NMRF/TR/01/2024

TECHNICAL REPORT

WRF preprocessing toolbox for IMDAA

Vivekananda Hazra, S. Indira Rani, John P. George, and Raghavendra Ashrit

March 2024

National Centre for Medium Range Weather Forecasting
Ministry of Earth Sciences, Government of India
A-50, Sector-62, NOIDA-201309, INDIA

WRF preprocessing toolbox for IMDAA

Vivekananda Hazra, S. Indira Rani, John P. George, and Raghavendra Ashrit

March 2024

National Centre for Medium Range Weather Forecasting
Ministry of Earth Sciences
A-50, Sector 62, NOIDA-201309, India

Ministry of Earth Sciences
National Centre for Medium Range Weather Forecasting
Document Control Data Sheet

1	Name of the Institute	National Centre for Medium Range Weather Forecasting
2	Document Number	NMRF/TR/01/2024
3	Date of publication	March 2024
4	Title of the document	WRF preprocessing toolbox for IMDAA
5	Type of Document	Technical Report
6	No. of pages, Tables & Figures	55, 4 & 5
7	Number of References	12
8	Author (s)	Vivekananda Hazra, S. Indira Rani, John P. George, and Raghavendra Ashrit
9	Originating Unit	NCMRWF
10	Abstract	The Weather Research and Forecasting (WRF) model is an important tool for forecasting weather at regional/local scales in the academic and research communities. The process of WRF model initialization while using the Indian Monsoon Data Assimilation and Analysis (IMDAA) reanalysis dataset as initial and boundary conditions (IC/BC) are addressed in this report. Users might encounter issues while running the conventional WRF Preprocessing System (WPS) when utilizing the operational IMDAA reanalysis. Therefore, the repository mentioned in this document encompasses a comprehensive set of scripts and tools to automate the intricate steps involved in the WPS which has various components (GEOGRID, UNGRIB, and METGRID), ensuring a seamless and error-free experience for researchers. The project encourages monsoon research in an accelerated phase by utilizing the high-resolution IMDAA reanalysis data for IC/BC for the WRF model.
11	Security classification	Non-Secure
12	Distribution	Unrestricted Distribution
13	Key Words	WRF, Preprocessing, GRIB2, IMDAA, WPS

i. List of abbreviations

AFWA	Air Force Weather Agency
AGRMET	Agricultural Meteorology
ARW	Advanced Research WRF
BUFR	Binary Universal Form for the Representation
CPU	Central Processing Unit
ECMWF	European Centre for Medium Range Weather Forecasting
ERA5	ECMWF Fifth Generation Re-analysis
FF	Fields file format
GFS	Global Forecast System
GRIB2	WMO General Regularly-distributed Information in Binary (Reading and writing)
H	Geopotential height
HDF	Hierarcial data format
HPC	High performance computer
HYCOM	HYbrid Coordinate Ocean Model
HYSPLIT	Hybrid Single Particle Lagrangian Integrated Trajectory
IMD	India Meteorological Department
IMDAA	Indian Monsoon Data Assimilation and Analysis
I/O	Input/Output
JTWC	Joint Typhoon Warning Center
MM5	Mesoscale Model fifth generation
MPI	Message Passing Interface
NAM	North American Mesoscale Forecast System
NCAR	National Centre for Atmospheric Research
NCEP	National Centre for Environmental Prediction
NCL	NCAR Command Language
NCMRWF	National Centre for Medium Range Weather Forecasting
NCUM	NCMRWF Unified Model
NCUM-G	NCUM global model
NCUM-R	NCUM regional model
NETCDF	Network Common Data Form
NEPS-G	NCUM ensemble prediction system - global
NWP	Numerical Weather Prediction

