WRF-Solar® System Description for New York State

3002025151

WRF-Solar® System Description for New York State

3002025151

Technical Update, November 2022

EPRI Project Manager

M. Bello

EPRI
3420 Hillview Avenue, Palo Alto, California 94304-1338 ▪ PO Box 10412, Palo Alto, California 94303-0813 ▪ USA
800.313.3774 ▪ 650.855.2121 ▪ askepri@epri.com ▪ www.epri.com

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

THE FOLLOWING ORGANIZATION, UNDER CONTRACT TO EPRI, PREPARED THIS REPORT:

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH (NCAR)

This is an EPRI Technical Update report. A Technical Update report is intended as an informal report of continuing research, a meeting, or a topical study. It is not a final EPRI technical report.

NOTE

For further information about EPRI, call the EPRI Customer Assistance Center at 800.313.3774 or e-mail askepri@epri.com.

Together…Shaping the Future of Energy®

ACKNOWLEDGMENTS

The following organization, under contract to EPRI, prepared this report:

National Center for Atmospheric Research
P.O. Box 3000
Boulder, CO 80307

Principal Investigator
J. A. Lee

This report describes research sponsored by EPRI. EPRI would like to acknowledge the New York Power Authority (NYPA), the New York State Energy Research and Development Authority (NYSERDA) and the DOE Solar Energy Technologies Office (DOE SETO) for their roles in co-sponsoring this effort.

This publication is a corporate document that should be cited in the literature in the following manner:

WRF-Solar® System Description for New York State. EPRI, Palo Alto, CA: 2022. 3002025151.

ABSTRACT

The WRF-Solar® numerical weather prediction system forms one of the key foundations of the Solar Forecasting for New York (NYSolarCast) system that is currently under development by the National Center for Atmospheric Research (NCAR). This Technical Update describes the elements of the WRF-Solar system as used in NYSolarCast, including configuration files, namelists, and Python scripts to run the entire system. It also provides figures illustrating sample time series and gridded model output, and a full list of all output variables.

Keywords

New York State
Numerical weather prediction
Renewable energy
Solar forecasting
WRF-Solar

Deliverable Number: 3002025151

Product Type: Technical Update

Product Title: WRF-Solar® System Description for New York State

---

PRIMARY AUDIENCE: Providers of renewable energy forecasting for utility and system operator end users

SECONDARY AUDIENCE: Users of renewable energy forecasting in bulk power system operations

KEY RESEARCH QUESTION

What is the best way to configure a quasi-operational numerical weather prediction system to predict global horizontal irradiance (GHI) in high-spatial resolution over New York State (NYS) at nowcast (0–6-hours) and day-ahead lead times. The gridded GHI forecasts developed here can then serve as the basis for subsequent machine learning algorithms that convert GHI forecasts to power forecasts at select utility-scale solar farms and for regionally aggregated distributed solar installations. These other downstream systems will be described in future Technical Updates upon their completion.

RESEARCH OVERVIEW

To facilitate provision of quasi-operational solar power forecasts across NYS, the work described here focused on configuring the Weather Research and Forecasting (WRF) numerical weather prediction model developed for solar applications (WRF-Solar®). This WRF-Solar configuration produces gridded forecasts of GHI and other relevant meteorological parameters at 3-km horizontal grid spacing and 15-minute (min) output across NYS at both nowcasting (0–6-hours) and day-ahead lead times. The nowcasting simulations are initialized hourly from 1100–1900 universal time coordinated (UTC), running “ahead of the clock” so that the entire 6-hour forecast is usable as a true nowcast. The day-ahead simulations are run once daily, starting at 0600 UTC and go out 42 hours. Python scripts are provided to run each step of the WRF-Solar system automatically from a crontab entry.

KEY FINDINGS

- One of the foundational elements of the in-development Solar Forecasting for New York (NYSolarCast) system is 3-km WRF-Solar irradiance and meteorological forecasts, run several times per day, setup in this effort and described herein.
- WRF-Solar and WRF preprocessing system (WPS) are in the public domain and available on GitHub.com.
- Key configuration files and WRF/WPS namelist templates are included and described.
- Two Python scripts, setup_wrf.py and setup_wrf_wrapper.py, which run every step of the WPS/WRF system for NYSolarCast, are publicly available on GitHub and described here.
- Crontab entry examples are included.

WHY THIS MATTERS

Numerical weather modeling provides the basis for all other renewable forecasting used in operations. This report describes the numerical weather prediction model, WRF-Solar, that is one of the foundational components of the NYSolarCast system, how it is configured, and how to run the entire WPS/WRF workflow quasi-operationally using Python scripts. This WRF-Solar system can be run on any HPC environment, and the descriptions provided here should enable users to implement this system quasi-operationally on any HPC environment. This will be used with other forecasting systems and data as part of the overall NYSolarCast system being developed as part of the project.

HOW TO APPLY RESULTS

These scripts are written to run the WRF-Solar system on NCAR's Cheyenne high-performance computing (HPC) system. To run them on a different HPC system, the user will need to download and compile WRF and the WRF Preprocessing System (WPS) and modify the Python and queue submission scripts accordingly. WRF and WPS are available in the public domain via GitHub (https://github.com/wrf-model), and the WRF Users' website (http://www2.mmm.ucar.edu/wrf/users/) includes abundant documentation for installing and running WPS and WRF. This report assumes the user already knows how to install and run the elements of the WPS/WRF system.

LEARNING AND ENGAGEMENT OPPORTUNITIES

- This work is being conducted as part of a New York State Energy Research and Development Authority project, in parallel (and with co-funding from) the U.S. Department of Energy Solar Energy Technology Office (DOE SETO) Systems Integration Program: (https://www.energy.gov/eere/solar/systems-integration-team). Further reports from this project will describe other parts of the forecasting system, culminating in a roadmap for solar forecasting across the state. Such information could also support solar forecasting deployment in other regions by providing useful underlying information and models that could be applied.
- EPRI continues to collaborate with members and other industry stakeholders to improve the development of and integration of renewable forecasting. Similar application of forecasting to other regions is possible and could provide benefit. For those in the New York area, engaging with this effort would allow for use of these forecasts in operations. Please contact the below leads for more details.

EPRI CONTACTS: Aidan Tuohy, Sr. Program Manager, atuohy@epri.com; Mobolaji Bello, Technical Leader, mbello@epri.com

PROGRAM: Bulk System Integration of Renewables and Distributed Energy Resources, P173

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1
INTRODUCTION

Background and Motivation

This report describes the overall flow of how to install and run the WRF-Solar® (WRF-Solar is a registered trademark of the University Corporation for Atmospheric Research) numerical weather prediction (NWP) modeling system that the National Center for Atmospheric Research (NCAR) has developed and configured to serve as the foundation for the Solar Forecasting for New York (NYSolarCast) system. The NYSolarCast is being developed as part of a larger effort, of which this report meets one of the milestones. Other reports will be published addressing other parts of the project.

This description includes both the Weather Research and Forecasting (WRF) model itself, necessary configuration files and namelists, and Python scripts to run the entire system, either from the command line or in an automated manner using crontab entries on a high-performance computing (HPC) system. The scripts described here are configured for operation on NCAR's Cheyenne HPC but can be easily modified to allow for installation and operation on other HPC systems.

There are two sets of simulations for WRF-Solar that are run for NYSolarCast, and which feed into all downstream components of the solar irradiance and solar forecasting system. (Future Technical Reports will describe those elements when they are developed and available.) For both the "nowcast" and "day-ahead" WRF systems, the same domain is used, with 3-km gridded output generated every 15 min in wrfout files (netCDF-4 format). The WRF domain and model terrain for NYSolarCast is shown in Figure 1-1, and a map of the grid point locations over a small portion of the domain can be seen in Figure 1-2.

