* generated by **configure** [see Chapter 2 of (NCAR, 2014)] to update several library paths; the modified *configure.wps* is then used by the WPS **compile** script.

The build system allows users to specify **configure** options for the ARWPOST, RIP4, UPP, WPS, and WRFV3 components, as well as template *Makefile* names for the GEOS2WRF, GOCART2WRF, CASA2WRF, GSDSU, LISCONFIG, LVT, MET, PREP_CHEM_SOURCES, and SST2WRF components. Users also specify whether to compile UPP, WPS, and WRFV3 with MPI. These options are stored in the build config file (with names like *discover.cfg* and *pleiades.cfg*). This approach should aid in porting NU-WRF to new compilers, MPI implementations, and/or operating systems, a process discussed more fully in Appendix B.

With this system, the user can build executables by invoking a single Bash driver script called *build.sh* located in the top-level directory. This script accepts three types of command-line arguments:

- *Configuration.* The **--config** flag followed by the name of a configuration file specifying critical environment variables (e.g., the path to the netCDF library). Current configuration files are included in the top-level directory: *discover.cfg*, *discover_intel13_impi4_sp1.cfg*, *discover_intel13_mvapich2_sp1.cfg*, *discover_intel13_sgimpt_sp3.cfg pleiades.cfg*, and *pleiades_intel13_impi.cfg*. Users may develop their own configuration file to customize settings. If the configuration arguments are skipped, *build.sh* will default to either *discover.cfg* or *pleiades.cfg* based on the local environment, or exit if the software is on an unrecognized computer. Note that the configurations for Discover will check for specific operating system versions and abort if they are not found – this is to ensure consistency with the operating system on the desired compute nodes (currently SLES 11.3 on Haswell and SLES 11.1 on Sandy Bridge).

- *Options.* The user may specify **cleanfirst**, **debug**, and/or **nest=n** where **n** is an integer ranging from 1 to 3.
  - The **cleanfirst** option will cause the build system to "clean" a target (delete object files and static libraries) before starting compilation.
  - The **debug** option forces the WRFV3 or WPS build subsystems to use alternative compilation flags set in the configuration file (e.g., for disabling optimization and turning on run-time array bounds checking. This option is currently ignored by other NU-WRF components.
  - The **nest=n** option specifies compiling WRF with basic nesting (**n=1**), preset-moves nesting (**n=2**), or vortex-tracking nesting (**n=3**). Basic nesting is assumed by default. Note that WRF cannot be run coupled to LIS if preset-moves or vortex-tracking nesting is used. Similarly, WRF-Chem only runs with basic nesting.
  - The **profileMAP** option links WRF against special libraries required by the Allinea MAP commerical profiling tool (see www.allinea.com). This only works on Discover, and is only intended to perform special profiling runs for subsequent code modification and runtime improvement.
- *Targets.* The user can compile all or select executables with variations on chemistry support, as well as delete all executables, object files, and static libraries. The recognized targets are:
  - The **all** target compiles all executables without WRF-Chem.
  - The **allchem** target compiles all executables with WRF-Chem but without the Kinetic Pre-Processor.
  - The **allclean** target deletes all executables, object files, and static libraries.
  - The **allkpp** target compiles all executables with WRF-Chem and including the Kinetic Pre-Processor.
  - The **arwpost** target compiles executables in the *ARWpost* directory.
  - The **casa2wrf** target compiles executables in the *utils/casa2wrf* directory (both preprocessor pproc/ and casa2wrf in src/).
  - The **chem** target compiles executables in the *WRFV3* directory – except for LIS – with WRF-Chem support but without the Kinetic Pre-Processor. The compiled executables include CONV_EMISS.
  - The **geos2wrf** target compiles executables in the *utils/geos2wrf_2* directory (both GEOS2WRF and MERRA2WRF).
  - The **gocart2wrf** target compiles executables in the *utils/gocart2wrf_2* directory.
  - The **gsdsu** target compiles executables in the *GSDSU* directory.

- The **kpp** target compiles executables in the *WRFV3* directory – except for LIS – with WRF-Chem and Kinetic Pre-Processor support. The compiled executables include CONV_EMISS.
- The **lis** target compiles LIS in uncoupled mode in the *WRFV3/lis/make* directory.
- The **lisconfig** target compiles executables in the *utils/lisconfig* directory.
- The **lvt** target compiles executables in the *lvt* directory.
- The **merra2wrf** target compiles executables in the *utils/geos2wrf_2* directory (both MERRA2WRF and GEOS2WRF).
- The **met** target compiles executables in the *MET* directory.
- The **plot_chem** target compiles executables in the *utils/plot_chem* directory.
- The **prep_chem_sources** target compiles executables in the *utils/prep_chem_sources* directory.
- The **rip** target compiles executables in the *RIP4* directory.
- The **sst2wrf** target compiles executables in the *utils/sst2wrf* directory.
- The **upp** target compiles executables in the *UPP* directory.
- The **wps** target compiles executables in the *WPS* directory.
- The **wrf** target compiles executables in the *WRFV3* directory – except for LIS and CONV_EMISS – without WRF-Chem support.

One complication addressed by the build system is that the WPS and UPP components are dependent on libraries and object files in the *WRFV3* directory. To account for this, the build system has the following behavior:

- If both WPS and UPP are to be built, a check is made to ensure both are compiled with or without MPI. The decisions whether to compile with MPI are set in the *.cfg* file with environmental variables WPS_USE_MPI and UPP_USE_MPI; 1 indicates MPI and 0 indicate serial. If inconsistent MPI selections are found for WPS and UPP, the build system will abort with an error message.

- If WPS and/or UPP are to be built, *WRFV3* will be checked to ensure it was already built with a consistent MPI or non-MPI option. The *WRFV3* directory will also be checked to ensure the object files and static libraries required by WPS and/or UPP exist. If any of these checks fail, WRFV3 will be cleaned and automatically compiled with the appropriate MPI or non-MPI option. (Exception: If the *.cfg* file has an inconsistent MPI selection for WRFV3, the build system will abort.) *This will occur even if the **wrf** target is not listed on the command line.* The decision to compile WRF with MPI is determined by the WRF_USE_MPI environment variable set in the *.cfg* file; 1 indicates MPI and 0 indicates serial.

An additional complication is the coupling of LIS to WRF requires linking WRFV3 to the ESMF and ZLIB libraries. As a result, the *configure.wrf* file [see Chapter 2 of NCAR (2014)] is modified to link against these libraries. A similar modification occurs for UPP. (No modification is needed for WPS as long as WPS is compiled with GRIB2 support.)

The most straight-forward way to compile the full NU-WRF system on Discover or Pleiades is to type **./build.sh all** in the top level directory. If chemistry is required, the command is **./build.sh allchem** (**./build.sh allkpp** if KPP-enabled chemistry is needed). To fully clean the entire system, run **./build.sh allclean**. (Recall that the build script will automatically select **discover.cfg** or **pleiades.cfg**, as appropriate, if it detects the software is being compiled on Discover or Pleiades.) To explicitly specify the configuration file, type **./build.sh --config myconfig.cfg all**.

The user can selectively build components by listing specific targets. For example, to build the WRF model without chemistry along with WPS and UPP, type **./build.sh wrf wps upp**.

5 Front-End Workflows

In this section we will summarize several "front-end" workflows involving the main NU-WRF model and different pre-processors. (Post-processing is discussed in Section 6). The intent is to illustrate the roles of the pre-processors within the NU-WRF system, and to show several different configurations possible with NU-WRF (e.g., advanced land surface initialization, aerosol coupling, and CO_2 tracer simulation).

5.1 Basic Workflow

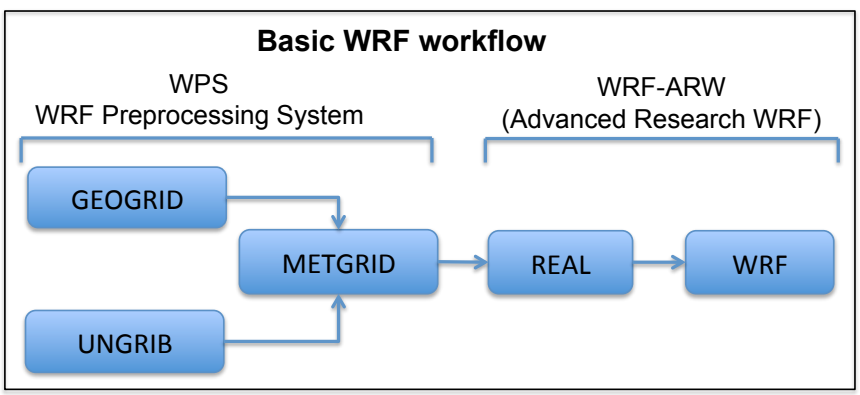
This is the most simple approach to running simulations with NU-WRF. Neither chemistry nor advanced land surface initialization are used, so the user should compile NU-WRF with **./build.sh wrf wps**.

- **WPS**: The user must edit a *namelist.wps* file to customize the WRF domains, set the start and end dates, set the file formats, and provide information on desired terrestrial data and file prefixes. A sample namelist.wps can be found in the *WPS/* directory. The user must then run the following programs [see Chapter 3 of NCAR (2014)].
  - **GEOGRID**. This program will interpolate static and climatological terrestrial data (land use, albedo, vegetation greenness, etc) to each WRF grid. The user should use the *GEOGRID.TBL.ARW* located in the *WPS/geogrid* directory to specify interpolation options for each dataset selected in *namelist.wps*. The user is also responsible for obtaining the *geog* dataset from NCAR for processing by GEOGRID. Sample run scripts are available in the *scripts/* directory.
  - **link_grib.csh**. This script is used to create symbolic links to the GRIB or GRIB2 files that are to be processed. The links follow a particular naming convention (*GRIBFILE.AAA*, *GRIBFILE.AAB*, ..., *GRIBFILE.ZZZ*) that is required by UNGRIB.
  - **UNGRIB**. This program will read GRIB or GRIB2 files with dynamic meteorological and terrestrial data (soil moisture, soil temperature, sea surface temperature, sea ice, etc) and write specific fields in WPS intermediate format. The user must select an appropriate *Vtable* file in *WPS/ungrib/Variable_Tables* to specify the fields to be extracted.
  - **METGRID**. This program will horizontally interpolate the output from UNGRIB to the WRF domains, and combine them with the output from GEOGRID. The user must select the *METGRID.TBL.ARW* file to specify the interpolation methods used by METGRID for each field.
- **REAL**. This program will vertically interpolate the METGRID output to the WRF grid, and create initial and lateral boundary condition files.

REAL is described in Chapters 4 and 5 of NCAR (2014). The user must edit a *namelist.input* file to specify the WRF domains, start and end times, and WRF physics configurations for REAL. *A standard WRF land use model should be selected for this workflow. No chemistry options can be selected.*

- **WRF**. This program will perform a numerical weather prediction simulation using the data from REAL. User need to change the *namelist.input* file for specific run cases. A sample *namelist.input* file can be found in the *WRFV3/run/* directory. WRF is described in Chapter 5 of NCAR (2014). In addition to the normal WRF physics options, the user can specify the new Goddard 3ICE or 4ICE microphysics (mp_physics=55 or 56), and the new Goddard 2011 or 2014 radiation schemes (ra_lw_physics=55 or 56 for longwave, and ra_sw_physics=55 or 56 for shortwave), all without aerosol coupling. Note that the 2014 radiation scheme requires users to create a symbolic link to the *WRFV3/GODDARDRAD_SSLUT* directory, and to put that link in directory where the model is run.

  A new feature added to NU-WRF is the calculation of mean integrated vapor transport. The user may adjust the time-averaging period for this diagnostic by changing the IVT_INTERVAL flag in the &time_control block of *namelist.input*. A value of 0 indicates instantaneous values will be output as the "means", while positive values indicates averaging time periods in minutes.



5.2 Land Surface Initialization and LIS Coupling

This is a more advanced approach to running simulations with NU-WRF. Instead of using land surface fields interpolated from a coarser model or reanalysis, a custom-made land surface state is created by LIS on the same grid and with the same terrestrial data and land surface physics as WRF. WRF will then call LIS on each advective time step, providing atmospheric forcing data and receiving land surface data (fluxes, albedo, etc) in return.

For simplicity, this workflow uses no chemistry, so the user should compile NU-WRF with **./build.sh wrf wps lis ldt lisconfig**. However, an advanced user can combine this workflow with one of the chemistry workflows described further down; in that case, NU-WRF should be compiled with **./build.sh chem wps lis ldt lisconfig** (or if using KPP, **./build.sh kpp wps lis ldt lisconfig**).

- **WPS**. These steps are identical to those WPS steps in Section 5.1.
- **LISCONFIG**. The user must provide a *ldt.config* file (used by LDT) and a *lis.config* file (used by LIS). The LISCONFIG software will read the *namelist.wps* file and the netCDF4 output files from GEOGRID, and copy the WRF grid information to the two config files. LISCONFIG is divided into two executables: *lisWrfDomain* (a Fortran compiled program found in *utils/lisconfig/bin*) and *lisWrfDomain.py* (a Python wrapper script found in *utils/lisconfig/scripts*).