OSF	Ocean State Forecast
PP	Post-processed file
RIP4	Read Interpolate Plot 4th Generation
SuCS	Super Cyclonic Storm
TIGGE	THORPEX International Grand Global Ensemble
TOGA	Tropical Ocean Global Atmosphere
U	Zonal wind
UKMET	United Kingdom Meteorological Office
UM	Unified Model
V	Meridional wind
Vtable	Variable table
WMO	World Meteorological Organization
WMO	World Meteorological Organization
WPS	WRF Pre-Processing System
WRF	Weather Research and Forecasting

ii. Table of Contents

सारांश/ABSTRACT

प्राकृतिक आपदा अनुसंधान और पूर्वानुमान (डब्ल्यूआरएफ) मॉडल शैक्षिक और अनुसंधान समुदायों में क्षेत्रीय/स्थानीय स्तर पर मौसम का पूर्वानुमान करने के लिए एक महत्वपूर्ण उपकरण है। इस रिपोर्ट में भारतीय मानसून डेटा संघटन और विश्लेषण (IMDAA) से लिए गए प्रारंभ और सीमांत शर्तों (IC/BC) को WRF मॉडल के प्रारंभीकरण प्रक्रिया रूप में उपयोग करने का विवरण दिया गया है। उपयोगकर्ताओ के WRF के पूर्वसंस्करण सिस्टम (WPS) का संचालनशील IMDAA संघटन के साथ उपयोग करते समय उन्हें समस्याएं आ सकती हैं। इसलिए, इस दस्तावेज़ में उल्लेखित भंडार में एक समृद्ध सेट के स्क्रिप्ट्स और उपकरण शामिल हैं जो WPS में शामिल विभिन्न घटकों (GEOGRID, UNGRIB, और METGRID) में जुटे जटिल चरणों को स्वचालित करने के लिए हैं और साथ ही अनुसंधानकर्ताओं के लिए सुगम और त्रुटि-मुक्त अनुभव सुनिश्चित करते हैं। यह परियोजना WRF मॉडल के लिए IC/BC के रूप में उच्च रिज़ॉल्यूशन IMDAA संघटन डेटा का उपयोग करके मॉनसून अनुसंधान को तीव्रता से प्रोत्साहित करती है|

The Weather Research and Forecasting (WRF) model is an important tool for forecasting weather at regional/local scales in the academic and research communities. The process of WRF model initialization while using the Indian Monsoon Data Assimilation and Analysis (IMDAA) reanalysis dataset as initial and boundary conditions (IC/BC) are addressed in this report. Users might encounter issues while running the conventional WRF Preprocessing System (WPS) when utilizing the IMDAA reanalysis. Therefore, the repository mentioned in this document encompasses a comprehensive set of scripts and tools to automate the intricate steps involved in the WPS which has various components (GEOGRID, UNGRIB, and METGRID), ensuring a seamless and error-free experience for researchers. The project encourages monsoon research in an accelerated phase by utilizing the high-resolution IMDAA reanalysis data for IC/BC for the WRF model.

1. Introduction

Numerical Weather Prediction (NWP) models play a crucial role in understanding and predicting atmospheric phenomena. The Weather Research and Forecasting (WRF) model is a widely used NWP model that simulates atmospheric processes. The preprocessing system of WRF, known as the WRF Preprocessing System (WPS), is a critical step in setting up simulations by providing initial and boundary conditions (IC/BC) for the model. Indian Monsoon Data Assimilation and Analysis (IMDAA) reanalysis data has played a pivotal role in advancing meteorological research, providing comprehensive and multi-satellite observations for the study of precipitation patterns, understanding global rainfall variability and its impact on climate dynamics, enhance the accuracy of precipitation estimates, validate climate models, and investigate the complex interplay between atmospheric variables (Garg et al., 2023; Mondal et al., 2023; Routray et al., 2023). IMDAA reanalysis, which includes information such as atmospheric sounding profiles, surface observations, and satellite data, serves as valuable input for initializing and forcing the WRF model. There are so many research institutes and operational agencies that use IMDAA reanalysis for running the WRF model. However, users might encounter technical issues while running the conventional WPS on their own, specifically improper GRIB attributions for soil moisture (SM) data which might cause errors in the conventional WPS-generated metadata with operational IMDAA reanalysis. Thus, simulations employing the operational data might lack SM updates. Therefore, this technical document introduces an innovative solution for automating the WPS process using IMDAA reanalysis as IC/BC. This toolbox corrects the GRIB attributions of SM data and facilitates its integration into the UNGRIB component of the WPS. This integration ensures the proper updating of SM for BC during the WRF model simulations.