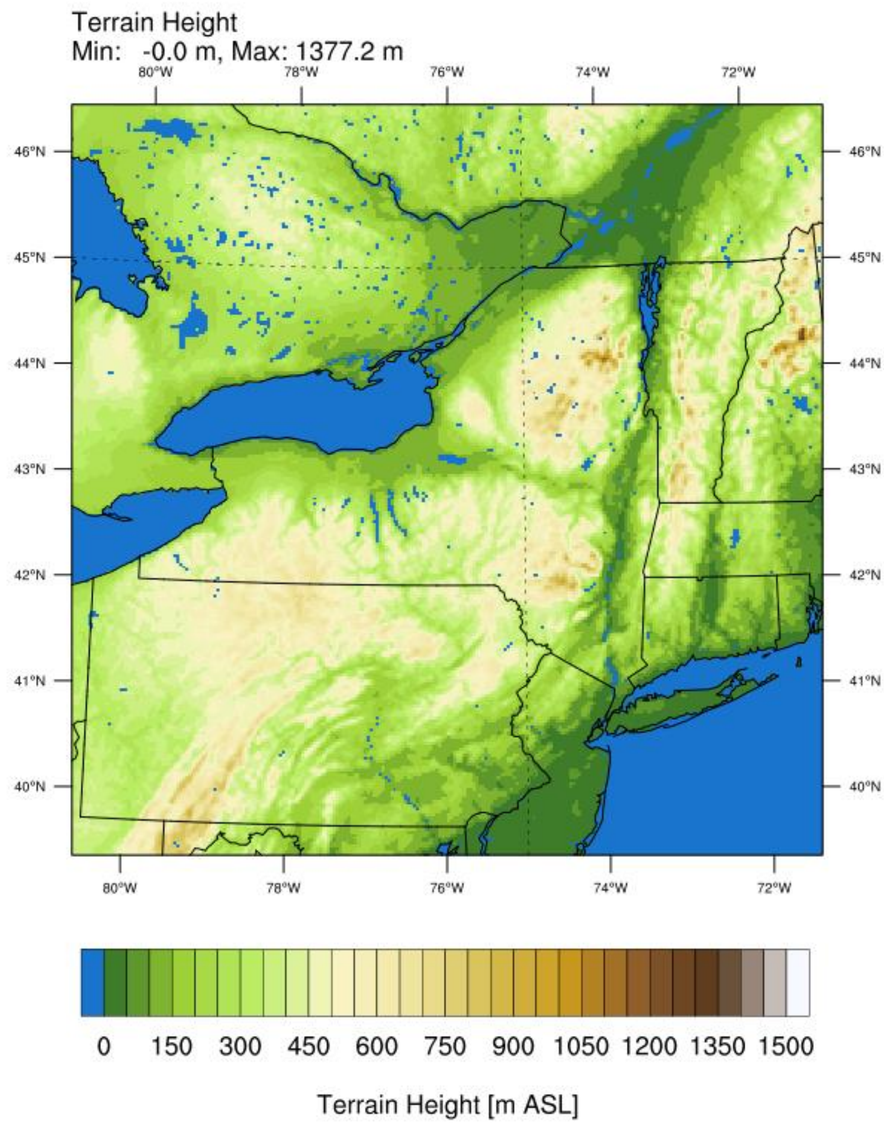

Figure 1-1
Map of the WRF domain and model terrain height for NYSolarCast

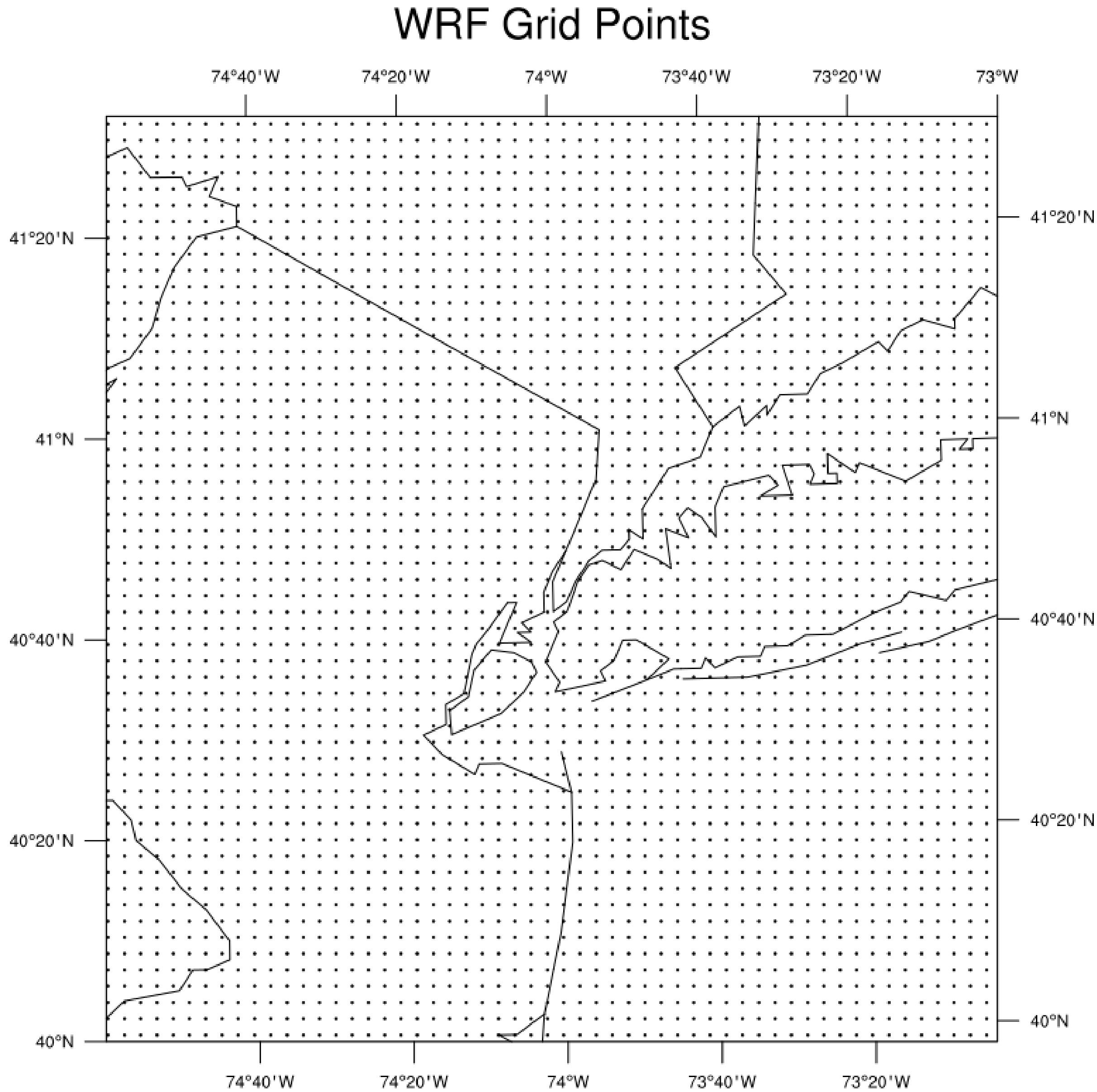

Figure 1-2
Map of the location of WRF grid points over a subset of the domain roughly centered on New York City. The horizontal grid spacing for this WRF domain is 3-km.