  The software can be run as **./lisWrfDomain.py DOMAINPROG LISCONFIG LDTCONFIG WPSDIR**, where **DOMAINPROG** is the path to *lisWrfDomain*, **LISCONFIG** is the path to *lis.config*, **LDTCONFIG** is the path to *ldt.config*, and **WPSDIR** is the directory containing *namelist.wps* and the GEOGRID netCDF4 output files.
- **LDT**. The user must further customize the *ldt.config* file and a separate parameter attributes file to specify the static and climatological terrestrial data to be processed in "LSM parameter processing mode" [see NASA (2015a)]. For NU-WRF Version 7, the following settings are recommended/supported:
  - Noah.3.3 is the recommended land surface model;
  - UMD, USGS, and MODIS are supported land usage datasets;
  - GTOPO30 and SRTM are supported elevation datasets;
  - STATSGOFAO is the recommended soil texture dataset;
  - NCEP monthly climatological albedo and and max snow free albedo are recommended;
  - NCEP monthly climatological and maximum/minimum vegetation greenness are recommended;
  - NCEP slope type is recommended; and
  - ISLSCP1 deep soil temperature with terrain lapse-rate correction is recommended (not using the lapse rate correction could result in warm biases in high terrain).
- **LIS**. The user must further customize *lis.config* for a "retrospective" run. This includes specifying the start and end dates of the "spin-up" simulation, identifying the LDT datasets, specifying the land surface model, and identifying the atmospheric forcing datasets. The user must also customize a forcing variables list file compatible with the forcing dataset, and

a model output attributes file. All these files are described in more detail in NASA (2015b).

- **LDT**. After running LIS, it is necessary to rerun LDT in "NUWRF preprocessing for real" mode. This requires modifications to *ldt.config* to specify the static output file from LDT and the dynamic output file from LIS. Fields from both will be combined and written to a new netCDF output file for use by REAL.

- **REAL**. REAL is run similarly to the configuration in Section 5.1, except that it also reads the static and dynamic land surface data collected by LDT. For this to work, the *namelist.input* file must include an additional namelist block:

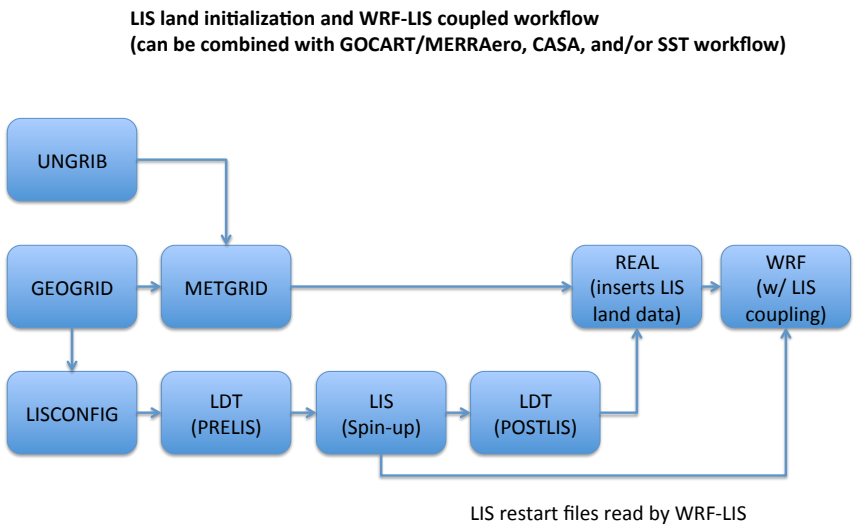
  ```
  &lis
  lis_landcover_type = 1,
  lis_filename = "lis_output.usgs.nc"
  /
  ```

  Here lis_landcover_type specifies the land use system used with LIS and LDT (1 = USGS, 2 = MODIS, 3 = UMD), and lis_filename is an array of character strings specifying the combined LDT/LIS files for each WRF domain.

  In addition, the user must specify LIS as the land surface model selection with WRF (i.e., set sf_surface_physics=55).

  The resulting initial and lateral boundary conditions will replace the land surface fields from UNGRIB with those from LDT/LIS.

- **WRF with LIS**. Running WRF in this case is similar to the basic case in Section 5.1, except that WRF will also read the *lis.config* file and the LIS restart files that were produced during the "retrospective" run. The user must modify *lis.config* to run in "WRF coupling mode", and specify *forcing_variables_wrfcplmode.txt* as the forcing variables list file. The start mode must also be changed to "restart", and the time step for each LIS domain must match that used with WRF (specified in *namelist.input*).



5.3 Use of GEOS-5 Meteorological Data

One source of initial and lateral boundary conditions is NASA's GEOS-5 global model (Rienecker et al., 2008). A number of dataset options exist from GEOS-5, including daily near-real-time simulations (see http://gmao.gsfc.nasa.gov/products), and archived reanalyses from the Modern-Era Retrospective Analysis for Research and Applications (MERRA) (Rienecker et al., 2011). A follow-up to MERRA (MERRA2) is also available to select users on the NASA Discover supercomputer, and is expected to be available to the general public within the next year. Finally, GEOS-5 can provide not just meteorological fields (temperature, pressure, wind, and moisture), but also aerosol fields due to the use of the GOCART aerosol module Chin et al. (2002).

There are several challenges to using GEOS-5 data. First, the GEOS-5 land surface data cannot be used to initialize WRF, due to fundamental differences in the GEOS-5 Catchment LSM Koster et al. (2000) and those in WRF. Users are therefore advised to use the GEOS-5 data in a workflow that also includes WRF-LIS (see Section 5.2 above). Second, GEOS-5 aerosol data cannot currently be handled by WPS and REAL, as these tools were designed for meteorological fields (temperature, pressure, wind, and moisture). Processing these aerosol fields requires a special workflow described in Section 5.5.

Remaining issues involve the format and organization of the GEOS-5 data. GEOS-5 outputs in netCDF (and historically HDF4 and HDFEOS2) instead of GRIB or GRIB2; GEOS-5 allows user-specification of variables and variable names for different output files, leading to wide variations between sim-

ulations; and GEOS-5 often does not output all the variables expected by WPS. To address these issues, special preprocessing software has been developed: *GEOS2WRF* (a collection of utilities designed for customized processing of GEOS-5 data, including derivation of missing variables), and *MERRA2WRF* (a monolithic program customized to process 6-hourly MERRA and MERRA2 reanalyses).

5.3.1 GEOS2WRF

GEOS2WRF can be broken down into four main sub-groups:

- **Front-end conversion**.
  - **GEOS2WPS**. A front end converter that can read HDF4, netCDF3, and netCDF4 files with GEOS-5 data. A *namelist.geos2wps* file is read in as input, and must be customized to list the location, name, and format of the GEOS-5 file; the names of the coordinate arrays in the GEOS-5 file; number of time slices, the indices of the slices, valid times and forecast hours; and the number of variables to process, along with their names, ranks, input and output names, units, and descriptions. This program takes the place of UNGRIB. The output from GEOS2WPS are written in WPS intermediate format, with the filename convention *$VARNAME_$LEVELTYPE:$YYYY-$MM-$DD_$HH*, where $VARNAME is the variable name, $LEVELTYPE is a string describing the type of level the data are on, $YYYY is the 4-digit year, $MM is the 2-digit month, $DD is the 2-digit day, and $HH is the 2-digit hour. Some example output file names:

    TT_MODEL_LEVEL:2009-08-25_00 # Temperature on model levels
    PSFC_GROUND_LEVEL:2009-08-25_00 # Surface pressure
    PMSL_MEAN_SEA_LEVEL:2009-08-25_00 # Mean sea level pressure
    VV_10M_ABOVE_GROUND_LEVEL:2009-08-25_00 # 10-meter V winds

    The $VARNAMEs (TT, PSFC, PMSL, and VV above) are listed in *namelist.geos2wps*, and can be customized by the user; however, they must match the values in the *METGRID.TBL* look-up file used by METGRID for those variables to be processed by WPS. (Intermediate variables used to derive other variables for WPS do not have this naming restriction.)

- **Temporal interpolation**.
  - **temporalInterpolation**. This program takes WPS intermediate file format data and linearly interpolate in time. This can be used, for example, to interpolate 6-hourly MERRA2 analyses at 00Z, 06Z, 12Z, and 18Z to 3-hourly intervals for initializing WRF at 03Z, 09Z,

15Z, or 21Z. Only a single, user-specified variable will be processed during a particular program invocation, and other variables in the input data files will be ignored. A *namelist.temporalInterpolation* file is used to specify the variable, input and output data files.

- **Variable-derivation**. Multiple tools for deriving missing variables required by WRF from existing variables. These should be used on a as-needed basis depending on the contents of the GEOS-5 files. Current programs that are in this category are:
  - **createSOILHGT**. A utility that reads in a WPS file with surface geopotential, and calculates the surface terrain field. The output WPS file will be named *SOILHGT_GROUND_LEVEL:$YYYY-$MM-$DD_$HH*. A *namelist.createSOILHGT* file is also used as input.
  - **createHGT**. A utility that reads in a WPS file with model layer pressure thicknesses, model layer temperatures, model layer specific humidity, and the model terrain field, and derives the geopotential heights on the GEOS-5 model levels. The output WPS files will be named *HGT_MODEL_LEVEL:$YYYY-$MM-$DD_$HH*. A *namelist.createHGT* file is also used as input. *This program is not needed when processing isobaric levels.*
  - **createLANDSEA**. A utility that reads in a WPS file with "lake fraction" and "ocean fraction" and derives a land-sea mask. The output WPS files will be named *LANDSEA_GROUND_LEVEL:$YYYY-$MM-$DD_$HH*. A *namelist.createLANDSEA* file is also used as input.
  - **createPRESSURE**. A utility that reads in a WPS file with model layer pressure thicknesses, and calculates the (mid-layer) pressures. The output WPS files will be named *PRESSURE_MODEL_LEVEL:$YYYY-$MM-$DD_$HH*. A *namelist.createPRESSURE* file is also used as input. *This program is not needed when processing isobaric levels.*
  - **createRH**. A utility that reads in a WPS file with either model or isobaric level temperatures, specific humidity, and pressure, plus optional surface pressure, 2-meter temperature, and 2-meter specific humidity, and derives relative humidity on those levels. The output WPS files will have prefixes of *RH_2M_ABOVE_GROUND_LEVEL*, *RH_MODEL_LEVEL*, and/or *RH_ISOBARIC_LEVEL*, and will end with the familiar *$YYYY-$MM-$DD_$HH* string. A *namelist.createRH* file is also used as input. *This program is recommended* because some versions of REAL do not correctly interpolate specific humidity, and because the WRF definition of RH is strictly w.r.t. liquid while some versions of GEOS-5 output a weighted average of RT w.r.t. liquid and ice that is a function of temperature.
- **Extrapolation**.

  - **extrapIsobaric**. A utility that reads in a WPS file with geopotential height, temperature, relative humidity, U and V winds all on isobaric levels, and extrapolates to those levels that are underground. The RH, U, and V nearest the ground is simply copied downward, while a specified lapse rate is used for temperature and the hypsometric equation is used for geopotential height. The output WPS files will be called *ISOBARIC:$YYYY-$MM-$DD_$HH* and will contain all the isobaric data (original data above ground, extrapolated data below ground.) A *namelist.extrapIsobaric* file is also used as input. *This program is not necessary when processing GEOS-5 model level data, since the GEOS-5 coordinate is terrain following. Users are advised to use the model level data whenever possible.*

- **Splitter utility**.
  - **splitWPS**. A utility that reads in a WPS file and divides the data into new WPS files, which each file containing a single 2D slab of data. The output WPS files will be called *$VARNAME_$LEVEL:$YYYY-$MM-$DD_$HH*, where $LEVEL is the "level code" for the slab. The "level code" follows WPS convention: pressure levels are simply the pressure in Pa; model levels are the indices of the slice ("1" indicates model top in GEOS-5); ground level, 2-meter AGL, and 10-meter AGL are represented as "200100"; and mean sea level is represented as "201300". A *namelist.splitWPS* is also used as input. *This program is not required for preparing data for WPS, but instead allows breaking up a WPS file into individual fields for examination.*

To proceed, the user must first compile the GEOS2WRF software with **./build.sh geos2wrf**. The user must then review the GEOS-5 data available to them and identify time slices and date/time stamps of interest, and the variables that can be used as-is by WRF. WRF will ultimately require the following fields on either isobaric or GEOS-5 model levels:

- pressure;
- geopotential height;
- horizontal winds;
- temperature; and
- moisture (preferably relative humidity w.r.t. liquid).