So, by automating the WPS with IMDAA reanalysis, we aim to streamline and enhance the efficiency of the pre-processing steps, allowing researchers and meteorologists to focus more on the scientific aspects of their simulations. The purpose of this project is to run the WPS program with IMDAA reanalysis to produce METGRID output files at user-defined intervals. The steps are as follows:

- This generates separate intermediate files (by UNGRIB) for different parameters. During this step, this toolbox handles the corrections of the GRIB attributes for SM data.
- Next is GEOGRID, which turns all static data into user-defined grid points.
- Then there is METGRID, which takes the GEOGRID output and uses these different parameters (in UNGRIB format) as intermediate data.

This repository follows GRIB2 parameter identities for the NCMRWF Unified Model (NCUM) output conventions. The *Appendix* section consists of the details of all the required tables that would be necessary to run the program.

2. Why this toolbox?

In the conventional WPS with operational IMDAA reanalysis, inaccuracies might arise due to improper GRIB attributions associated with SM data. It utilizes the default SM values, derived from tile information extracted during the GEOGRID process from static geographic data. As a consequence, simulations conducted using this data might lack the necessary updates for SM. To address this limitation, this toolbox intervenes by rectifying the GRIB attributions of the four-level SM data. Subsequently, the corrected SM data is fed into the UNGRIB component of the WPS, ensuring that the SM is appropriately inserted during the process of BC updates. This refinement enhances the accuracy of the WRF model simulations, particularly in scenarios where SM plays an important role.

In addition, traditional WPS setup can also be a time-consuming and intricate process, often requiring manual intervention and expertise. The integration of IMDAA reanalysis and automation into the WPS streamlines this workflow, offering a more accessible and efficient solution for researchers and meteorologists. This technical document provides a comprehensive guide to implementing and utilizing the automated WPS system with IMDAA reanalysis, allowing users to conduct advanced atmospheric simulations with greater ease and accuracy and allowing them to focus more on the core aspects of their modeling studies.

In the following sections, we will examine the IMDAA reanalysis, some basics of WRF, implementation details of this repository, provide step-by-step instructions for utilizing this automated solution for WRF preprocessing, and one case study.

3. IMDAA reanalysis

The IMDAA reanalysis dataset, providing a regional reanalysis with a high resolution of 12 km, has been made available to improve our understanding of the Indian monsoon and its variations. This study employed a 4D-Var data assimilation technique and utilized the U.K. Met Office (UKMO) Unified model output data to encompass the Indian monsoon region from 1979 to 2023 (Indira Rani et al., 2021). It is presently regarded as the most detailed atmospheric reanalysis available. Various sources, including Indian surface and upper air measures, are utilized in this reanalysis, using both conventional and satellite observations. Notably, some of these Indian measurements were previously unutilized in reanalyses. The reanalysis described in Indira Rani et al., 2021 encompasses various elements, including the verification and correction of observation quality control and bias, the data assimilation system, the analysis of land surfaces, and the validation of reanalysis products. The representation of notable weather events over India in each season by the

IMDAA reanalysis demonstrates a reasonably good agreement against data from the India Meteorological Department (IMD) and exhibits a reasonable comparison with the European Centre for Medium-Range Weather Forecasting (ECMWF) fifth-generation re-analysis (ERA5) data. The IMDAA reanalysis successfully captures crucial elements of the Indian summer monsoon (ISM). The characteristics of important semipermanent ISM features, such as the low-level jet and tropical easterly jet, in the IMDAA reanalysis, are consistent with those in ERA5. The IMDAA reanalysis well captured the average, year-to-year, and within-season fluctuations of ISM rainfall. The IMDAA captured the intricate details of an extraordinary occurrence of heavy rainfall over complex topography.