- **Nowcast**
  - The 6-hour WRF-Solar simulations initialized hourly with start times from 1100–1900 universal time coordinated (UTC). These forecasts are initialized from 2-hour old high-resolution rapid refresh (HRRR) forecasts. HRRR is an hourly operational 3-km NWP forecast covering the conterminous United States (CONUS), generated by the National Centers for Environmental Prediction (NCEP). We use 2-hour old HRRR forecasts to enable delivery of the WRF-Solar forecast prior to the WRF start time, to preserve its usefulness as a nowcasting tool (for example, the 11 UTC WRF-Solar run is initialized

from the 09 UTC HRRR forecast cycle). The latency of the HRRR model output available for free on Google Cloud is typically up to 1-hour and 25 min. Thus, the 11 UTC WRF-Solar crontab entry can start at 1025 UTC using the 09 UTC HRRR forecast. WRF-Solar can be run sooner if needed, triggering use of 3-hour old HRRR cycles if the 2-hour old HRRR cycle is not fully available yet, without appreciable decrease in forecast accuracy.

- **Day-ahead**
  - The 42-hour WRF-Solar simulations initialized once daily with a start time of 0600 UTC. These forecasts are initialized from the 06 UTC global forecasting system (GFS) model run as soon as it is available. GFS is a global operational forecast generated by NCEP four times per day out to 15 days at an effective grid spacing of 0.25°. The typical latency of GFS operational forecasts appearing on NCEP NOMADS servers is 3 hours and 40 min, so the WRF-Solar day-ahead run can be started from a crontab entry just after 0940 UTC. When HRRR v4 becomes operational at NCEP (this was originally scheduled to happen in June 2020 but was delayed, and a new implementation date has not been scheduled as of this writing), then the 06 UTC HRRR model cycle, which will then go out to 48 hours (the current HRRR v3 06 UTC cycle only goes out to 36 hour), can be used to initialize WRF. At that point, the crontab entry could start the WRF modeling system for the day-ahead run earlier, around 0730 or 0745 UTC, depending on the exact latency.

A schematic illustrating the schedule of these nowcast, and day-ahead WRF-Solar simulations is displayed in Figure 1-3. An example set of time series of global horizontal irradiance (GHI) forecasts from WRF-Solar at a New York State Mesonet (NYSM) [1] site is shown in Figure 1-4, and a sample gridded GHI forecast is displayed in Figure 1-5.

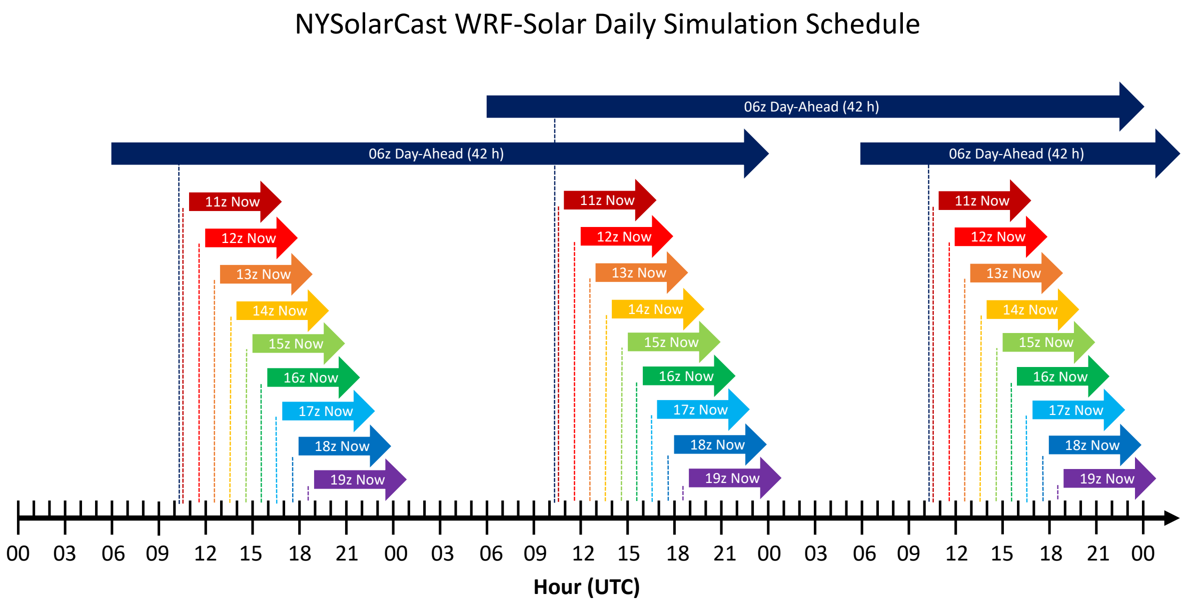

Figure 1-3
Schematic showing the time periods covered by each simulation (thick arrows) in the day-ahead and nowcast WRF-Solar simulations, and their typical quasi-operational availability times (dashed lines) on NCAR's Cheyenne HPC at the time of the writing of this technical report. Note that adjustments can be made to the crontab scheduling to adjust these availability times if needed, and that the day-ahead simulations can be available approximately 2–3 hours sooner than this diagram when HRRR v4 becomes operational at NCEP.