Recommended fields that are useful for interpolating or extrapolating near the WRF model terrain level include:

- surface pressure;
- sea level pressure;

- land-sea mask;
- sea-ice fraction;
- 2-m temperature;
- 2-m relative humidity;
- 10-m horizontal winds;
- skin temperature; and
- terrain height.

With this list in mind, the user must also identify GEOS-5 variables that can be used derive other variables for WRF. From the utilities listed above, the following derivations can be made:

- Surface geopotential can be used to derive terrain height (via createSOIL-HGT).
- Lake fraction and ocean fraction can be used together to derive a land-sea table (via createLANDSEA).
- Model layer pressure thicknesses can be used to derive model layer pressures (via createPRESSURE).
- Model layer pressure thicknesses can also be used (with model layer temperatures, model layer specific humidity, and the model terrain field) to derive model layer geopotential heights (via createHGT).
- Relative humidity on model levels, isobaric levels, and near ground level can be derived from model, isobaric, and 2-meter temperatures, model, isobaric, and 2-meter specific humidity, and model, isobaric, and surface pressure (via createRH).
- Isobaric temperature, relative humidity, U and V winds can be extrapolated underground (via extrapISOBARIC).

After assembling the list of variables, the user should run GEOS2WPS using a customized *namelist.geos2wps* for each GEOS-5 file. Execution occurs with a simple **./geos2wps** if in the current directory.

After extracting all the GEOS-5 variables, the user must employ the necessary utilities to derive the remaining variables for WRF. The appropriate namelist file (e.g., *namelist.createHGT*) must be customized, and the user must use the UNIX **cat** command to collect the relevant WPS files together. When ready, the user will execute by typing the program name (e.g., **./createHGT**).

The *namelist.geos2wps* file contains the following information:

Variable Names	Description
&files	
geosFileFormat	Integer, specifies GEOS-5 file format HDF4=1, netCDF3 or netCDF4 = 2, HDFEOS2=4.
geosFileName outputDirectory	String, specifies GEOS-5 input file name to read String, directory name to write WPS file.
&coordinates	
longitudeName	String, name of 1-D longitude array in GEOS-5 file.
latitudeName	String, name of 1-D latitude array in GEOS-5 file.
hasVertical-Dimension	Logical, specifies whether data with vertical dimension are to be processed from GEOS-5 file.
verticalName	String, name of 1-D vertical coordinate array in GEOS-5 file.
&forecast	
numberOfTimes	Integer, number of time slices to process from GEOS-5 file.
validTimes(:)	Array of Strings, specifies valid time(s) of each time slice to process. Format is $YYYY-$MM-$DD_$HH. One array entry should exist for each time slice.
timeIndices(:)	Array of Integers, specifies time slice indices to process. One array entry should exist for each time slice.
forecastHours(:)	Array of Integers, specifies nominal forecast hour length for each processed time slice. One array entry should exist for each time slice.
&variables	
numberOfVariables	Integer, specifies total number of variables to process from the GEOS-5 file.
variableRanks(:)	Array of Integers, specifies the ranks (number of dimensions) for each GEOS-5 variable to process. Data of rank 3 are assumed to be organized as (lat,lon,time), while rank 4 data are assumed to be organized as (lat,lon,vert,time). One array entry should be assigned for each processed variable.
variableLevel-Types(:)	Array of Integers, specifies level type for each processed variable. One array entry should be assigned for each variable. = 1, ground level ;= 2, 2-meters AGL = 3, 10-meters AGL ; = 4, mean sea level = 11, model level ; = 12, isobaric level

Variable Names	Description
&variables	
variableNamesIn(:)	Array of Strings, specifies name of each processed variable in GEOS-5 file. One array entry should be specified for each variable.
variableNames-Out(:)	Array of Strings, specifies name of each processed variable as written in the WPS file. One array entry should be specified for each variable. Not that if a processed variable is intended for direct use by WPS (instead of use in deriving something else), the variableNamesOut entry should match that in *METGRID.TBL* file used by METGRID.
variableUnits(:)	Array of Strings, specifies units of each processed GEOS-5 variable. One array entry should be specified for each variable. This is included because some GEOS-5 variables are known to be assigned the wrong units when output by the model.
variable-Descriptions(:)	Array of Strings, gives short descriptions of each processed variable as written in the WPS file. One array entry should be specified for each variable.
&subsetData	
subset	Logical, specifies whether to process entire GEOS-5 domain or to read and process a subset.
iLonMin	Integer, specifies minimum i (longitude) index of GEOS-5 grid to process. Only used if subset=.true.
iLonMax	Integer, specifies maximum i (longitude) index of GEOS-5 grid to process. Only used if subset=.true.
jLatMin	Integer, specifies minimum j (latitude) index of GEOS-5 grid to process. Only used if subset=.true.
jLatMax	Integer, specifies maximum j (latitude) index of GEOS-5 grid to process. Only used if subset=.true.
kVertMin	Integer, specifies minimum k (vertical) index of GEOS-5 grid to process. Only used if subset=.true.
kVertMax	Integer, specifies maximum k (vertical) index of GEOS-5 grid to process. Only used if subset=.true.

The *namelist.temporalInterpolation* file contains the following information:

Variable Names	Description
&all	
fieldName	String, lists name of variable to process.
&input1	
directory1	String, lists directory with WPS intermediate file.
prefix1	String, lises prefix of name of WPS intermediate file.
year1	Integer, lists valid year of WPS intermediate file.
month1	Integer, lists valid month of WPS intermediate file.
day1	Integer, lists valid day of WPS intermediate file.
hour1	Integer, lists valid hour of WPS intermediate file.
&input2	
directory2	String, lists directory with WPS intermediate file.
prefix2	String, lises prefix of name of WPS intermediate file.
year2	Integer, lists valid year of WPS intermediate file.
month2	Integer, lists valid month of WPS intermediate file.
day2	Integer, lists valid day of WPS intermediate file.
hour2	Integer, lists valid hour of WPS intermediate file.
&output	
directoryOutput	String, lists directory with WPS intermediate file.
prefixOutput	String, lises prefix of name of WPS intermediate file.
yearOutput	Integer, lists valid year of WPS intermediate file.
monthOutput	Integer, lists valid month of WPS intermediate file.
dayOutput	Integer, lists valid day of WPS intermediate file.
hourOutput	Integer, lists valid hour of WPS intermediate file.

The *namelist.createSOILHGT* file contains the following information:

Variable Names	Description
&input	
directory	String, directory for input and output WPS files.
prefix	String, lists filename prefix of input WPS files.
year	Integer, lists valid year of WPS file.
month	Integer, lists valid month of WPS file.
day	Integer, lists valid day of WPS file.
hour	Integer, lists valid hour of WPS file.
surfaceGeopotential-Name	String, name of the surface geopotential field in WPS file.

The *namelist.createHGT* file contains the following information:

Variable Names	Description
&input	
directory	String, directory for input and output WPS files.
prefix	String, lists filename prefix of input WPS files.
year	Integer, lists valid year of WPS file.
month	Integer, lists valid month of WPS file.
day	Integer, lists valid day of WPS file.
hour	Integer, lists valid hour of WPS file.
layerPressure-ThicknessName	String, name of pressure thickness variable between GEOS-5 model levels in the input WPS file.
layerTemperature-Name	String, name of model layer temperatures in the input WPS file.
layerSpecific-HumidityName	String, name of model layer specific humidity variable in the input WPS file.
soilHeightName	String, name of surface terrain variable in the input WPS file.
modelTopPressure	Real, air pressure (in PA) at very top of GEOS-5 grid. For GEOS-5, this is typically 1 Pa (0.01 mb).

The *namelist.createLANDSEA* file contains the following information:

Variable Names	Description
&input	
directory	String, directory for input and output WPS files.
prefix	String, lists filename prefix of input WPS files.
year	Integer, lists valid year of WPS file.
month	Integer, lists valid month of WPS file.
day	Integer, lists valid day of WPS file.
hour	Integer, lists valid hour of WPS file.
lakeFractionName	String, name of the GEOS-5 variable specifying fraction of grid point covered by lakes in the WPS input file.
oceanFractionName	String, GEOS-5 variable name specifying fraction of grid point covered by ocean in the WPS input file.

The *namelist.createPRESSURE* file contains the following information:

Variable Names	Description
&input	
directory	String, directory for input and output WPS files.
prefix	String, lists filename prefix of input WPS files.
year	Integer, lists valid year of WPS file.
month	Integer, lists valid month of WPS file.
day	Integer, lists valid day of WPS file.
hour	Integer, lists valid hour of WPS file.
layerPressure-ThicknessName	String, names variable with pressure thicknesses between GEOS model levels in the WPS input file.
modelTopPressure	Real, air pressure (in PA) at very top of GEOS-5 grid. For GEOS-5, this is typically 1 Pa (0.01 mb).

The *namelist.createRH* file contains the following information:

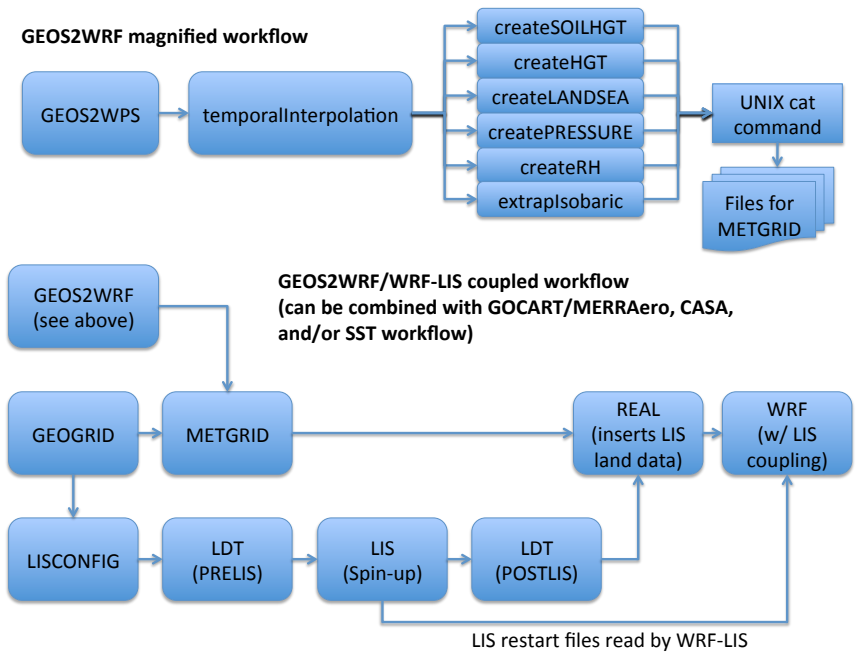
Variable Names	Description
&input	
directory	String, lists directory for input and output WPS files.
prefix	String, lists filename prefix of input WPS files.
year	Integer, lists valid year of WPS file.
month	Integer, lists valid month of WPS file.
day	Integer, lists valid day of WPS file.
hour	Integer, lists valid hour of WPS file.
processSurface-Pressure	Logical, indicates whether or not to read in surface pressure from the WPS input file.
onIsobaricLevels	Logical, indicates whether upper air levels are isobaric instead of model level.
surfacePressure-Name	String, name of surface pressure variable in WPS input file. Ignored if processSurfacePressure=.false.
pressureName	String, name of upper-level pressure fields in WPS input file. Ignored if onIsobaricLevels=.true.
temperatureName	String, name of temperature fields in WPS input. file. If 2-meter temperatures are included, then the surface pressure must also be supplied and processSurfacePressure must be set to .true.
specificHumidity-Name	String, name of specific humidity fields in WPS input file. If 2-meter specific humidities are included, then the surface pressure must also be supplied and processSurfacePressure must be set to .true.

The *namelist.extrapIsobaric* file contains the following information:

Variable Names	Description
&input	
directory	String, lists directory for input and output WPS files.
prefix	String, lists filename prefix of input WPS files.
year	Integer, lists valid year of WPS file.
month	Integer, lists valid month of WPS file.
day	Integer, lists valid day of WPS file.
hour	Integer, lists valid hour of WPS file.
geopotentialHeight-Name	String, name of isobaric geopotential height fields in WPS input file.
temperatureName	String, name of isobaric temperature fields in the WPS file.
relativeHumidity-Name	String, name of isobaric relative humidities in the WPS input file.
uName	String, name of isobaric zonal wind field in WPS input file.
vName	String, name of isobaric meridional wind field in WPS input file.

The *namelist.splitWPS* file contains the following information:

Variable Names	Description
&input	
directory	String, lists directory for input and output WPS files.
prefix	String, lists filename prefix of input WPS files.
year	Integer, lists valid year of WPS file.
month	Integer, lists valid month of WPS file.
day	Integer, lists valid day of WPS file.
hour	Integer, lists valid hour of WPS file.