Sorting the downloaded IMDAA reanalysis

If a user only wants to download the IMDAA reanalysis, a shell script like below is required. Choose variables from *IMDAA 3-Hourly Pressure Level Dataset* or *IMDAA 1-Hourly Single Level Dataset* as per the requirement. Select the “GRIB” file format. Click the Submit button. After a successful process of submitting a query, the user will receive an email with a data download link (wget shell script).

```
#!/bin/bash
yyyy=2020     # year in four-digit
mm=01         # month in two-digit
dd=01         # day in two-digit
date=${yyyy}${mm}${dd}
mkdir finaldata

for param in UGRD-prl VGRD-prl TMP-prl TMP-sfc
do
        for hour in 00 06 12 18
        do
                rundate=${date}${hour}
                echo "$date of $hour"
                echo $rundate
                wgrib2 *_${param}_* -match_fs "=${rundate}" -match_fs \
                "anl" -grib_out finaldata/${param}_${date}_${hour}.grib2
        done
done
```

The user is required to download and run that shell script; it will download multiple GRIB2 files. A sample bash script extracts and sorts these four variables (UGRD-prl, VGRD-prl, TMP-prl, and TMP-sfc) in six hourly intervals as mentioned above where users can modify the variables and intervals as per their requirements.

4. Weather Research and Forecasting (WRF) model

The Weather Research and Forecasting (WRF) model is a popular NWP system used for atmospheric research as well as operational forecasting (Skamarock et al., 2008). The model, which was developed through a collaborative effort led by the National Center for Atmospheric Research (NCAR), has gained popularity due to its flexibility, versatility, and ability to simulate a wide range of meteorological phenomena. It is intended to simulate a wide range of atmospheric processes and phenomena, such as regional weather patterns, tropical cyclones, precipitation, and climate-related research. WRF is an open-source model that has been developed in collaboration with numerous institutions, universities, and meteorological agencies around the world. The WRF model is comprised of multiple components, which encompass the atmospheric dynamical core, physical parameterizations, data assimilation systems, and post-processing tools. Individuals can personalize the model by choosing various configurations and parameterizations that align with their particular research or forecasting requirements.

The WRF model provides support for various grid types, such as Cartesian, polar, and nested grids, enabling users to concentrate on particular areas of interest. The system can function at different spatial resolutions, encompassing both broad-scale climate simulations and more refined regional models that enable precise weather predictions. The WRF model is widely utilized by numerous meteorological agencies to conduct operational forecasting at both regional and local scales.

5. WRF Pre-Processing System

The WRF Pre-Processing System (WPS) is a compilation of Fortran and C programs that furnish data utilized as input for the real.exe and real_nmm.exe applications (https://www2.mmm.ucar.edu/wrf/users/download/get_source.html). The WPS consists of three programs (GEOGRID, UNGRIB, and METGRID) that collectively prepare input for real-data simulations in the actual program. Each of the programs performs one stage of the preparation as follows:

- The GEOGRID establishes the boundaries of the model domains, which are determined by the user. It aims to interpolate the static geographical data onto the grids.
- The UNGRIB function retrieves meteorological data from files encoded in GRIB format; and
- The METGRID performs the horizontal interpolation of the meteorological fields obtained from UNGRIB, onto the model grids specified by GEOGRID.

The process of vertically interpolating meteorological fields to WRF eta levels is carried out within this software. The WPS utilizes a build mechanism that closely resembles the one employed by the WRF model. This mechanism offers many alternatives for assembling the WPS on different systems. When there are accessible message-passing interface (MPI) libraries and appropriate compilers, the METGRID and GEOGRID programs can be built for distributed memory execution. This enables faster processing of large model domains. The UNGRIB program's tasks cannot be divided and executed simultaneously, hence it can only be run on a single CPU. Each program has been described in detail in the following sub-sections.