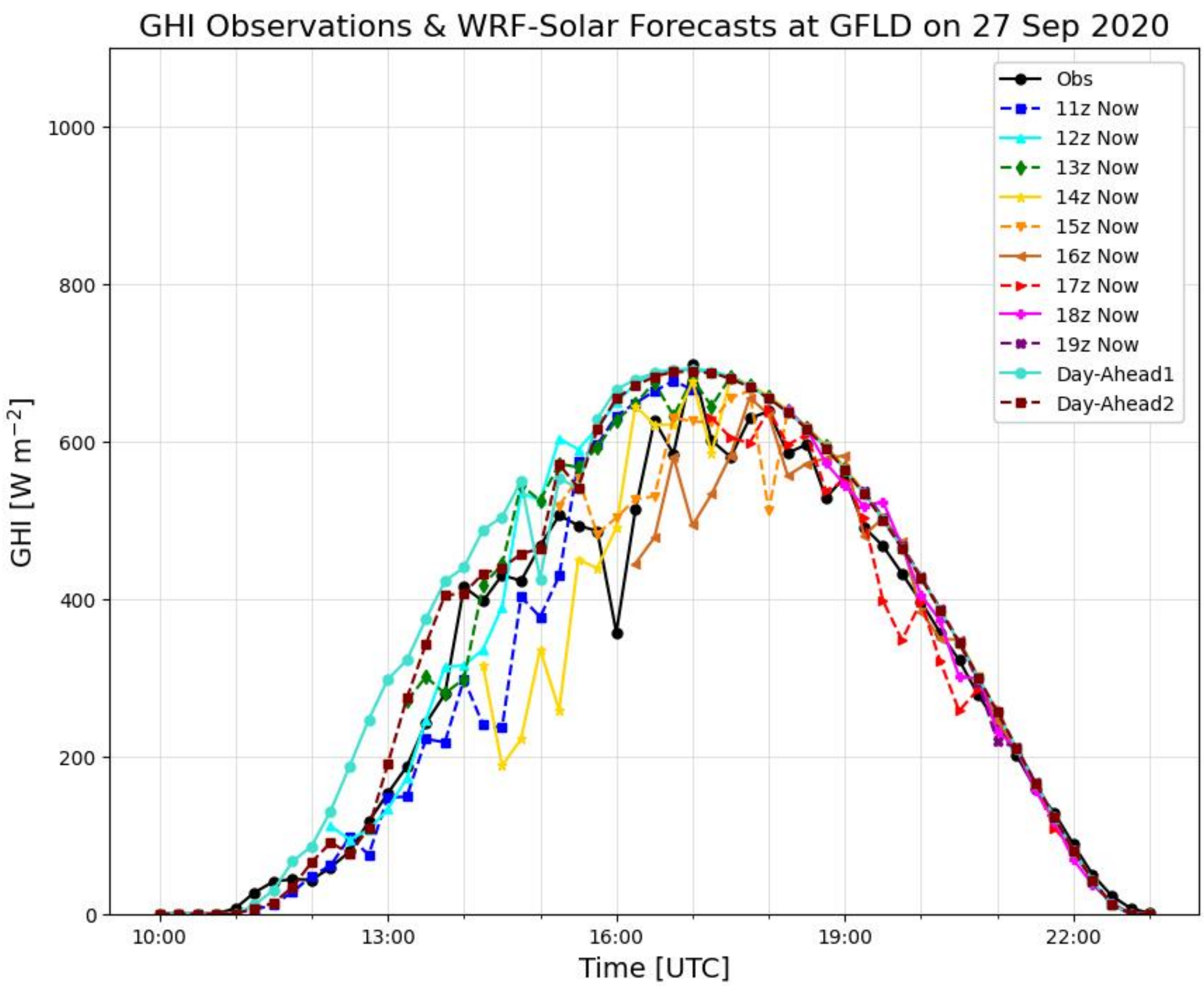

Figure 1-4
Uncorrected WRF-Solar GHI forecasts from each cycle valid over September 27, 2020, at the GFLD (Glenfield, NY) New York State Mesonet site. As such, 11z Now through 19z Now are for the nowcast cycles issued from 11 UTC to 19 UTC. Day-ahead1 is for the day-ahead cycle initialized at 06 UTC on September 26, 2020, while day-ahead2 is for the day-ahead cycle initialized at 06 UTC on September 27, 2020.

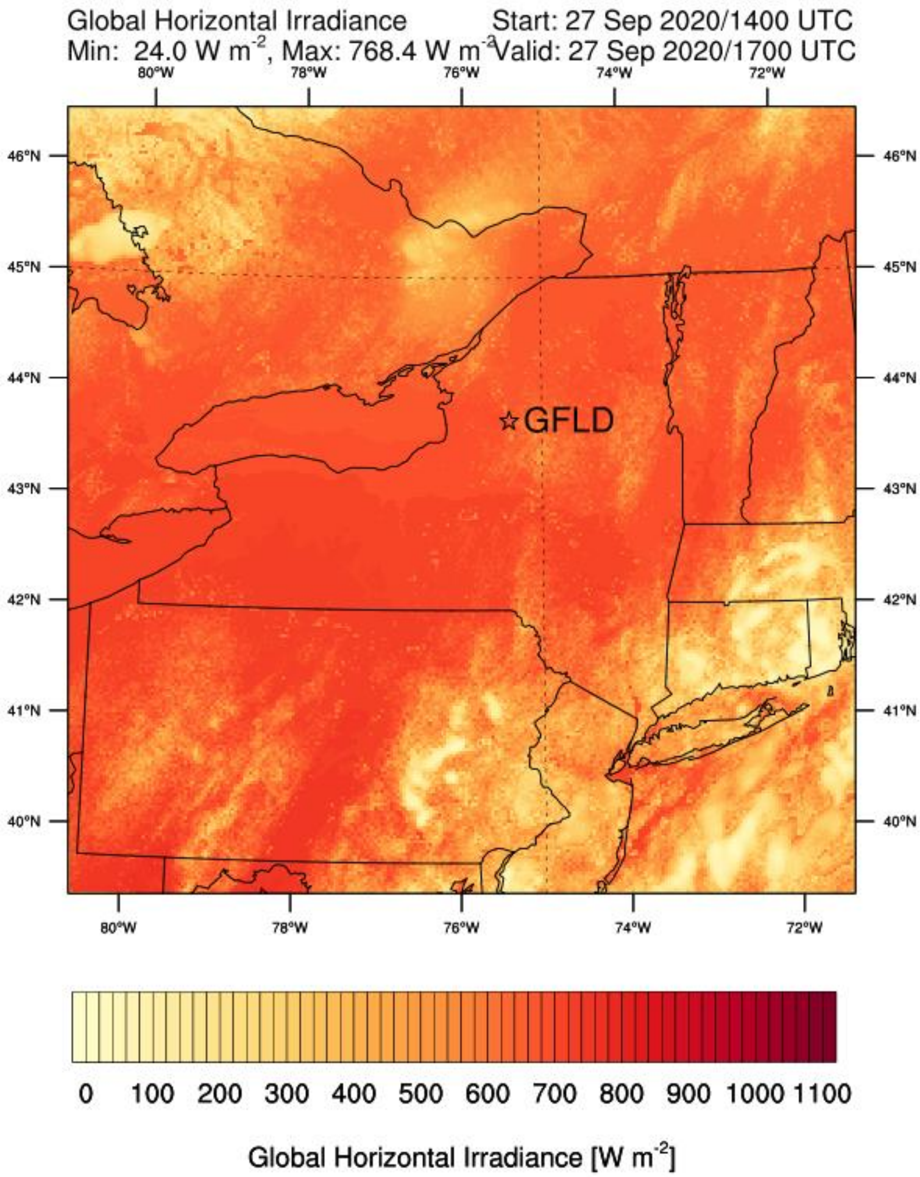

Figure 1-5
WRF-Solar GHI forecast from the 1400 UTC nowcast cycle on September 27, 2020, valid at 1700 UTC, with the GFLD NYSM station labeled (see the yellow line in Figure 1-4)

2
WRF MODEL

The WRF model [2, 3] is a numerical weather prediction model in the public domain that was developed chiefly by NCAR scientists in collaboration with scientists at several other institutions, and is maintained as a community model by NCAR. The WRF-Solar model [4, 5] is a version of WRF that has been customized and configured for solar forecasting applications, and which has been used in other renewable energy forecasting systems developed by NCAR [6–8]. As of WRF v4.2, the WRF-Solar code improvements have been merged into the public WRF release, so maintaining a separate code base or branch is no longer necessary. One of these additions is a simple namelist option to write a suite of 2D solar diagnostics to the history output stream (that is, wrfout files).

For WRF-Solar, we use WRF release-v4.2, which is in the public domain and available for download from GitHub, at https://github.com/wrf-model/WRF. At the time of installing WRF on another system to run NYSolarCast, we recommend that the most recent public release at that time be downloaded and installed (at the time of this report, the current version is release-v4.2.1), so that any bug fixes or other model improvements found and incorporated by WRF developers or community members are automatically included.

The WRF code must be compiled for real-data cases (as opposed to ideal-data cases). Instructions for the compilation and use of WRF can be found in the WRF Users' Guide, which is available at http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/contents.html. Expect successful compilation of WRF to take approximately one hour. Compiling with Intel® or PGi compilers build WRF executables for Real (real.exe) and WRF (wrf.exe) that run faster than gnu compiler builds. On NCAR's Cheyenne HPC, we use the intel compiler to build WRF.

The configuration of the WRF-Solar model used for NYSolarCast is mostly a standard configuration of WRF-Solar that generally performs well in the CONUS. There were no additional modifications made to the WRF/WRF-Solar code base to customize it specifically for the climate conditions of New York; the initial and lateral boundary conditions are sufficient to allow for accurate simulations of the weather in New York State (NYS). Meteorological and irradiance observations from the New York State Mesonet will be used to correct the WRF-Solar gridded forecasts further downstream in the NYSolarCast system. This process will be discussed in a subsequent technical report.

One customization that was made was to use an improved source of aerosols, which can impact solar irradiance at the surface. Aerosols in the atmosphere, including sea salt, dust, sulfates, black carbon, and organic carbon, all of which can come from both natural and anthropogenic sources, scatter and absorb radiation—this is the aerosol direct effect. Thus, higher-aerosol loading will generally decrease irradiance at the surface. This impact is larger on direct normal irradiance (DNI) than it is for GHI, as GHI includes a contribution from diffuse irradiance. DNI is converted into power by concentrated solar power (CSP) plants, while GHI is converted into power by photovoltaic (PV) solar panels and plants. NYSolarCast currently is designed to forecast GHI and conversion to power for select PV plants around New York.