A sample script (*scripts/discover/run_geos2wrf_merra2_3hrassim.sh*) is available to use GEOS2WRF to process 3-hourly MERRA2 assimilation data. These data are output from GEOS-5 when the global model is adjusting toward a MERRA2 6-hourly analysis via Incremental Analysis Updates [see section 4.2 of Rienecker et al. (2008)]. This script will run GEOS2WPS, createLANDSEA, createSOILHGT, and createRH to process the data. To run, the user must edit the accompanying *config.discover.sh* file to set the path to the NUWRF code, the work directory, and the modules used to compile GEOS2WRF; then, the *run_geos2wrf_merra2_3hrassim.sh* should be modified to specify the start and end dates and hours to process. (Users who wish to use 6-hourly MERRA or MERRA2 data can use either GEOS2WRF or MERRA2WRF; however, the 3-hourly MERRA2 data can only be processed with GEOS2WRF.)

5.3.2 MERRA2WRF

MERRA2WRF is a monolithic program customized to process the 6-hourly reanalyses from MERRA and MERRA2. It was first developed for MERRA under the assumption that the archived data files (called "collections" in GEOS-5 terminology) would be permanent, making it possible to build a single robust preprocessing tool. Recently support was added for 6-hourly MERRA2 fields; however, 3-hourly MERRA2 processing is not possible due to significant differences in the data collections (users must fall back to GEOS2WRF for these 3-hourly data).

MERRA files are accessible from the NASA Goddard Earth Sciences Data and Information Services Center web page (http://disc.sci.gsfc.nasa.gov/daac-

bin/DataHoldings.pl), and are available to the general public. At the time of this writing, MERRA2 fields are only accessible on the NASA Discover supercomputer in */discover/nobackup/projects/gmao/merra2/merra2/scratch/*, and are only available to select users authorized by the GMAO.

MERRA2WRF must be compiled using **./build.sh merra2wrf**. The following files must then be gathered from the MERRA or MERRA2 datasets:

- **const_2d_asm_Nx** (in HDFEOS2 or NETCDF format):
  - 'XDim' or 'lon' (longitude)
  - 'YDim' or 'lat' (latitude)
  - 'PHIS' (surface geopotential)
  - 'FRLAKE' (lake fraction)
  - 'FROCEAN' (ocean fraction)
- **inst6_3d_ana_Nv** (variable names are HDF4 or netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hour)
  - 'levels' or 'Height' or 'lev' (nominal pressure for each model level)
  - 'ps' or 'PS' (surface pressure)
  - 'delp' or 'DELP' (layer pressure thicknesses)
  - 't' or 'T' (layer temperature)
  - 'u' or 'U' (layer eastward wind)
  - 'v' or 'V' (layer northward wind)
  - 'qv' or 'QV' (layer specific humidities)
- **inst6_3d_ana_Np** (variable names are HDF4 or netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hours)
  - 'slp' or 'SLP' (sea level pressure)
- **tavg1_2d_slv_Nx** (variable names are HDF4 or netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hours)
  - 'u10m' or 'U10M' (10-meter eastward wind)
  - 'v10m' or 'V10M' (10-meter northward wind)

  - 't2m' or 'T2M' (2-meter temperature)
  - 'qv2m' or 'QV2M' (2-meter specific humidity)
  - 'ts' or 'TS' (skin temperature)

- **tavg1_2d_ocn_Nx** (variable names are HDF4/netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hours)
  - 'frseaice' or 'FRSEAICE' (sea ice fraction)

Note that the tavg1_2d_slv_Nx and tavg1_2d_ocn_Nx collections are 1-hour averages that are valid at the bottom of the hour. For simplicity, MERRA2WRF uses the 00:30Z average data with the 00Z instantaneous fields, the 06:30Z average data with the 06Z instantaneous fields, and so on.

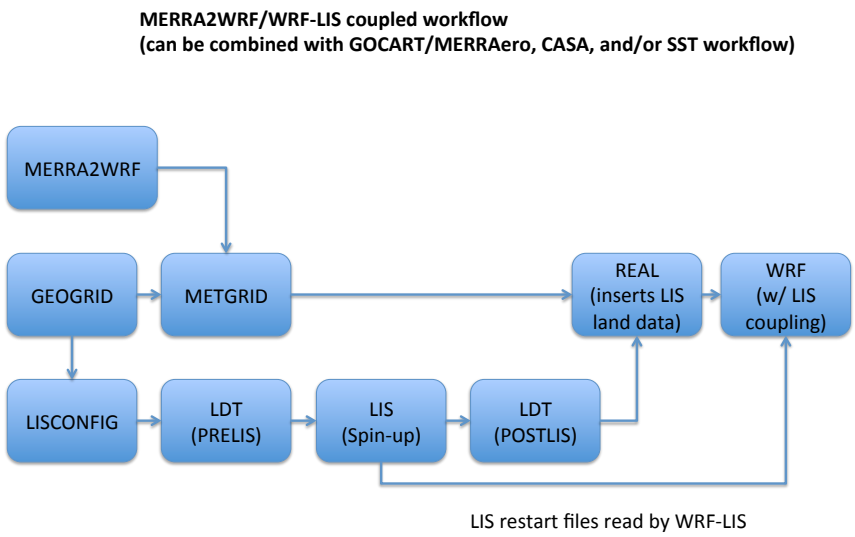
User can use *utils/geos2wrf_2/RUN_MERRA/Run_MERRA.csh* to ftp the MERRA data and run MERRA2WRF from a specified start date and end date using the command **Run_MERRA.csh StartDate EndDate OutputDir NUWRFDIR**. A namelist file will be created for each processing date, and files readable for METGRID will be generated.

To use MERRA2 reanalysis, user can use *utils/geos2wrf_2/RUN_MERRA2/Run_MERRA2.csh* to copy the MERRA2 data and run MERRA2WRF from a specified start date and end date using the command **Run_MERRA2.csh StartDate EndDate OutputDir NUWRFDIR**. A namelist file will be created for each processing date, and files readable for METGRID will be generated.

Alternatively, the user can customize the *utils/geos2wrf_2/namelist/namelist.merra2wrf* to process the selected MERRA data and *namelist.merra2_2wrf* to process the selected MERRA2 data. The namelist files consists of a single block:

Variable Names	Description
&input	
outputDirectory	String, lists directory for writing WPS output files.
merraDirectory	String, lists directory containing MERRA or MERRA2 input files.
merraFormat_const_2d_asm_Nx	Integer, specifies format of const_2d_asm_Nx file. 1=HDF4, 2=netCDF, 4=HDFEOS2.
merraFile_const_2d_asm_Nx	String, name of const_2d_asm_Nx file.
numberOfDays	Integer, lists number of days to process. Each MERRA or MERRA2 collection. (excluding const_2d_asm_Nx) will have one file per day.
merraDates(:)	Array of strings, list each day to be processed (format is YYYY-MM-DD).
merraFormat_inst6_3d_ana_Nv	Integer, specifies format of inst6_3d_ana_Nv files. 1=HDF4, 2=netCDF, 4=HDFEOS2.
merraFiles_inst6_3d_ana_Nv(:)	Array of strings, specifying names of inst6_3d_ana_Nv files.
merraFormat_inst6_3d_ana_Np	Integer, specifies format of inst6_3d_ana_Np files. 1=HDF4, 2=netCDF, 4=HDFEOS2.
merraFiles_inst6_3d_ana_Np(:)	Array of strings, specifying names of inst6_3d_ana_Np files.
merraFormat_tavg1_2d_slv_Nx	Integer, specifies format of tavg1_2d_slv_Nx files. 1=HDF4, 2=netCDF, 4=HDFEOS2.
merraFiles_tavg1_2d_slv_Nx(:)	Array of strings, specifying names of tavg1_2d_slv_Nx files.
merraFormat_tavg1_2d_ocn_Nx	Integer, specifies format of tavg1_2d_ocn_Nx files. 1=HDF4, 2=netCDF, 4=HDFEOS2.
merraFiles_tavg1_2d_ocn_Nx(:)	Array of strings, specifying names of tavg1_2d_ocn_Nx files.

The software is run by typing **./merra2wrf namelist.merra2wrf**. The output files will be named *MERRA:$YYYY-$MM-$DD_$HH*, where $YYYY is the four-digit year, $MM is the two-digit month, $DD is the two-digit day, and $HH is the two-digit hour. These files are readable by METGRID.



5.4 Use of New Erodible Soil Options

NU-WRF includes several new options for specifying erodible soil (EROD) for dust emissions. The workflow depends a bit on the particular option selected, but requires compilation of WRF-Chem and WPS (**./build.sh chem wps** if using normal chemistry, or **./build.sh kpp wps** if using KPP chemistry).

The four available EROD options are:

- **EROD_STATIC**. This is the EROD option inherited from the community WRF. Annual EROD at 0.25 deg resolution for sand, silt, and clay is processed and fed to WRF-Chem.
- **EROD_MDB**. This is a new seasonal EROD dataset derived from MODIS-Deep Blue climatological aerosol products. [See Ginoux et al. (2012) and Ginoux et al. (2010) for description of estimating frequency of occurrance of optical depth – these are converted to EROD]. Data are subdivided into three groups (for sand, silt, and clay) at 0.1 deg resolution for four meteorological seasons (December-January-February, March-April-May, June-July-August, September-October-November). These are processed and passed to WRF-Chem.
- **EROD_DYN_CLIMO**. This is a new "dynamic climatological" EROD option. It uses a monthly surface bareness field derived at 30 arc second resolution from the community WRF climatological MODIS vegetation fraction dataset (greenness_fpar_modis), with adjustments from the community WRF's soil type and MODIS and USGS land use datasets to

screen out water bodies. It also uses a 30 arc second topographic depression dataset derived from the community WRF's terrain dataset. These fields are passed to WRF-Chem, which will create an instantaneous EROD field from these variables with adjustments to screen out snowy or very cold locations.

- **EROD_DYN**. This is a new "dynamical" EROD option. It uses the same topographic depression field as **EROD_DYN_CLIMO**, plus a 0.01 deg bareness field based on the NASA SPoRT daily NDVI product over the United States. For consistency, SPoRT NDVI-based greenness is also processed to ensure consistency between greenness and bareness. The data are passed to WRF-Chem, which will construct an instantaneous EROD field from the bareness and topographic depression with adjustments to screen out snowy or very cold locations.

A sample workflow for EROD is presented here:

- **Assemble GEOG data**. Several new EROD-related fields must be obtained from the NU-WRF group and placed in subdirectories with the standard GEOG data available with the community WRF. For **EROD_MDB**, they are *erod_mdb_clay_0.1deg*, *erod_mdb_sand_0.1deg*, and *erod_mdb_silt_0.1deg*. For **EROD_DYN_CLIMO**, they are *bareness_dyn_climo* and *TOPODEP_30s*. For **EROD_DYN**, the *TOPODEP_30s* data must be staged; in addition, the user must obtain NDVI-based greenness in GEOGRID format from NASA SPoRT and place the files in a *gvfsport* directory.

- **Create NDVI-based bareness for EROD_DYN option**. An experimental program called ndviBareness4Wrf can be obtained from Eric Kemp (eric.kemp@nasa.gov) to read a NASA SPoRT 0.01 deg NDVI-greenness product and convert to WPS intermediate format files with *BARE_DYN* file prefixes.

- **GEOGRID**. Run for terrestrial processing. Use the *GEOGRID.TBL.ARW_CHEM_NUWRF* file to ensure EROD related fields are specified. If preparing for the **EROD_DYN** option, "gvfsport" should be specified as the first part of the "geog_data_res" option in *namelist.wps* – this will ensure the SPoRT greenness is processed rather than the climatological greenness data available with the community WRF.

- **UNGRIB**. Run for normal GRIB file processing.

- **METGRID**. Run for normal processing. If preparing for the **EROD_DYN** option, user should modify *namelist.wps* to include the *BARE_DYN* file prefix for the bareness data in the "fg_name" namelist option. Note that the *METGRID.TBL.ARW* file in NU-WRF has been modified to recognize the new EROD-related fields.

- **REAL**. Run to produce initial and lateral boundary conditions. Namelist variable "chem_opt" should be set to 401 (for simple dust treatment).

New namelist variable "erod_option" in the &chem block should be set to "static", "mdb", "dyn_climo", or "dyn". If not set, "static" is assumed.

- **WRF**. Run for EROD simulation. Optionally set the new "gocart_dustemiss_layer" namelist variable to highest vertical model level to spread dynamic dust (default is 1), or set "gocart_dustemiss_suppress=1" to shut off dynamic dust emissions. The new variable EROD_TIMESTEP in the wrfout netCDF file will show the instantaneous EROD field for sand, silt, and clay for whatever EROD option is used. Other variables of note are:
  - BARENESS_DYN_CLIMO: Monthly input bareness field for **EROD_DYN_CLIMO** option.
  - EROD_MDB_CLAY: Seasonal EROD for clay for **EROD_MDB** option.
  - EROD_MDB_SAND: Seasonal EROD for sand for **EROD_MDB** option.
  - EROD_MDB_SILT: Seasonal EROD for silt for **EROD_MDB** option.
  - EROD_STATIC: The standard EROD field available with the community WRF.
  - TOPODEP: The topographic depression values for sand, silt, and clay for the **EROD_DYN_CLIMO** and **EROD_DYN** options.