Geogrid

The primary function of a GEOGRID is to establish the simulation domains and interpolate different terrestrial data sets onto the model grids. The simulation domains are established based on the user's specifications in the "geogrid" namelist entry of the WPS NAMELIST file, namelist.wps. GEOGRID will automatically interpolate various data, such as soil categories, land use categories, terrain height, annual mean deep soil temperature, monthly vegetation fraction, monthly albedo, maximum snow albedo, and slope category, to the model grids. Additionally, it will compute the latitude, longitude, and map scale factors at each grid point. The WRF download website provides global data sets for each field (http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html). These data sets are time-invariant, meaning they do not change over time; therefore, they only need to be downloaded once. Some data sets are only accessible in one resolution, while others are provided at resolutions of 30 arc seconds, 2 arc minutes, 5 arc minutes, and 10 arc minutes. The user is not required to download all available resolutions for data collection. However, it is normally recommended to utilize a resolution of data that is close to that of the simulation domain to obtain more accurate interpolated fields. Nevertheless, users anticipating working with domains encompassing a wide variety of grid spacings may ultimately desire to obtain all the available resolutions of the static terrestrial data.

In addition to interpolating the default terrestrial fields, the GEOGRID program can interpolate a wide range of continuous and categorical fields onto the simulated domains. The simulation domain can incorporate new or additional data sets by utilizing the table file, GEOGRID.TBL, for interpolation. The

GEOGRID.TBL file specifies the characteristics of each field generated by GEOGRID. It outlines the interpolation techniques employed for each field and indicates the file system location of the corresponding dataset. The output generated by GEOGRID is stored in the WRF I/O API format. However, by choosing the NETwork Common Data Form (NetCDF) input/output (I/O) format, GEOGRID may be configured to save its output in NetCDF format. This allows for convenient display using external software tools like ncview, NCAR Command Language (NCL), and RIP4.

Ungrib

The UNGRIB program interprets GRIB files, extracts the data, and saves it in an easy-to-read format known as the intermediate format. The GRIB files consist of dynamic meteorological variables that change over time and are usually sourced from a different regional or global model, such as NCEP's North American Mesoscale Forecast System (NAM) or Global Forecast System (GFS) models. The UNGRIB software can interpret GRIB Edition 1 files and, if configured with a "GRIB2" option, it can also handle GRIB Edition 2 files. GRIB files often have an excess of fields that exceed the requirements for initializing WRF. Both iterations of the GRIB format employ distinct codes to designate the variables and levels present in the GRIB file. UNGRIB utilizes variable tables (Vtable), which are tables of codes, to specify the fields to be extracted from the GRIB file and transferred to the intermediate format. Information regarding the codes can be located in the World Meteorological Organization (WMO) GRIB documentation as well as in the documentation provided by the originating center. The UNGRIB package includes Vtables for commonly used GRIB model output files. Vtables are available for NAM 104 and 212 grids, NAM AWIP format, GFS, NCEP/NCAR Reanalysis stored at NCAR, RUC (pressure level data and hybrid coordinate data), Air Force Weather Agency (AFWA)'s agricultural meteorology (AGRMET) land surface model output, ECMWF, and other datasets. Users can generate their own Vtable for alternative model output by utilizing any of the existing Vtables as a reference. Additional information regarding the significance of the fields inside a Vtable can be found in the section dedicated to the creation and modification of Vtables. UNGRIB can generate intermediate data files in three different forms, which can be chosen by the user. WPS is a recently developed format that includes extra data that is beneficial for subsequent programs. SI is the former intermediate format used in the WRF system. The MM5 format is also included here to enable the usage of UNGRIB for providing GRIB2 input to the MM5 modeling system. WPS can utilize any of these formats to initialize WRF, although it is advisable to use the WPS format.

Metgrid

The METGRID application performs the horizontal interpolation of the intermediate-format meteorological data obtained by the UNGRIB program onto the simulation domains provided by the GEOGRID program. The interpolated METGRID output can be used in the WRF real program. The interpolation of dates performed by METGRID is determined by the "share" option in the namelist entry in the WPS NAMELIST file. Each simulation domain must have its own individual date range given in the NAMELIST. Due to the time-dependent nature of the METGRID program, similar to the UNGRIB program, METGRID is executed whenever a new simulation is initiated. The

METGRID.TBL file allows for precise control over the interpolation of each meteorological field. The METGRID.TBL file contains sections for each field, allowing for the specification of options such as interpolation methods, and grid staggering (e.g., U, V in advanced research WRF (ARW); H, V in non-hydrostatic mesoscale model (NMM)) for field interpolation. The output generated by METGRID is in the WRF I/O API format. By choosing the NetCDF I/O format, METGRID can be configured to write its output in NetCDF. This allows for convenient display using external software packages, such as the latest version of Read/Interpolate/Plot (RIP4).