The source [9] of default aerosol climatology in WRF and WRF-Solar. Other strategies to incorporate other aerosol sources include imposing the aerosol optical depth (AOD) from various operational forecasts produced two times or four times daily by National Aeronautics and Space Administration's (NASA's) GEOS-5 model [10–12], National Oceanographic and Atmospheric Administration's (NOAA's) NGAC model [13], or the Copernicus Atmosphere Modeling Service (CAMS) Near-Real-Time model [14] in Europe onto the WRF-Solar forecasts. Such an approach adds additional complexity and with the risk of additional points of failure for an operational nowcasting system, and so might or might not be justifiable for the potential benefit.

Another approach is to use a different aerosol climatology, such as that used by the Thompson-Eidhammer aerosol-aware microphysics scheme [15] in WRF. The Thompson-Eidhammer microphysics scheme's climatology is a monthly climatology of water-friendly and ice-friendly aerosols derived from a history of GEOS-5 analyses. These water-friendly and ice-friendly aerosols then serve as cloud condensation nuclei and ice condensation nuclei, and thus impact cloud formation, which in turn impacts surface irradiance. This is the aerosol indirect effect. Ongoing research on a separate project to compare WRF-Solar performance in clear-sky conditions with different aerosol representations across CONUS (unpublished research by Jared A. Lee of NCAR; manuscript in preparation) found that mean absolute error of clear-sky GHI at the Pennsylvania State University (PSU) Surface Radiation Network (SURFRAD) station [16, 17], which is the closest observing site to NYS in this particular study, was lowest with the Thompson-Eidhammer aerosol climatology (Figure 2-1). Thus, for this system that forecasts GHI, the Thompson-Eidhammer microphysics scheme and aerosol climatology is at least as good in clear-sky conditions as more complex systems that impose aerosol optical depth from the GEOS-5 or CAMS models in the northeastern United States.

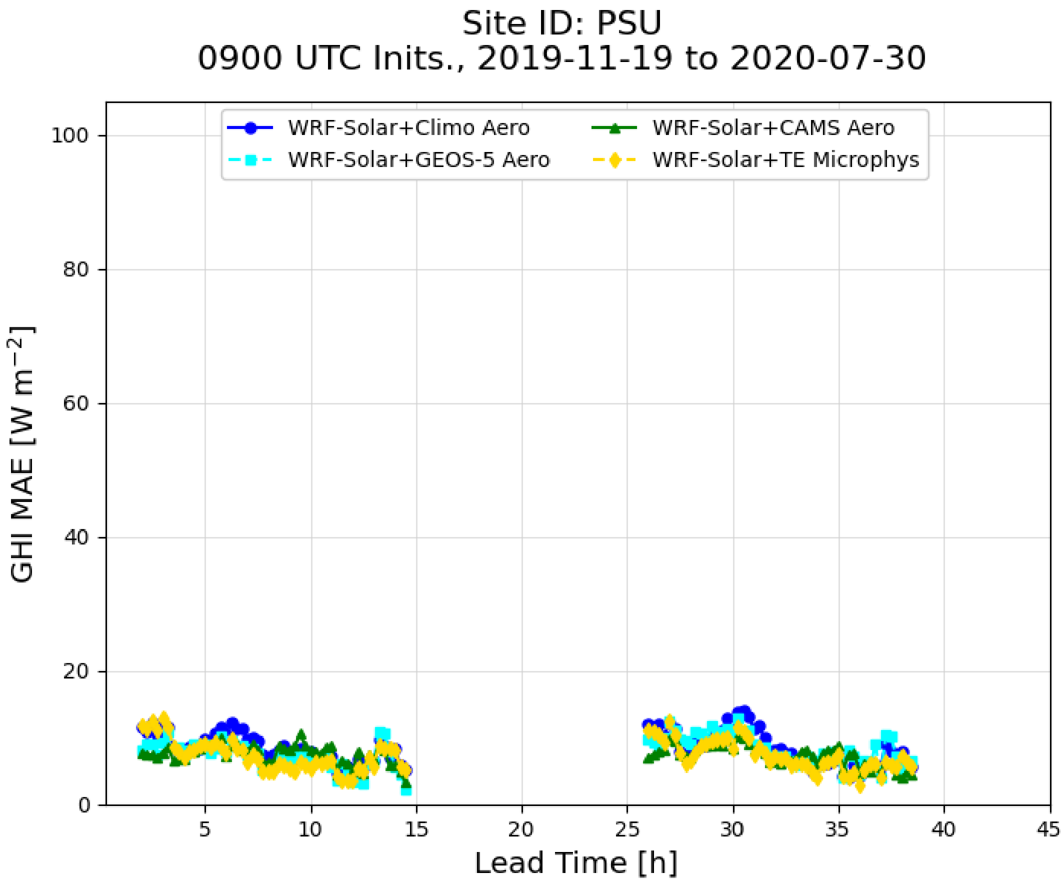

Figure 2-1
Clear-sky GHI mean absolute error at the PSU SURFRAD station as a function of lead time for 9-km WRF-Solar forecasts with different aerosol representations (blue: default climatology; cyan: GEOS-5 aerosols; green: CAMS aerosols; yellow: Thompson-Eidhammer aerosol climatology), for 45-hour forecasts issued at 0900 UTC daily from November 19, 2019 to July 30, 2020. Observations are filtered to include only times when clear sky conditions were observed at the PSU SURFRAD site near Pennsylvania State University.

WRF-Solar in the NYSolarCast system uses the Thompson-Eidhammer microphysics scheme and aerosol climatology. The namelist settings required to use it correctly are described in Section 4 of this report.

A summary of the key output variables in the wrfout files is included below in Table 2-1. Many of these come from turning on the solar diagnostics option in the WRF namelist, discussed in Section 4.

Table 2-1
Table listing the variables in the wrfout files in the system as delivered and as specified by the iofields_d01.txt file described in Section 4

Variable	Description	Units	Type
Times	Valid time	UTC	Character
XLAT	Latitude	°N	Float
XLONG	Longitude	°E	Float
T2	2-min Temperature	K	Float
Q2	2-min Vapor mixing ratio	kg kg^{-1}	Float
PSFC	Surface (station) pressure	Pa	Float
U10	Zonal (U) wind component at 10 min	m s^{-1}	Float
V10	Meridional (V) wind component at 10 min	m s^{-1}	Float
AREA2D	Horizontal grid cell area, using dx, dy, and map factors	m^2	Float
DX2D	Horizontal grid distance: sqrt (area2d)	m	Float
XTIME	Min since 2018-07-15 15:00:00	UTC	Float
KTOP_PLUME	k-level of highest penetrating plume	–	Integer
RAINC	Accumulated total cumulus precipitation	mm	Float
RAINSH	Accumulated total shallow cumulus precipitation	mm	Float
RAINNC	Accumulated total grid scale precipitation	mm	Float
SWDOWN	Global horizontal irradiance	W m^{-2}	Float
SWDDNI	Direct normal irradiance	W m^{-2}	Float
SWDDIF	Diffuse irradiance	W m^{-2}	Float
TAOD5502D	2D Aerosol optical depth at 550 nm from mp_physics=28	–	Float
CLDFRAC2D	2D Max cloud fraction	%	Float
WVP	Water vapor path	kg m^{-2}	Float
LWP	Liquid cloud water path	kg m^{-2}	Float
IWP	Ice cloud water path	kg m^{-2}	Float
SWP	Snow cloud water path	kg m^{-2}	Float
WP_SUM	Sum of LWP+IWP+SWP	kg m^{-2}	Float
LWP_TOT	Liquid cloud water path (resolved+unresolved)	kg m^{-2}	Float
IWP_TOT	Ice cloud water path (resolved+unresolved)	kg m^{-2}	Float
WP_TOT_SUM	Sum of LWP+IWP+SWP (resolved+unresolved)	kg m^{-2}	Float
RE_QC	Mass-weighted liquid cloud effective radius	m	Float
RE_QI	Mass-weighted ice cloud effective radius	m	Float
RE_QS	Mass-weighted snow cloud effective radius	m	Float