5.5 Use of GOCART or MERRA Aerosol Data

NU-WRF offers advanced aerosol modeling using the implementation of GOCART [see Chin et al. (2002) and Ginoux et al. (2001)] in WRF-Chem. Running GOCART in WRF allows for aerosol coupling with the Goddard 3ICE and 4ICE microphysics schemes and with the 2011 and 2014 Goddard radiation schemes, providing simulation of the direct and indirect aerosol effects on weather and climate. For best results, it is necessary to provide initial and lateral boundary conditions for GOCART, plus surface based emissions. To that end, the NU-WRF modeling system includes the new GOCART2WRF preprocessor for providing chemical boundary conditions from the GEOS-5, and includes the community PREP_CHEM_SOURCES program for emissions. To run, the user must compile with **./build.sh chem wps gocart2wrf prep_chem_sources plot_chem**.

MERRAero data (Kishcha et al., 2014) also can be processed with the GOCART2WRF preprocessor.

A workflow supporting use of GOCART or MERRAero in NU-WRF might look like this:

- **WPS**. Perform terrestrial and meteorological preprocessing as normal.
- **REAL**. Generate meteorological initial and lateral boundary conditions as normal.

- **GOCART2WRF**. Process GEOS-5/GOCART data and insert into output files from REAL. The user must edit a *namelist.gocart2wrf* to specify the number of WRF domains, location of REAL output files, and location and file prefixes of the GEOS-5/GOCART netCDF4 files. GOCART2WRF is then executed at the command line using **./gocart2wrf** with *namelist.gocart2wrf* in the current working directory. GOCART2WRF will obtain the required dates and times from the REAL netCDF files, search the GEOS-5/GOCART files for the corresponding dates and times, read and interpolate the required GOCART variables, and essentially append those fields to the REAL files. Currently 17 GOCART variables are processed: Hydrophobic and Hydrophilic Black Carbon, Hydrophobic and Hydrophilic Organic Carbon, dust particles with 0.5 μm, 1.4 μm, 2.4 μm, 4.5 μm, and 8.0 μm effective radii, sea salt particles with 0.3 μm, 1.0 μm, 3.2 μm, and 7.5 μm effective radii, and concentrations of Dimethyl Sulfide, Methanesulfonic Acid, Sulfur Dioxide, and Sulfate.

  MERRAero data can be used in place of GOCART data with minor changes in the *namelist.gocart2wrf* files for directory name specifications. The *namelist.gocart2wrf* file contains the following entries:

Variable Names	Description
&wrf	
max_dom	Integer, specifies number of WRF domains.
wrf_dir	String, specifies directory with wrfinput and wrfbdy files.
&gocart_shared	
gocart_format	Integer, specifies GEOS version of GOCART or MERRAero data. Currently must be set to 5 (for GEOS-5 netCDF4 files).
gocart_dir	String, specifies directory with GEOS-5 netCDF4 GOCART or MERRAero files.
gocart_prefix	String, specifies file name prefix for GEOS-5 netCDF4 GOCART or MERRAero files.

- **PREP_CHEM_SOURCES**. This community tool processes a number of biogenic, anthropogenic, volcanic, and wildfire emissions (Freitas et al., 2011). Operating this program is largely described in ESRL (2013) and requires customizing a *prep_chem_sources.inp* file and downloading emissions data supported by the program. The NU-WRF version has several modifications:

  - **Map projection**. The map projection code from WPS has been added to PREP_CHEM_SOURCES to ensure consistency in emission interpolation. This support is automatic when using the NU-WRF build system, and no user action is required.
  - **Improved GOCART background fields**. Processing emissions for GOCART requires monthly climatological background fields of

Hydrogen Peroxide, Hydroxide, and Nitrate. The community data for PREP_CHEM_SOURCES contains 55 vertical levels and is missing some information useful for vertical interpolation. A new, 72-level dataset has been added with improved interpolation. The user should edit *prep_chem_sources.inp* and set the new gocart_bg_data_type variable to "old" or "new" to switch between the two. Note that the new 72-level files must be in a gocart_bg_new subdirectory on the same level as the gocart_bg subdirectory containing the old 55-level files.

- **Improved interpolation of GOCART background fields**. The community PREP_CHEM_SOURCES program uses simple averaging of GOCART background grid points to each WRF grid point, which implicitly assumes the WRF grid is at a coarser resolution than the background. This results in unphysical blocky fields, which are further marred by gradients associated with sharp WRF terrain. In the NU-WRF version of PREP_CHEM_SOURCES, bilinear interpolation is used in cases where less than two background grid points are averaged to a WRF grid point, leading to much smoother fields.
- **GFEDv3.1**. The GFEDv3.1 biomass burning emissions dataset [van der Werf et al. (2010) and Randerson et al. (2013)] is supported. The user must edit the *prep_chem_sources.inp* file, toggle the new variable use_gfedv3=1, set the new gfedv3_data_dir variable to specify the directory with the GFEDv3.1 data, and edit the new gfedv3_suffix variable to list the species to process (e.g., "BC,OC,PM2p5,SO2") Note that the species are used to construct the names of the monthly emissions files (e.g., *GFED3.1_200301_BC.txt*). Also, note that GFEDv3.1 and GFEDv2 data cannot be used simultaneously. Currently the following species can be processed: BC, C2H4, C2H4O, C2H4, C2H5OH, C2H6S, C2H6, C3H6O, C3H6, C3H8, CH4, C5H8, CH2O, CH3OH, CH4, CO2, CO, NH3, NOx, OC, PM2p5, SO2, Terpenes, Toluene_lump, and TPM. DM can also be processed but is not currently used by WRF-Chem.
- **QFED**. The NASA QFED wildfire emissions dataset [Darmenov and da Silva (2013)] is supported. The user must edit the *prep_chem_sources.inp* file, toggle the new variable use_qfed=1, set the new qfed_data_dir variable to specify the top directory with QFED data (yearly subdirectories like Y2005 are assumed, with monthly subdirectories like M05 within each annual subdirectory), and edit the new qfed_suffix variable to list the species to process (e.g, "bc,oc,pm25,so2"). Note that the species are used to construct the names of the emissions files (e.g, *qfed2.emis_bc.005.20140404.nc4*, so the names are case sensitive. Also, note that QFED cannot be used simultaneously with GFEDv3.1 or GFEDv2. Currently the following QFED species can be processed: acet, ald2, alk4, bc, c2h6, c3h6, c3h8, ch2o, ch4, co, co2, mek, nh3, no, oc, pm25, and so2.
- **Output of map projection data**. New *.map* files with map pro-

jection data are automatically output. These files are intended for use by PLOT_CHEM to visualize the fields.

- **PLOT_CHEM**. This is an optional step to create simple visualizations of emissions output from PREP_CHEM_SOURCES. The program reads in a GrADS control file produced by PREP_CHEM_SOURCES, the corresponding GrADS binary file, and the special *.map* file with critical map projection information. PLOT_CHEM will then create visualizations of each field using NCAR Graphics. The plots are not publication quality and are only intended for sanity checking. To run, the user must first create a symbolic link *grads.ctl* to the desired GrADS control file, and then run **./plot_chem** in the same directory as the GrADS and *.map* files. The output is a *gmeta* file which can be viewed using the NCAR Graphics **idt** program (see http://ngwww.ucar.edu/ for information on NCAR Graphics).

- **CONVERT_EMISS**. This is a community WRF-Chem preprocessor that takes the output from PREP_CHEM_SOURCES and rewrites the fields in new netCDF files for reading by WRF-Chem. This program is described in ESRL (2013) and requires modifying a *namelist.input* file to specify domain information, physics, and chemistry options. The program is then run in the same directory as the *namelist.input* and the PREP_CHEM_SOURCES output files.

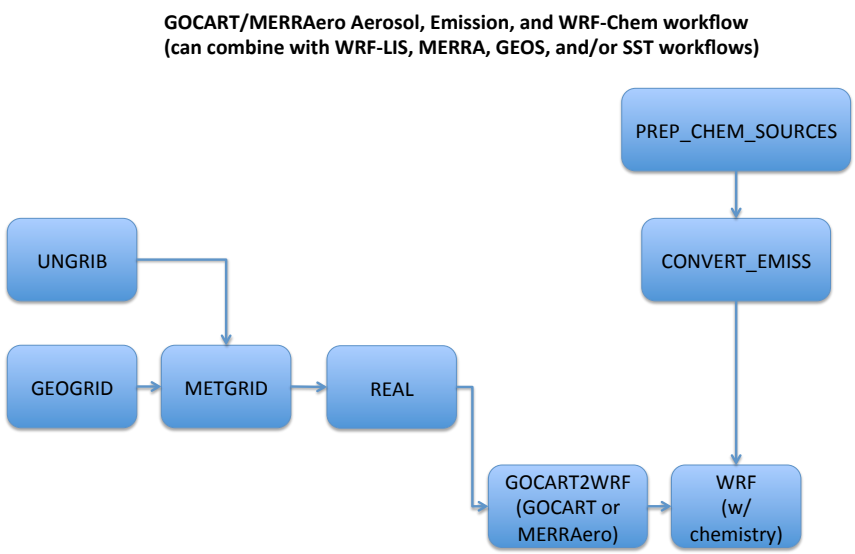
  There are two issues to keep in mind:

  - First, CONVERT_EMISS does not currently support processing more than one domain at a time. Thus, the user must process each domain in separate executions, and must rename the input files to use "d01" before execution, regardless of what the actual domain number is. The output files must then be renamed to restore the actual domain numbers for WRF-Chem. In addition, a *namelist.input* file must be customized for each execution with the max_dom variable set to "1" and the first grid domain variables (e.g., dx, nx, etc) conforming to whichever domain is being processed.
  - Second, the output files from PREP_CHEM_SOURCES must be renamed to conform to the naming convention expected by CONVERT_EMISS. The naming convention for the different emissions files are documented in ESRL (2013).

  The NU-WRF version of CONVERT_EMISS includes support for the improved GOCART background fields and most GFEDv3.1 and QFED species without any special action required by the user. However, if the user processes C2H4, CH4, CO2, terpenes, or DM emissions from GFEDv3.1 with PREP_CHEM_SOURCES – or ch4 or co2 from QFED – then the new *namelist.input* variable gfedv3_biomass_burn_extravars in the &chem block must be set to 1. This will cause CONVERT_EMISS to

read the additional fields at the end of the *emissfire* binary file (these are not produced by the community version of PREP_CHEM_SOURCES, so reads are not attempted by default to preserve backward compatibility).

- **WRF-Chem**. Running WRF-Chem is similar to the basic case in section 5.1, but requires the &chem namelist block to be included in the *namelist.input* with with GOCART activated (chem_opt = 300, 301, 302, or 303). Aerosol coupling with the Goddard 3ICE or 4ICE schemes (mp_physics=55 or 56) will be activated if the new gsfcgce_gocart_coupling namelist variable is set to 1 (set by default). Likewise, aerosol coupling with the NUWRF or 2014 Goddard radiation schemes (ra_lw_physics=55 or 56 and ra_sw_physics=55 or 56) will be activated if the new gsfcrad_gocart_coupling namelist variable is set to 1 (set by default).

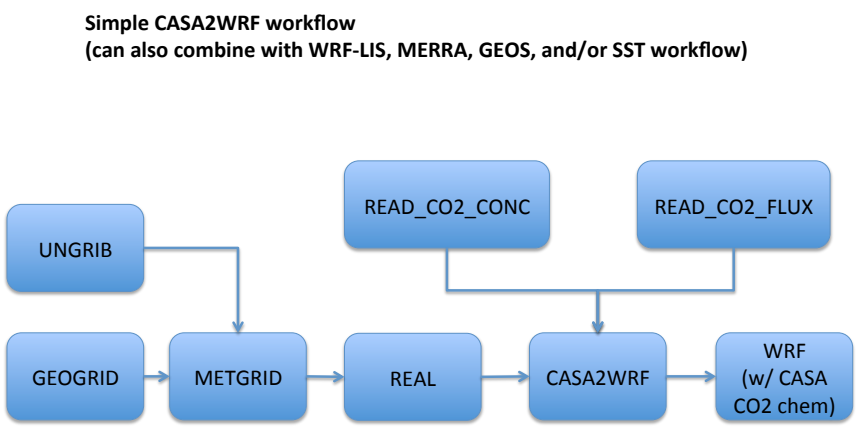


5.6 Use of CASA CO_2 Data

NU-WRF has a new capability for running simulations with CO_2 treated as a tracer (i.e., no interaction with physics). This requires specifying initial, lateral boundary, and flux emission fields of CO_2. To that end, several utilities have been written to process CASA global climatological CO_2 concentrations and flux emissions and provide them to WRF-Chem: READ_CO2_CONC, READ_CO2_FLUX, and CASA2WRF. To compile, the user must type **./build.sh casa2wrf**.

Instructions for running these programs, including namelist definitions, are provided in Tao et al. (2014). A sample workflow is provided below:

- **WPS**. Perform terrestrial and meteorological preprocessing as normal.
- **REAL**. Generate meteorological initial and lateral boundary conditions as normal.
- **READ_CO2_CONC**. Reads the CASA CO_2 concentration files in flat binary format, and converts to netCDF with a time stamp.
- **READ_CO2_FLUX**. Reads the CASA CO_2 flux files in flat binary format, and converts to netCDF with a time stamp.
- **CASA2WRF**. Read the CO_2 netCDF files, interpolates concentration and flux data to the WRF grids (single or nested), calculates the rates of change of flux/hour at the user specified time frequency, appends the interpolated concentrations to the initial and lateral boundary condition netCDF files (*wrfinput* and *wrfbdy*), and writes the interpolated fluxes to a new netCDF file (*CO2_domain_date*).
- **WRF-Chem**. Run WRF-Chem with CASA CO_2 chemistry options in the *namelist.input* (including chem_opt = 18, emiss_opt=18, and emiss_inpt_opt=18). Other appropriate settings are listed in Tao et al. (2014).



5.7 Use of RSS Sea Surface Temperature Data

A special preprocessor included with NU-WRF is SST2WRF, which processes several sea surface temperature (SST) products produced by Remote Sensing

Systems (RSS; see http://www.remss.com). These products are potential alternatives to the SST or skin temperature fields often provided in meteorological GRIB files (e.g., from the NOAA GFS or NAM models). Because RSS products are not available in GRIB format, UNGRIB cannot process them and a tool like SST2WRF is required as a substitute.

SST2WRF currently supports several different analysis products classified by source instrument and by algorithm version. The instrument SST analyses are

- **mw_ir**. 9-km global SST valid at 1200 UTC based on microwave (TMI, AMSR-E, AMSR2, WindSat) and Infrared (Terra MODIS, Aqua MODIS) data.
- **mw**. 25-km global SST valid at 0800 LT, based on Microwave (TMI, AMSR-E, AMSR2, WindSat) data.

The algorithm versions are:

- **rt**. The real-time algorithm.
- **v04.0**. Version 4 algorithm.

See http://www.remss.com/measurements/sea-surface-temperature/oisst-description for a description of these products.

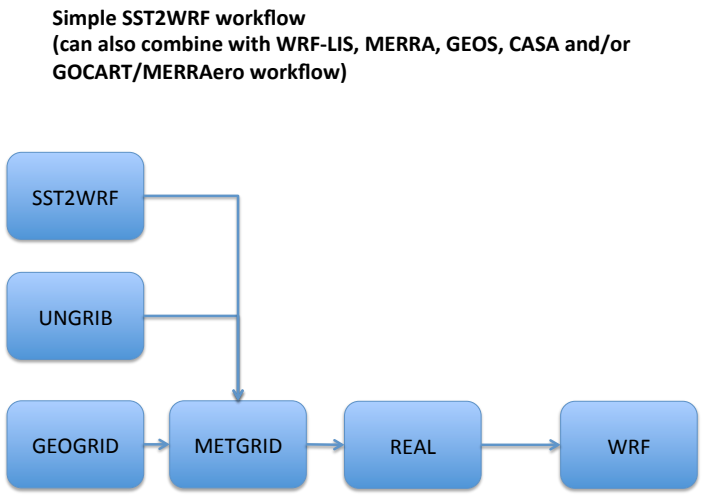
A workflow for SST2WRF could be similar to that in section 5.1, but would require running SST2WRF *in addition to* UNGRIB. UNGRIB is responsible for processing meteorological fields, while SST2WRF will process only the SST fields from RSS. (One could also replace UNGRIB with GEOS2WRF or MERRA2WRF; for simplicity, we will assume UNGRIB is used.)

The user must compile using **./build.sh sst2wrf**. A script given in *utils/sst2wrf/scripts/Run_SST.csh* can be used to ftp SST data and process them by typing *Run_SST.csh start-date enddate instrument_type*. Alternatively, the user can customize a sample namelist file given in *utils/sst2wrf/namelist/namelist.sst2wrf* file to provide the following information:

Variable Names	Description
&input	
instrument	String, specifies instrument(s) used for analysis; options are "mw_ir", "mw".
year	Integer, specifies valid year of analysis.
dayOfYear	Integer, specifies valid day of year of analysis.
version	String, specifies algorithm for analysis; options are "rt","v04.0".
inputDirectory	String, specifies directory with SST file.
&output	
outputDirectory	String, specifies directory for output file.
prefixWPS	String, specifies file name prefix for output; prefix must also be in *METGRID.TBL* for METGRID to process.
&fakeoutput	
numFakeHours	Integer, specifies number of hours of each day that additional WPS files should be written for. Currently only one SST analysis is available per day, but METGRID requires all time varying fields to have same time interval. Thus, we optionally output daily SST at multiple times corresponding to when atmospheric data are available.
fakeHours	Array of integers, specifies nominal hours of day in UTC for an input daily SST analysis to be output.

The program is then run by typing **./sst2wrf** in the same directory as *namelist.sst2wrf*.

The resulting output files will be in WPS intermediate format. The user must then edit *namelist.wps* and list both the meteorological (from UNGRIB) and SST (from SST2WRF) file prefixes in the "fg_name" namelist variable [see Chapter 3 of NCAR (2014)]. METGRID will replace the SST or skin temperature from UNGRIB with that from SST2WRF. The remaining steps (REAL and WRF) can be completed as normal.


Simple SST2WRF workflow
(can also combine with WRF-LIS, MERRA, GEOS, CASA and/or GOCART/MERRAero workflow)
SST2WRF
UNGRIB
GEOGRID
METGRID
REAL
WRF

6 Post-Processors

In Section 5, sample workflows are presented to initialize and run WRF in multiple configurations. In the present section we address the question of post-processing the WRF (and LIS) output. All of the post-processors address the task of evaluation, either subjective or objective. Several tools are available to prepare visualizations of model fields, while others allow for calculating verification metrics against observations or gridded analyses.

It should be noted that other generic tools exist that can be used to evaluate NU-WRF output. These include:

- NCL (http://www.ncl.ucar.edu);
- R (http://www.r-project.org);
- Python (http://www.python.org) with the Matplotlib library (http://matplotlib.org) or PyNGL and PyNIO libraries (https://www.pyngl.ucar.edu);
- VAPOR (https://www.vapor.ucar.edu); and
- Ncview (http://meteora.ucsd.edu/ pierce/ncview_home_page.html).

6.1 G-SDSU

The Goddard Satellite Data Simulator Unit [G-SDSU; Matsui et al. (2014)] is a program developed by NASA for use with high-resolution weather model data. The program can simulate multiple microwave, radar, visible and infrared, lidar, and broadband satellite products from the input model fields. These simulations can be used for detailed verification against actual satellite observations (Matsui et al., 2009), for assimilation of satellite radiances, or for exploring future satellite missions. The software is compiled by typing **./build.sh gsdsu**. Instructions on running the program are available in Matsui and Kemp (2014).

6.2 RIP4

The community Read/Interpolate/Plot Version 4 software package is capable of processing WRF netCDF files, deriving new variables (e.g., air temperature, relative humidity, CAPE), interpolating to isobaric, isentropic, or constant height levels as well has vertical cross-sections, and plotting the fields in NCAR Graphics *gmeta* format. Advanced options also exist, including calculating and plotting trajectories, interpolating between coarse and fine grid resolutions, and writing data in a format readable by the Vis5D visualization package (see http://vis5d.sourceforge.net).

The RIP software is compiled by typing **./build.sh rip**. The two most important executables in RIP4 are:

- **RIPDP_WRFARW**. This program will read WRF netCDF files and transform the data to an internal binary data format. The user will have

the option of processing either a basic set of variables or all the variables in the files.

- **RIP**. This program will process the output of RIPDP_WRFARW based on the user's settings in a provided input file.

Users are referred to Stoelinga (2006) for detailed instructions on using RIP. Sample namelist files are included in the NU-WRF package in *scripts/rip*.

6.3 ARWPOST

The ARWPOST program is a post-processor developed by NCAR for converting WRF-ARW netCDF data into GrADS format. Analogous to RIP, ARWPOST supports derivation of certain variables from the model output, and interpolation of those fields to isobaric or constant height levels. GrADS (see http://www.iges.org/grads) can then be used to visualize the data. The program is compiled by typing **./build.sh arwpost**. Instructions for running ARWPOST are given in Chapter 9 of NCAR (2014).

6.4 UPP

The UPP program is the "Unified Post-Processor" developed by NOAA NCEP for processing all NCEP model data. As with RIP and ARWPOST, UPP can read WRF netCDF output files, derive a number of meteorological fields from the provided model data, and interpolate to user specified levels. In the case of UPP, the data are output in GRIB format. The program is compiled by typing **./build.sh upp**. Instructions for running UPP are given in Chapter 7 of DTC (2014).

The NU-WRF version of UPP includes several modifications provided by NASA SPoRT. These are experimental severe weather diagnostics:

- **Instantaneous Lighting Threat 1**. Based on grid-resolved graupel flux at -15C. Specified as "LIGHTNING THREAT 1" in *parm/wrf_cntrl.parm* file.

- **Instantaneous Lightning Threat 2**. Based on vertically integrated ice. Specified as "LIGHTNING THREAT 2" in *parm/wrf_cntrl.parm* file.

- **Instantaneous Lightning Threat 3**. Based on Threat 1 and 2 products. Specified as "LIGHTNING THREAT 3" in *parm/wrf_cntrl.parm* file.

- **Interval Maximum Lighting Threat 1**. Based on grid-resolved graupel flux at -15C. Specified as "MAX LTG THREAT 1" in *parm/wrf_cntrl.parm* file.

- **Interval Maximum Lightning Threat 2**. Based on vertically integrated ice. Specified as "MAX LTG THREAT 2" in *parm/wrf_cntrl.parm* file.

- **Interval Maximum Lightning Threat 3**. Based on Threat 1 and 2 products. Specified as "MAX LGT THREAT 3" in *parm/wrf_cntrl.parm* file.

6.5 MET

The MET software is a community meteorological verification toolkit developed by the DTC. This is a generic tool for comparing gridded model forecasts and analyses against numerous observations – METARs, Mesonets, rawinsondes, MODIS satellite data, and Air Force cloud analysis data. MET expects the model data to be in GRIB format, a requirement that forces the user to run UPP on the WRF output first (see Section 6.4). Observation data formats include PREPBUFR and MADIS. With this input data, MET can be used for a number of different meteorological verifications, including point-to-point verification, object-oriented verification, and wavelet verification. Numerous statistical measures can be calculated with confidence intervals, and plotting capabilities are available.

The MET software is compiled by typing **./build.sh met**. Thorough instructions on running the software are provided in DTC (2013a) and DTC (2013b).

6.6 LVT

LVT is a NASA developed land surface verification toolkit. It is intended to compare LIS output files against numerous in-situ, remotely sensed, and reanalysis products. Fields that can be evaluated include surface fluxes, soil moisture, snow, and radiation. Multiple verification metrics can be calculated, and advanced features include data masking, time series, temporal averaging, and analysis of data assimilation impacts. The software is compiled by typing **./build.sh lvt** or **./build.sh all**. Detailed instructions on running LVT can be found in NASA (2015c).

A Frequently Asked Questions

Q: What modules should I use on Discover and Pleiades when running NU-WRF?

A: Load the identical modules that you used to compile NU-WRF. The current defaults on Discover are (assuming Bash shell):

```
source /usr/share/modules/init/sh
module purge
unset LD_LIBRARY_PATH
module load comp/intel-13.1.3.192
module load mpi/impi-5.0.2.044
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib64
```

And on Pleiades:

```
source /usr/share/modules/init/sh
module purge
unset LD_LIBRARY_PATH
module load comp/intel-2013.5.192
module load mpi-sgi/mpt
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib64
```

If you use a non-default build configuration file, compare that file to the above and change the above modules accordingly. Likewise adjust the syntax if you are using a shell other than bash.

The module settings should be done in your shell if you are running a program at the command line. If instead you are launching a batch job to SLURM or PBS, the above settings should be in the batch script, so the commands are run on the job's compute node.

Q: What other environment settings should I use?

A: Make sure to set stack size to unlimited. This can help prevent memory allocation errors for automatic arrays. If using bash:

```
ulimit -s unlimited
```

If using csh:

```
limit stacksize unlimited
```

Q: What namelist settings should I use with WRF?

A: It's not possible to provide a single configuration optimal for all types of simulations (LES, regional climate, cloud system resolving, chemical transport, etc), but we have recommendations that provide a reasonable first-guess.