Table 2-1 (continued)
Table listing the variables in the wrfout files in the system as delivered and as specified by the iofields_d01.txt file described in Section 4

Variable	Description	Units	Type
RE_QC_TOT	Mass-weighted liquid cloud effective radius (resolved+unresolved)	m	Float
RE_QI_TOT	Mass-weighted ice cloud effective radius (resolved+unresolved)	m	Float
TAU_QC	Mass-weighted liquid cloud optical thickness	–	Float
TAU_QI	Mass-weighted ice cloud optical thickness	–	Float
TAU_QS	Mass-weighted snow cloud optical thickness	–	Float
TAU_QC_TOT	Mass-weighted liquid cloud optical thickness (resolved+unresolved)	–	Float
TAU_QI_TOT	Mass-weighted ice cloud optical thickness (resolved+unresolved	–	Float
CBASEHT	Cloud base height	m	Float
CTOPHT	Cloud top height	m	Float
CBASEHT_TOT	Cloud base height (resolved+unresolved)	m	Float
CTOPHT_TOT	Cloud top height (resolved+unresolved)	m	Float
CLRNIDX	Clearness index	–	Float
SZA	Solar zenith angle	°	Float

Also note that there is a known minor bug in the calculation of solar zenith angle (SZA) in WRF v4.2. Our research team discovered this bug while exploring the utility of the SZA calculated by WRF-Solar and alerted the WRF-Solar developers about it. The bug causes calculations to be incorrect by a small amount and differ from calculations given by Python packages such as *pysolar*, but there is not yet an approved bug "fix" at the time of this report. However, this SZA bug does not impact any operations in the NYSolarCast system under development as it is not used in any downstream applications or calculations.

3
WPS (WRF PRE-PROCESSING SYSTEM)

Also required to run the WRF system is WPS, the WRF preprocessing system. WPS can be compiled after WRF has been successfully compiled. Instructions to compile and use WPS are also found in the WRF Users' Guide linked at the end of the previous section. We use release-v4.2 of WPS, which is also in the public domain on GitHub, at https://github.com/wrf-model/WPS. We made no modifications to WPS release-v4.2.

Successful compilation of WPS normally takes about 10 min and will generate executables for three programs: Geogrid (geogrid.exe), Ungrib (ungrib.exe), and Metgrid (metgrid.exe). Geogrid defines the horizontal grid used by WPS/WRF, referencing static terrain and land use/vegetation databases. When the user is satisfied with the geographic domain files, Geogrid does not need to be run subsequently. Ungrib decodes grib/grib2-formatted model data (for example, from HRRR or GFS) that is used as the initial condition (IC) and lateral boundary condition (LBC) data for the WRF simulation, and outputs those data in an intermediate format. Metgrid converts the Ungrib atmospheric and soil variable output and interpolates it onto the Geogrid-defined horizontal grid in netCDF files. The output from Metgrid is then fed into Real, which interpolates the atmospheric data to the vertical levels defined in the WRF namelist.

WPS can be compiled either for serial or shared/distributed memory execution. In this system as designed by NCAR, WPS was compiled serially. The same compiler that was used to compile WRF must also be used to compile WPS.

4
WRF/WPS CONFIGURATION FILES AND NAMELISTS

In the config folder of the files delivered to EPRI, there are a few files that will be described briefly. The Python scripts described in a later section assume these files are located together in a folder called config.

- geo_em.d01.nc
  - This is the output of the Geogrid program from WPS for the 3-km New York State domain used for this project. It only takes a few minutes to generate this file with the Geogrid program, but after it has been generated, there is no need to spend time re-running geogrid for subsequent model runs.
- iofields_d01.txt
  - Text file whose location is specified in the WRF namelist (namelist.input) that instructs WRF which variables to include or exclude from the history output stream (hereafter, simply referred to as wrfout files). This file allows for run-time inclusion/exclusion of variables in the wrfout files without having to modify the Registry, which requires re-compiling the WRF code.
- pbs_submit_real.bash
  - PBS bash submission script for the real.exe program on NCAR's Cheyenne HPC.
- pbs_submit_wrf_dayahead.bash
  - PBS bash submission script for the wrf.exe program on NCAR's Cheyenne HPC for the day-ahead runs.
- pbs_submit_wrf_nowcast.bash
  - PBS bash submission script for the wrf.exe program on NCAR's Cheyenne HPC for the nowcast runs.
- QNWFA_QNIFA_SIGMA_MONTHLY.dat
  - File with monthly water-friendly aerosol and ice-friendly aerosol climatologies, for use with the Thompson-Eidhammer aerosol-aware microphysics scheme in WRF (mp_physics=28). This file is accessed in the Metgrid step of WPS, and its location is specified in the namelist.wps file.
- wps_domain.png
  - Map of the Geogrid domain (geo_em.d01.nc) generated by the plotgrids.ncl utility from WPS (see Figure 1-1).

WRF/WPS Namelist Templates

In the config/nml_templates folder, there are several templates for Fortran namelists used by both WPS and WRF, configured for the NYSolarCast system. The files are listed below, followed by a brief highlight of some of the key options used in the namelists. Note that these namelists point to file paths on NCAR's Cheyenne HPC and will need to be updated if run on another system or by another user.

- namelist.input
  - A copy of namelist.input.nowcast (described below). Both real.exe and wrf.exe require that the namelist be named namelist.input.
- namelist.input.dayahead.newgfs
  - The WRF namelist template for the day-ahead runs, when using GFS forecasts issued from the 12 Jun 2019/12 UTC cycle and forward (num_metgrid_levels=34). This uses GFS forecast output files every three hours (interval_seconds=10800). Note that if HRRR v4 is eventually used instead of GFS to drive the day-ahead WRF-Solar simulation, set num_metgrid_levels=51, num_metgrid_soil_levels=9, and interval_seconds=3600.
- namelist.input.dayahead.oldgfs
  - The WRF namelist template for the day-ahead runs, when using GFS forecasts issued from before the 12 Jun 2019/12 UTC cycle (num_metgrid_levels=32). This uses GFS forecast output files every three hours (interval_seconds=10800).
- namelist.input.nowcast
  - The WRF namelist template for the nowcast runs, when using HRRR forecasts, with output files every one hour (interval_seconds=3600).
- namelist.wps.gfs
  - The WPS namelist template for the day-ahead WRF runs initialized from GFS model forecasts (interval_seconds=10800). Note that if HRRR v4 is eventually used instead of GFS to drive the day-ahead WRF-Solar simulation, then use the two namelist.wps files listed below for use with HRRR.
- namelist.wps.hrrr.hybr
  - The WPS namelist template for the nowcast WRF runs initialized from HRRR model forecasts (interval_seconds=3600). This namelist is to be used with the HRRR native hybrid-level output, which is used for the atmospheric variables, as it has higher-vertical resolution than the pressure-level files, as well as some additional variables that are used for WRF initialization.
- namelist.wps.hrrr.soil
  - The WPS namelist template for the nowcast WRF runs initialized from HRRR model forecasts (interval_seconds=3600). This namelist is to be used with the HRRR pressure-level output, which is used for the soil variables, which are unfortunately not included in the hybrid-level output files.

Special Notes about the WPS Namelist (namelist.wps)

Consult the WRF Users' Guide for a full definition of each namelist option.

If running geogrid.exe, geog_data_path will need to point to the location of the static terrain and land use/vegetation input data. This is a separate, large download. See the README at https://github.com/wrf-model/WPS for details.