Category	Selection	Namelist	Explanation
Microphysics	NUWRF Goddard 2011 3ICE with Graupel	mp_physics=55 gsfcgce_hail=0 gsfcgce_2ice=0	Most stable and generally applicable NASA option.
Radiation	NUWRF Goddard 2011 Radiation	ra_lw_physics=55 ra_sw_physics=55	Most stable NASA option.
Aerosol Coupling	GOCART simple aerosol	chem_opt=300 gsfcgce_gocart_coupling=1 gsfcrad_gocart_coupling=1 vertmix_onoff=1 chem_conv_tr=1 dust_opt=1 seas_opt=1 dmsemis_opt=1	Simple aerosol, coupled with radiation and microphysics, no gas chemistry.
LSM	LIS with Noah	sf_surface_physics=55 num_soil_layers=4	Spin-up LIS on WRF grid for detailed initial fields; cannot use moving nests; must have lis.config file
PBL	MYNN2	bl_pbl_physics=5 sf_sfclay_physics=5	Replaces MYJ PBL scheme; used in RAPv2 and HRRR; reportedly gives unbiased PBL depth, moisture, and temperature.
Cumulus	G3	cu_physics=5 ishallow=1	Third-gen Grell scheme; tackles "grey zone"; compatible with used in RAPv2; handles shallow cumulus.

Category	Selection	Namelist	Explanation
Diffusion	2nd Order on coordinate surfaces; eddy coefficient based on deformation	diff_opt=1 km_opt=4	Appropriate for real data cases (dx $\geq$ 1 km)
6th-order horizontal diffusion	Monotonic	diff_6th_opt=2 diff_6th_factor=0.12	Removed 2*dx noise in light winds; can be tuned.
Advection	5th-order, positive-definite	moist_adv_opt=1 scalar_adv_opt=1 tke_adv_opt=1 chem_adv_opt=1 momentum_adv_opt=1 h_mom_adv_order=5 h_sca_adv_order=5 v_mom_adv_order=5 v_sca_adv_order=5	Used in RAPv2 and HRRR; positive-definite prevents negative mixing ratios from original non-negative values.
Rayleigh Damping	Implicit	damp_opt=3 zdamp=5000. dampcoef=0.2	Prevents gravity waves from reflecting off model top; designed for real-data cases; used in RAPv2 and HRRR.
Vertical Velocity Damping	Activated	w_damping=1	Damps updrafts approaching CFL limit; best used for long or quasi-operational runs.
Time Off-Centering	Tuning factor	epssm=0.1	Controls vertically-propagating sound waves; set to max slope of model terrain.
Nesting	1-Way	feedback=0	2-way nesting does not work with LIS, and can lead to strange results with high-res nesting.

The above recommendations have some caveats:

- The NUWRF 3ICE scheme can be configured to use hail instead of graupel, but the hail setting is best suited for deep continental convection cases (e.g., cases with supercells). The NUWRF 4ICE scheme is intended

to replace the 3ICE and cover all scenarios, but it is still experimental and does not currently support aerosol coupling.

- The NUWRF 2014 Goddard radiation scheme should eventually replace the 2011 version, but is still experimental and undergoing calibration.
- The PBL and cumulus settings are the most uncertain physics selections, and are mostly picked because of their use in the NCEP RAPv2 and HRRR models which speak to their robustness. In addition, WRF-Chem requires a Grell cumulus scheme for most simulations, placing these schemes at an advantage over other popular choices for cumulus such as Kain-Fritsch and Betts-Miller-Janjic.
- 6th-order diffusion was added to WRF because the normal diffusion scheme is tied to the wind speed, and can insufficiently smooth the fields in light winds. Users running a short case may wish to turn off the 6th-order scheme to see if 2*dx features develop in the vertical velocity and divergence fields without it.
- A popular alternative to the positive-definite advection filter is the monotonic choice, which damps both positive and negative spikes in the advected fields (the positive-definite only damps negative spikes). Unfortunately monotonic advection may lead to excessive smoothing when 6th-order diffusion is also turned on. The user may wish to experiment with monotonic advection and turning off 6th-order diffusion, particularly with short chemistry runs where winds are not too light.
- Vertical velocity damping is artificial and is recommended mostly for situations where CFL violations are particularly unwelcome (e.g., quasi- operational runs).
- 2-way nesting cannot be used with WRF-LIS coupling because the feedback routine may change the land/sea mask for a parent WRF grid to better match the child WRF grid.

Q: What settings should I use with GEOGRID?

A: For GEOGRID, we recommend these settings:

- Use MODIS land use data instead of USGS. This requires changing the geog_data_res variable in *namelist.wps* to something like:

  geog_data_res = 'modis_30s+10m','modis_30s+2m','modis_30s+30s',

  GEOGRID will check the *GEOGRID.TBL* settings to relate the selections to each dataset (terrain, soil type, etc). The 'modis_30s' will only match with the land-use data and will force processing of MODIS data; the remaining data types will fall back on '10m', '2m', or '30s' for the respective WRF grid. See Chapter 3 of (NCAR, 2014) for more information.

- Process all EROD data with GEOGRID. This requires use of the new *GEOGRID.TBL.ARW_CHEM_NUWRF* table which lists entries for four different EROD datasets. The user does not need to decide which EROD option to use until running REAL.

Q: Why does WRF or REAL fail with this error about NUM_LAND_CAT?

——————— ERROR ———————-

namelist : NUM_LAND_CAT = 20

input files : NUM_LAND_CAT = 24 (from geogrid selections).

————— FATAL CALLED —————

FATAL CALLED FROM FILE: ⟨stdin⟩ LINE: 709

Mismatch between namelist and wrf input files for dimension NUM_LAND_CAT

A: There are two possible causes.

First, you are running WRF without LIS coupling, your land use data is coming from GEOGRID, and your namelist settings for land use are inconsistent between GEOGRID, REAL, and WRF. Normally these programs expect USGS data (with 24 categories). If you configure GEOGRID to process MODIS instead, you must set NUM_LAND_CAT in namelist.input to 20 for consistency.

Second, you are trying to run WRF coupled with LIS, REAL is replacing the landuse data from GEOGRID with that from LDT, and your namelist.input is not consistent. In this case, NUM_LAND_CAT should match the value from GEOGRID when you run REAL, but should match the value from LDT when you run WRF. (The reference to geogrid in the error message from WRF is incorrect in this case, and stems from the community WRF not knowing anything about LDT.) Note that LDT can provide USGS (24 categories), MODIS (20 categories), or UMD (14 categories).

B Porting NU-WRF

Currently NU-WRF is only supported using Intel compilers and using Intel MPI (Discover and Pleiades), SGI MPT (Discover or Pleiades), or MVAPICH2 (Discover). The underlying software should, however, run on other systems as long as the appropriate tools (compilers, MPI implementation, make, Perl, csh, bash, etc.) are available. Users who wish to port NU-WRF will need to take the following steps:

- Libraries:
  - Compile the libraries listed in section 4.2.
  - Determine the paths to the *yacc* binary and the *flex* library. Make sure *yacc* and *flex* are in your path.
  - Copy *discover.cfg* to a new top level build config file (called here *newconfig.cfg*, but any name can be used).
  - Edit *newconfig.cfg* to update the library paths.
  - Edit *newconfig.cfg* to change modules for system binaries and libraries (compilers, MPI, etc). If the Modules package is not installed on your system, comment out the module commands and explicitly edit the PATH and LD_LIBRARY_PATH environment variables.
- ARWpost:
  - Inspect and edit *ARWpost/arch/configure.defaults* to ensure a block exists for the desired operating system, hardware, and compilers. Note that ARWpost is serial only (no MPI support).
  - Run **ARWpost/configure** at the command line to identify the integer value of the appropriate build selection.
  - Edit *newconfig.cfg* to enter the configure option as environment variable ARWPOST_CONFIGURE_OPT.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg arwpost** to test the build.
- CASA2WRF:
  - Create a new makefile template in directory *utils/casa2wrf* to specify compilers and compiler flags.
  - Edit *newconfig.cfg* to enter the new makefile template name as environmental variable CASA2WRF_MAKEFILE.
  - In the top NU-WRF directory run **./build.sh --config newconfig.cfg casa2wrf** to test the build.
- GEOS2WRF:

- Create a new makefile template in directory *utils/geos2wrf_2* to specify compilers and compiler flags.
- Edit *newconfig.cfg* to enter the new makefile template name as environmental variable GEOS2WRF_MAKEFILE.
- In the top NU-WRF directory, run **./build.sh --config newconfig.cfg geos2wrf** to test the build.

- GOCART2WRF:
  - Create a new makefile template in directory *utils/gocart2wrf_2* to specify compilers and compiler flags.
  - Edit *newconfig.cfg* to enter the new makefile template name as environmental variables GOCART2WRF_MAKEFILE.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg gocart2wrf** to test the build.

- GSDSU:
  - Create a new makefile template in directory *GSDSU/SRC* specifying the appropriate compilers, compiler flags, and MPI library if applicable.
  - Edit *newconfig.cfg* to enter the new makefile template name as environment variable SDSU_MAKEFILE.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg gsdsu** to test the build.

- LDT:
  - Edit *ldt/arch/Config.pl* to specify the compiler flags. Compilers are specified using the LDT_ARCH environment flag (e.g., 'linux_ifc' indicates Intel compilers on Linux).
  - Edit *newconfig.cfg* to specify the correct LDT_ARCH value.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg ldt** to test the build.

- LISCONFIG:
  - Create a makefile template in directory *utils/lisconfig* to specify compilers and compiler flags.
  - Edit *newconfig.cfg* to enter the new makefile template name as environmental variable LISCONFIG_MAKEFILE
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg lisconfig** to test the build.

- LVT:

- Edit *lvt/arch/Config.pl* to specify the compiler flags. Compilers are specified using the LVT_ARCH environment flag (e.g., 'linux_ifc' indicates Intel compilers on Linux).
- Edit *newconfig.cfg* to specify the correct LVT_ARCH value.
- In the top NU-WRF directory, run **./build.sh --config newconfig.cfg ldt** to test the build.

- MET:
  - Create a new makefile template in directory *MET* specifying the appropriate compiler flags.
  - Edit *newconfig.cfg* to enter the new makefile template name as environmental variable MET_USERS_DEFS_MK.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg met** to test the build.

- PREP_CHEM_SOURCES:
  - Create a new makefile template in directory *utils/prep_chem_sources/bin/build* to specify compilers and compiler flags. Note that the file name must use the naming convention *include.mk.**.
  - Edit *newconfig.cfg* to enter the suffix of the new Makefile template name as environmental variable MAKEPSC_OPT.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg prep_chem_sources** to test the build.

- RIP4:
  - Inspect and edit *RIP4/arch/configure.defaults* to ensure a block exists for the desired operating system, hardware, and compilers. Note that RIP4 is serial only (no MPI support).
  - Run **RIP4/configure** at the command line to identify the integer value of the appropriate build selection.
  - Edit *newconfig.cfg* to enter the configure option as environmental variable RIP_CONFIGURE_OPT.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg rip** to test the build.

- SST2WRF:
  - Create a new makefile template in directory *utils/sst2wrf* to specify compilers and compiler flags.
  - Edit *newconfig.cfg* to enter the makefile template name as environmental variable SST2WRF_MAKEFILE.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg sst2wrf** to test the build.

- UPP:
  - *NOTE: Make sure WRFV3 is ported first.*
  - Inspect and edit *UPP/arch/configure.defaults* to ensure a block exists for the desired operating system, hardware, compilers, and choice of parallelism. It is recommended that two blocks (for serial and MPI) exist.
  - Run **UPP/configure** at the command line to identify the integer value(s) of the appropriate build selections(s).
  - Edit *newconfig.cfg* to enter the configure options as environment variables UPP_CONFIGURE_MPI_OPT and UPP_CONFIGURE_NOMPI_OPT. Also set environmental variable UPP_USE_MPI to toggle MPI on or off.
  - Finally, in the top NU-WRF directory, run **./build.sh --config newconfig.cfg upp** to test the build.
- WPS:
  - *NOTE: Make sure WRFV3 is ported first.*
  - Inspect and edit *WPS/arch/configure.defaults* to ensure a block exists for the desired operating system, hardware, compilers, and choice of parallelism. It is recommended that two blocks (for serial and MPI) exist.
  - Run **WPS/configure** at the command line to identify the integer value(s) of the appropriate build selection(s).
  - Edit *newconfig.cfg* to enter the configure options as environmental variables WPS_CONFIGURE_MPI_OPT and WPS_CONFIGURE_NOMPI_OPT. Also set environmental variable WPS_USE_MPI to toggle MPI on or off.
  - In the top NU-WRF directory, run **./build.sh --config newconfig.cfg wps** to test the build.
- WRFV3 and LIS:
  - Inspect and edit *WRFV3/arch/configure_new.defaults* to ensure a block exists for the desired operating system, hardware, compilers, and choice of parallelism. It is recommended that two blocks (for serial and MPI) exist.
  - Run **WRFV3/configure** to identify the integer value(s) of the appropriate build selection(s).
  - Create a new configure.lis makefile template in *WRFV3/lis/arch* with appropriate compiler selections. It is recommended that two templates (for serial and MPI) exist. NOTE: This approach is used instead of running the LIS *configure* script because LDFLAGS must

be absent from the configure.lis file if the LIS code is compiled for coupling; also, it is easier to to pass consistent debugging compiler flags to WRF and LIS by having the NU-WRF build system do it on the fly.