Also, interval_seconds governs how frequently IC/LBC files will be decoded by Ungrib, and available for use in Metgrid. While the NCEP NOMADS server stores GFS output every hour (interval_seconds=3600), only the most recent 10 days of forecast cycles are stored on NCEP NOMADS. We use interval_seconds=10800 because the historical archive of GFS forecasts on NCAR's Research Data Archive (https://rda.ucar.edu) only stored GFS forecast output every three hours.

Special Notes about the WRF Namelist (namelist.input)

Consult the WRF Users' Guide and run/README.namelist in the WRF installation directory for a full definition of each namelist option. Descriptions of a few select namelist options are given below.

&time_control

iofields_filename

- Points to the iofields_d01.txt file in the config folder mentioned above.

&domains

eta_levels

- These are the same 51 eta levels used in the operational HRRR model.

wif_input_opt = 1

num_wif_levels = 30

- These two variables are required for the Thompson-Eidhammer aerosol-aware microphysics scheme to work properly with the QNWFA_QNIFA_SIGMA_MONTHLY.dat monthly climatology file.

&physics

mp_physics = 28

- This selects the Thompson-Eidhammer aerosol-aware microphysics scheme.

ra_lw_physics = 4

ra_sw_physics = 4

- These options select the RRTMG longwave and shortwave radiation schemes.

radt = 10

- Calls the radiation scheme every 10 min.

swint_opt = 1

- Interpolates shortwave radiation based on updated solar zenith angle between radiation time steps.

sf_sfclay_physics = 5

bl_pbl_physics = 5

- Uses the MYNN surface layer and boundary layer schemes.

sf_surface_physics = 2

- Uses the NOAH land surface scheme.

cu_physics = 0

- Cumulus scheme turned off (explicit convection like this is recommended for grid spacing of 4 km or less).

aer_opt = 3

- Uses the water/ice-friendly aerosol climatology for radiation feedbacks (in conjunction with mp_physics=28).

&diags

solar_diagnostics = 1

- Writes out a suite of solar diagnostics to the wrfout files, including variables like cloud base height, liquid water path, and many more. See Registry/registry.solar_fields in the WRF installation for more details.

5
SETUP_WRF PYTHON SCRIPTS

In addition to the WRF and WPS code bases, the other crucial part of the WRF-Solar software package in the NYSolarCast system are two Python scripts that control all aspects of the system: setup_wrf.py and setup_wrf_wrapper.py. These scripts are described below.

The primary script, setup_wrf.py, is a Python script designed to set up and run all the steps of WPS and WRF on NCAR's Cheyenne HPC. The user should modify paths as needed to various executables, input data directories, output data directories, and so on, and adjust Boolean flags that control whether certain WPS/WRF steps are run or not. This script takes one required and several optional command-line arguments, as described more completely below.

A second script, setup_wrf_wrapper.py, is an optional Python script that is intended to be called by a crontab entry for quasi-operational execution of setup_wrf.py. Details of usage and inputs/outputs are found in the README included with the GitHub repository https://github.com/NCAR/setup_wrf_epri, but are also reproduced here for completeness. The scripts can be downloaded directly from that GitHub repository.

setup_wrf.py

This script links to (and downloads, if necessary) all the files needed to run WPS/WRF. Each program in the WPS/WRF workflow can be optionally executed if its option is set to True. WRF output files can also be optionally moved to an archival directory (arc_dir).

Usage:

```
setup_wrf.py [-h] [-l INIT_DT_LAST] [-I INIT_STRIDE_H] [-s SIM_LENGTH_H] [-f
ICBC_FC_DT] [-r] [-a] init_dt_first
```

Positional arguments:

```
init_dt_first
```

Beginning date/time of first WRF simulation [YYYYMMDD_HH]

Optional arguments:

```
-h, --help
```

Show this help message and exit

```
-l INIT_DT_LAST, --init_dt_last INIT_DT_LAST
```

Beginning date/time of last WRF simulation [YYYYMMDD_HH] (default: same as init_dt_first)

```
-i INIT_STRIDE_H, --init_stride_h INIT_STRIDE_H
```

Integer number of hours between forecast cycles (default: 24)

`-s SIM_LENGTH_H, --sim_length_h SIM_LENGTH_H`

Integer number of hours for WRF simulation (default: 48)

`-f ICBC_FC_DT, --icbc_fc_dt ICBC_FC_DT`

Integer number of hours prior to WRF initialization time for IC/LBC model cycle (default: 0)

`-r, --realtime`

Flag when running in real-time to keep this script running until WRF is done (so that the script can then be run again in archival mode)

`-a, --archive`

Flag to archive wrfout, wrfinput, wrfbdy, and namelist files off of scratch to an archive directory (arc_dir)

setup_wrf_wrapper.py

This Python script is intended to be called by a crontab entry with three arguments: WRF start hour, WRF simulation length, and IC/LBC delta hours. This script gets the current UTC date to pair with the specified WRF start hour to provide the necessary arguments for calling setup_wrf.py.

Usage:

`setup_wrf_wrapper_py [-h] [-s SIM_LENGTH_H] [-f ICBC_FC_DT] init_hr`

Positional arguments:

`init_hr`

Two-digit hour of WRF start time (for example, 06)

Optional arguments:

`-h, --help`

Show this help message and exit

`-s SIM_LENGTH_H, --sim_length_h SIM_LENGTH_H`

Integer number of hours for WRF simulation (default: 6)

`-f ICBC_FC_DT, --icbc_fc_dt ICBC_FC_DT`

Integer number of hours prior to WRF initialization time for IC/LBC model cycle (default: 0)

6
CRONTAB EXAMPLE

The file crontab.txt contains an example set of crontab statements to start the day-ahead WRF run and each of the nowcast WRF runs, by calling setup_wrf_wrapper.py, which then calls setup_wrf.py. Note that the PATH and LD_LIBRARY_PATH variables at the top of the crontab must include entries from the user's environment pertaining to the same compiler and libraries that were used to build WRF/WPS. Failure to do that will cause any of the WPS/WRF executables to fail when called by setup_wrf.py because of shared libraries that cannot be opened or do not exist.

Note that on Cheyenne, the system time is in Mountain Local Time, meaning that the crontab scheduling must be manually updated at the switches between Mountain Standard Time (UTC-7) and Mountain Daylight Time (UTC-6).

The full crontab.txt example is reproduced here for completeness.

```
# crontab -e

SHELL=/bin/bash

HOME=/glade/u/home/jaredlee

MAILTO='jaredlee@ucar.edu'

## To get WPS/WRF executables to work, I had to copy all compiler/netcdf-relevant entries from my environment's PATH and LD_LIBRARY_PATH variables

## NOTE: I have not yet done testing to isolate if only a subset of these are needed.

PATH=/glade/u/apps/ch/opt/proj/6.2.1/intel/19.0.5/bin:/glade/work/jaredlee/python/my_npl_clone_20200417/bin:/glade/u/apps/ch/opt/netcdf/4.7.3/intel/19.0.5/bin:/glade/u/apps/ch/opt/ncarcompilers/0.5.0/intel/19.0.5/mpi:/glade/u/apps/ch/opt/mpt/2.22/bin:/glade/u/apps/ch/opt/ncarcompilers/0.5.0/intel/19.0.5:/glade/u/apps/opt/intel/2019u5/vtune_amplifier/bin64:/glade/u/apps/opt/intel/2019u5/inspector/bin64:/glade/u/apps/opt/intel/2019u5/advisor/bin64:/glade/u/apps/opt/intel/2019u5/compilers_and_libraries/linux/bin/intel64:/ncar/opt/slurm/latest/bin:/glade/u/apps/opt/vncmgr:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/glade/u/apps/ch/os/usr/bin:/opt/pbs/bin

LD_LIBRARY_PATH=/glade/apps/opt/intel/2017u1/compilers_and_libraries_2017.1.132/linux/compiler/lib/intel64:/glade/u/apps/opt/intel/2019u5/mkl/lib/intel64:/glade/u/apps/opt/intel/2019u5/tbb/lib/intel64/gcc4.7:/glade/u/apps/ch/opt/mpt_fmods/2.22/intel/19.0.5:/glade/u/apps/ch/opt/mpt/2.22/lib:/glade/u/apps/opt/intel/2019u5/compilers_and_libraries/linux/lib/intel64:/ncar/opt/slurm/latest/lib:/glade/u/apps/ch/os/usr/lib64:/glade/u/apps/ch/os/usr/lib:/glade/u/apps/ch/os/lib64:/glade/u/apps/ch/os/lib:/usr/lib64


# For details see man 4 crontabs
```

```
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR
sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed


## NOTE: All times on Cheyenne are in Mountain Local Time, so UTC-dependent
processes need to be updated between MDT and MST (sigh)
## MDT cron (UTC-6)
## EPRI Phase 3 WRF Day-Ahead run (can probably bring this back from :42 to
:40 safely)
42 3 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 06 -s 42 -f 0 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_06.log 2>&1


## EPRI Phase 3 WRF Nowcast runs (some files sporadically don't come in until
:25, but most are in by :10)
25 4 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 11 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_11.log 2>&1
25 5 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 12 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_12.log 2>&1
25 6 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 13 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_13.log 2>&1
25 7 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 14 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_14.log 2>&1
25 8 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 15 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_15.log 2>&1
25 9 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 16 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_16.log 2>&1
25 10 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 17 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_17.log 2>&1
25 11 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 18 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_18.log 2>&1
25 12 * * * python3 ~/tools/python/EPRI/setup_wrf_wrapper.py 19 -s 6 -f 2 >
/glade/scratch/jaredlee/EPRI/Phase3/cron/cron_19.log 2>&1
```

7
REFERENCES

1. J. A. Brotzge *et al.*, “A Technical Overview of the New York State Mesonet Standard Network,” *J. Atmos. Oceanic Technol.*, vol. 37, no. 10, pp. 1827–1845, Oct. 2020, doi: 10.1175/JTECH-D-19-0220.1.
2. J. G. Powers *et al.*, “The Weather Research and Forecasting Model: Overview, System Efforts, and Future Directions,” *Bull. Amer. Meteor. Soc.*, vol. 98, no. 8, pp. 1717–1737, Jan. 2017, doi: 10.1175/BAMS-D-15-00308.1.
3. W. C. Skamarock *et al.*, “A Description of the Advanced Research WRF Model Version 4,” *NCAR Technical Note*, no. NCAR/TN–556+STR, p. 162 pp., Mar. 2019, doi: 10.5065/1dfh-6p97.
4. P. A. Jiménez *et al.*, “WRF-Solar: Description and Clear-Sky Assessment of an Augmented NWP Model for Solar Power Prediction,” *Bull. Amer. Meteor. Soc.*, vol. 97, no. 7, pp. 1249–1264, Jul. 2016, doi: 10.1175/BAMS-D-14-00279.1.
5. P. A. Jiménez *et al.*, “The Role of Unresolved Clouds on Short-Range Global Horizontal Irradiance Predictability,” *Mon. Wea. Rev.*, vol. 144, no. 9, pp. 3099–3107, Jun. 2016, doi: 10.1175/MWR-D-16-0104.1.
6. S. E. Haupt *et al.*, “Building the Sun4Cast System: Improvements in Solar Power Forecasting,” *Bull. Amer. Meteor. Soc.*, vol. 99, no. 1, pp. 121–136, Jun. 2017, doi: 10.1175/BAMS-D-16-0221.1.
7. J. A. Lee, S. E. Haupt, P. A. Jiménez, M. A. Rogers, S. D. Miller, and T. C. McCandless, “Solar Irradiance Nowcasting Case Studies near Sacramento,” *J. Appl. Meteor. Climatol.*, vol. 56, no. 1, pp. 85–108, Oct. 2016, doi: 10.1175/JAMC-D-16-0183.1.
8. S. E. Haupt *et al.*, “Combining Artificial Intelligence with Physics-based Methods for Probabilistic Renewable Energy Forecasting,” *Energies*, vol. 13, no. 8, Art. no. 8, Jan. 2020, doi: 10.3390/en13081979.
9. I. Tegen, P. Hollrig, M. Chin, I. Fung, D. Jacob, and J. Penner, “Contribution of Different Aerosol Species to the Global Aerosol Extinction Optical Thickness: Estimates from Model Results,” *Journal of Geophysical Research: Atmospheres*, vol. 102, no. D20, pp. 23895–23915, 1997, doi: 10.1029/97JD01864.
10. M. M. Rienecker *et al.*, “The GEOS-5 Data Assimilation System-Documentation of Versions 5.0.1, 5.1.0, and 5.2.0,” NASA, NASA/TM–2008–104606-VOL-27, Dec. 2008. Accessed: Jul. 30, 2020. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=20120011955.
11. A. Molod, L. Takacs, M. Suarez, and J. Bacmeister, “Development of the GEOS-5 Atmospheric General Circulation Model: Evolution from MERRA to MERRA2,” *Geoscientific Model Development*, vol. 8, no. 5, pp. 1339–1356, May 2015, doi: https://doi.org/10.5194/gmd-8-1339-2015.
12. V. Buchard *et al.*, “Using the OMI Aerosol Index and Absorption Aerosol Optical Depth to Evaluate the NASA MERRA Aerosol Reanalysis,” *Atmospheric Chemistry and Physics*, vol. 15, no. 10, pp. 5743–5760, May 2015, doi: https://doi.org/10.5194/acp-15-5743-2015.

13. J. Wang *et al.*, “The Implementation of NEMS GFS Aerosol Component (NGAC) Version 2.0 for Global Multispecies Forecasting at NOAA/NCE –Part 1: Model Descriptions,” *Geoscientific Model Development*, vol. 11, no. 6, pp. 2315–2332, Jun. 2018, doi: https://doi.org/10.5194/gmd-11-2315-2018.
14. Copernicus, “CAMS Near-real-time Global Analysis and Forecast Daily Data,” *ECMWF*, Feb. 19, 2020. https://www.ecmwf.int/en/forecasts/dataset/cams-near-real-time-global-analysis-and-forecast-daily-data (accessed Aug. 03, 2020).
15. G. Thompson and T. Eidhammer, “A Study of Aerosol Impacts on Clouds and Precipitation Development in a Large Winter Cyclone,” *Journal of the Atmospheric Sciences*, vol. 71, no. 10, pp. 3636–3658, Oct. 2014, doi: 10.1175/JAS-D-13-0305.1.
16. J. A. Augustine, J. J. DeLuisi, and C. N. Long, “SURFRAD–A National Surface Radiation Budget Network for Atmospheric Research,” *Bull. Amer. Meteor. Soc.*, vol. 81, no. 10, pp. 2341–2358, Oct. 2000, doi: 10.1175/1520-0477(2000)081<2341:SANSRB>2.3.CO;2.
17. J. A. Augustine, G. B. Hodges, C. R. Cornwall, J. J. Michalsky, and C. I. Medina, “An Update on SURFRAD—The GCOS Surface Radiation Budget Network for the Continental United States,” *J. Atmos. Oceanic Technol.*, vol. 22, no. 10, pp. 1460–1472, Oct. 2005, doi: 10.1175/JTECH1806.1.

About EPRI

Founded in 1972, EPRI is the world's preeminent independent, non-profit energy research and development organization, with offices around the world. EPRI's trusted experts collaborate with more than 450 companies in 45 countries, driving innovation to ensure the public has clean, safe, reliable, affordable, and equitable access to electricity across the globe. Together, we are shaping the future of energy.

3002025151

EPRI
3420 Hillview Avenue, Palo Alto, California 94304-1338 • PO Box 10412, Palo Alto, California 94303-0813 • USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com