- Edit the *newconfig.cfg* to enter the configure options as environmental variables WRF_CONFIGURE_MPI_OPT and WRF_CONFIGURE_NOMPI_OPT. Also list the makefile templates with environmental variables WRF_CONFIGURE_LIS_MPI and WRF_CONFIGURE_LIS_NOMPI. Also set environmental variables WRF_USE_MPI to toggle MPI on or off. Also, set environmental variable LIS_ARCH to the value appropriate for your operating system and compiler [see NASA (2015b) for options].
- In the top NU-WRF directory, run **./build.sh --config newconfig.cfg wrf** to test the build. Also test with the chem and kpp targets.

References

Barker, D., X.-Y. Huang, Z. Liu, T. Auligne, X. Zhang, S. Rugg, R. Ajjaji, A. Bourgeois, J. Bray, Y. Chen, M. Demirtas, Y.-R. Guo, T. Henderson, W. Huang, H.-C. Lin, J. Michalakes, S. Risvi, and X. Zhang, 2012: The Weather Research and Forecasting model's community variational/ensemble data assimilation system: WRFDA. *Bulletin of the American Meteorological Society*, **93**, 831–843.

Bleck, R., G. Halliwell, A. Wallcraft, S. Carroll, K. Kelly, and K. Rushing, 2002: HYbrid Coordinate Ocean Model (HYCOM) user's manual. HYCOM consortium, 199pp. [Available online at http://hycom.org/attachments/063_hycom_users_manual.pdf].

Chin, M., R. B. Rood, S.-J. Lin, J.-F. Muller, and A. M. Thompson, 2002: Atmospheric sulfur cycle simulated in the global model GOCART: Model description and global properties. *Journal of Geophysical Research*, **104**, 24671–24687.

Darmenov, A. S. and A. da Silva, 2013: The Quick Fire Emissions Dataset (QFED) - Documentation of versions 2.1, 2.2, and 2.4. NASA Goddard Space Flight Center, Technical Report Series on Global Modeling and Assimilation, Vol 32, 174 pp.

DTC, 2013a: Model Evaluation Tools – Tropical Cyclone version 4.1 (MET-TC) user's guide. Developmental Testbed Center, Boulder, CO. [Available online at www.dtcenter.org/met/users/docs/users_guide/MET-TC_Users_Guide_v4.1.pdf].

— 2013b: Model Evaluation Tools Version 4.1 (METV4.1) user's guide 4.1. Developmental Testbed Center, Boulder, CO. [Available online at www.dtcenter.org/met/users/docs/users_guide/MET_User_Guide_v4.1.pdf].

— 2014: User's guide for the NMM core of the Weather Research and Forecast (WRF) modeling system version 3. Developmental Testbed Center, Boulder, CO. [Available online at www.dtcenter.org/wrf-nmm/users/docs/user_guide/V3/users_guide_nmm_chap1-7.pdf].

ESRL, 2013: WRF/Chem version 3.5 user's guide. NOAA Earth System Research Laboratory, 94pp. [Available online at http://ruc.noaa.gov/wrf/WG11/Users_guide.pdf].

Freitas, S. R., K. M. Longo, M. F. Alonso, M. Pirre, V. Marecal, G. Grell, R. Stockler, R. F. Mello, and M. S. Gacita, 2011: PREP-CHEM-SRC 1.0: a preprocessor of trace gas and aerosol emission fields for regional and global atmospheric chemistry models. *Geoscientific Model Development*, **4**, 419–433.

Ginoux, P., M. Chin, I. Tegen, J. Prospero, B. Holben, O. Dubovik, and S.-J. Lin, 2001: Sources and distributions of dust aerosols simulated with the GOCART model. *Journal of Geophysical Research*, **106**, 20225–20273.

Ginoux, P., D. Garbuzov, and N. C. Hsu, 2010: Identification of anthropogenic and natural dust sources using Moderate Resolution Imaging Spectroradiometer (MODIS) Deep Blue level 2 data. *Journal of Geophysical Research*, **115**, doi:10.1029/2009JD012398.

Ginoux, P., J. M. Prospero, T. E. Gill, N. C. Hsu, and M. Zhao, 2012: Global-scale attribution of anthropogenic and natural dust sources and their emission rates based on MODIS Deep Blue aerosol products. *Reviews of Geophysics*, **50**, DOI: 10.1029/2012RG000388.

Gochis, D., W. Yu, and D. Yates, 2013: The NCAR WRF-Hydro technical description and user's guide. National Center for Atmospheric Research, 120pp. [Available online at http://www.ral.ucar.edu/projects/wrf_hydro/images/WRF_Hydro_Technical_Description_and User_Guide_v1.0.pdf].

Kishcha, P., A. M. da Silva, B. Starobinets, and P. Albert, 2014: Air pollution over the Ganges basin and northwest Bay of Bengal in the early postmonsoon season based on NASA MERRAero data. *Journal of Geophysical Research: Atmospheres*, **119**, 1555–1570, doi:10.1002/2013JD020328.

Koster, R. D., M. J. Suarez, A. Ducharne, M. Stieglitz, and P. Kumar, 2000: A catchment-based approach to modeling land surface processes in a general circulation model: 1. Model structure. *Journal of Geophysical Research: Atmospheres*, **105**, 24809–24822, doi:10.1029/2000JD900327.

Kumar, S. V., C. D. Peters-Lidard, Y. Tian, P. R. Houser, J. Geiger, S. Olden, L. Lighty, J. L. Eastman, B. Doty, P. Dirmeyer, J. Adams, K. Mitchell, E. F. Wood, and J. Sheffield, 2006: An interoperable framework for high resolution land surface modeling. *Environmental Modeling and Software*, **21**, 1402–1415, doi:10.1016/j.envsoft.2005.07.004.

Matsui, T. and J. P. Jacob, 2014: Goddard radiation scheme for NU-WRF version 2014. NASA Goddard Space Flight Center, 4pp.

Matsui, T. and E. Kemp, 2014: Goddard Satellite Data Simulator Unit (G-SDSU) version 3.3.3. NASA Goddard Space Flight Center, 26pp.

Matsui, T., J. Santanello, J. J. Shi, W.-K. Tao, D. Wu, C. Peters-Lidard, E. Kemp, M. Chin, D. Starr, M. Sekiguchi, and F. Aires, 2014: Introducing multi-sensor satellite radiance-based evaluation for regional Earth system modeling. *Journal of Geophysical Research: Atmospheres*, **119**, 8450–8475, doi:10.1002/2013JD021424.

Matsui, T., X. Zeng, W.-K. Tao, H. Masunaga, W. S. Olson, and S. Lang, 2009: Evaluation of long-term cloud-resolving model simulations using satellite radiance observations and multifrequency satellite simulators. *Journal of Atmospheric and Oceanic Technology*, **26**, 1261–1274.

NASA, 2015a: LDT 7.1 users' guide. NASA Goddard Space Flight Center, 124pp. [Available online at http://lis.gsfc.nasa.gov/LDT/LDT_documentation.php].

— 2015b: LIS 7.1 users' guide. NASA Goddard Space Flight Center, 175pp. [Available online at http://lis.gsfc.nasa.gov/LIS_documentation.php].

— 2015c: LVT 7.1 users' guide. NASA Goddard Space Flight Center, 98pp. [Available online at http://lis.gsfc.nasa.gov/LVT/LVT_documentation.php].

NCAR, 2014: User's guide for the Advanced Research WRF (ARW) modeling system version 3.5. National Center for Atmospheric Research, 413pp. [Available online at http://www.mmm.ucar.edu/wrf/users/docs/user_guide_V3.5/ARWUsersGuideV3.pdf].

Peters-Lidard, C. D., P. R. Houser, Y. Tian, S. V. Kumar, J. Geiger, S. Olden, L. Lighty, B. Doty, P. Dirmeyer, J. Adams, K. Mitchell, E. F. Wood, and J. Sheffield, 2007: High-performance Earth system modeling with NASA/GSFC's Land Information System. *Innovations in Systems and Software Engineering*, **3**, 157–165, doi:10.1007/s11334–007–0028–x.

Peters-Lidard, C. D., E. M. Kemp, T. Matsui, J. A. Santanello, Jr., S. V. Kumar, J. P. Jacob, T. Clune, W.-K. Tao, M. Chin, A. Hou, J. L. Case, D. Kim, K.-M. Kim, W. Lau, Y. Liu, J. Shi, D. Starr, Q. Tan, Z. Tao, B. Zaitchik, B. Zavodsky, S. Q. Zhang, and M. Zupanski, 2015: Integrated modeling of aerosol, cloud, precipitation, and land processes at satellite-resolved scales with a NASA Unified Weather Research and Forecasting model. *Environmental Modeling and Software*, **67**, 149–159, doi:10.1016/j.envsoft.2015.01.007.

Randerson, J. T., G. R. van der Werf, L. Giglio, G. J. Collatz, and P. S. Kasibhatla, 2013: Global Fire Emissions Database, version 3 (GFEDv3.1). Data set. Available on-line [http://daac.ornl.gov/] from Oak Ridge National Laboratory Distributed Active Archive Center, Oak Ridge, Tennessee, USA.

Rienecker, M. M., M. J. Suarez, R. Gelaro, R. Todling, J. Bacmeister, E. Liu, M. G. Bosilovich, S. D. Schubert, L. Takacs, G.-K. Kim, S. Bloom, J. Chen, D. Collins, A. Conaty, A. da Silva, W. Gu, J. Joiner, R. D. Koster, R. Lucchesi, A. Molod, T. Owens, S. Pawson, P. Pegion, C. R. Redder, R. Reichle, F. R. Robertson, A. G. Ruddick, M. Sienkiewicz, and J. Woollen, 2011: MERRA – NASA's Modern-Era Retrospective Analysis for Research and Applications. *Journal of Climate*, **24**, 3624–3648.

Rienecker, M. M., M. J. Suarez, R. Todling, J. Bacmeister, L. Takacs, H.-C. Liu, W. Gu, M. Sienkiewicz, R. D. Koster, R. Gelaro, I. Stajner, and J. E. Nielsen, 2008: The GEOS-5 data assimilation system – Documentation of versions 5.0.1, 5.1.0, and 5.2.0. NASA Goddard Space Flight Center, Technical Report Series on Global Modeling and Assimilation, Vol 27, 118 pp. [Available online at http://gmao.gsfc.nasa.gov/pubs/docs/tm27.pdf].

Shi, J. J., T. Matsui, W.-K. Tao, Q. Tan, C. Peters-Lidard, M. Chin, K. Pickering, N. Guy, S. Lang, and E. M. Kemp, 2014: Implementation of an aerosol-cloud-microphysics-radiation coupling into the NASA Unified WRF: Simulation results for the 6-7 August 2006 AMMA special observing period. *Quarterly Journal of the Royal Meteorological Society*, DOI: 10.1002/qj.2286.

Skamarock, W. C., J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. G. Duda, X.-Y. Huang, W. Wang, and J. G. Powers, 2008: A description of the Advanced Research WRF version 3. NCAR Technical Note, NCAR/TN-475+STR, 113pp. [Available online at http://www.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf].

Stoelinga, M. T., 2006: A users' guide to RIP version 4: A program for visualizing mesoscale model output. Web document. [Available online at https://www.mmm.ucar.edu/mm5/documents/ripug_V4.html].

Tallapragada, V., L. Bernardet, S. Gopalakrishnan, Y. Kwon, Q. L. Liu, T. Marchok, M. Tong, R. Tuleya, R. Yablonsky, and X. Zhang, 2013: Hurricane Weather Research and Forecasting (HWRF) model: 2013 scientific documentation. August 2013 – HWRF v3.5a. Developmental Testbed Center Technical Report, 99pp. [Available online at http://www.emc.ncep.noaa.gov/HWRF/HWRFScientificDocumentation2013.pdf].

Tao, Z., J. P. Jacob, and E. Kemp, 2014: User guide: CASA2WRF for NU-WRF. NASA Goddard Space Flight Center, 4pp.

van der Werf, G. R., J. T. Randerson, L. Giglio, G. J. Collatz, M. Mu, P. S. Kasibhatla, D. C. Morton, R. S. DeFries, Y. Jin, and T. T. van Leeuwen, 2010: Global fire emissions and the contribution of deforestation, savanna, forest, agricultural, and peat fires (1997-2009). *Atmospheric Chemistry and Physics*, **10**, 11707–11735.