6. Data formats

NWP model outputs utilize many file formats, including but not limited to fields file (PP/FF), NETCDF, GRIB, hierarchical data format (HDF), Binary, Binary Universal Form for the Representation (BUFR), and ASCII, for the storage of meteorological data. The Met Office, UK, has developed proprietary file formats known as the PP-format (Post Processing Format) and FF-format (Fields File Format) to store meteorological data. The Met Office's Unified Model is utilized to conduct simulations of meteorological phenomena, enabling its application in a range of weather and climate-related contexts. The data typically pertains to meteorology and is often particular to this field. It may consist of averaged data for various parameters, such as global surface temperatures or accumulations of rainfall at specified latitude-longitude coordinates. Nevertheless, the Unified Model can produce numerous advanced user-defined diagnostics in FF format. The files under consideration are binary streams that adhere to a proprietary file format. These streams can then undergo processing and conversion into alternative forms, such as the PP format, which are more universally compatible and portable. The primary rationale behind employing this particular format is to enhance the speed at which data can be transferred from the model to the storage medium.

The GRIB (GRIdded Binary or General Regularly-distributed Information in Binary form) format is widely employed in the field of meteorology for the storage of meteorological and climate forecast data. It is recognized as an open standard and is known for its compactness. The standardization of Basic Systems by the World Meteorological Organization's Commission is referred to as GRIB FM 92-IX, as specified in the WMO Manual on Codes No.306. At now, there are three iterations of the GRIB format. Version 0 was utilized to a restricted degree by initiatives such as TOGA and is presently not employed for operational purposes. The initial release of GRIB1 in 1994 is widely utilized by meteorological centers globally for the operational storage of NWP results. A more recent iteration of the GRIB format, referred to as GRIB 2 (reading and writing), was launched in 2003. This newer generation has been widely embraced by many operational NWP centers such as NCEP, ECMWF, and NCMRWF, among others, to disseminate data to the user community.

The GRIB2 format data is crucial for the execution of the WRF model, as it encapsulates meteorological information in a structured and efficient manner. IMDAA serves a pivotal role by providing WRF with the necessary GRIB2-formatted data, ensuring that the model receives the essential meteorological parameters required for its simulations. The seamless integration of GRIB2 data from IMDAA enhances the precision and reliability of WRF outputs, contributing to improved meteorological analyses and forecasts.

7. UMRider

Within the framework of the Unified Model (UM) Partnership, NCMRWF has been consistently progressing towards a cohesive approach to NWP modeling. As part of this effort, the NCMRWF routinely employs various configurations of the NCMRWF Unified Model (NCUM), such as the Global Model (with a resolution of 12 km) the Global-Ensemble Model (also with a resolution of 12 km), the Regional Model (with resolutions of 4 km, 1.5 km, and 330 m), the Regional-Ensemble Model (with a resolution of 4 km), etc. These models are complemented by both oceanic and atmospheric data assimilation systems. The University of Michigan utilizes the proprietary file formats PP-format (Post Processing Format) or FF-format (Field Files Format) to store forecast outputs. These formats were developed by the UK Met Office (UKMO) to improve the speed at which model data is written to disk. This is crucial for ensuring that real-time forecast delivery to forecasters is maintained within the required timeframe.

A parallel post-processing program called UMRider has been created at NCMRWF to address the difficulties faced by internal users in dealing with the UM fields file format for operational and research-oriented tasks (Arulalan, 2020). The purpose of UMRider is to perform post-processing on the forecast outputs of NCUM, such as NCUM-G Global Deterministic model post-processed products, Ocean State Forecast (OSF) model input products, HYbrid Coordinate Ocean Model (HYCOM) model input products, Hybrid Single Particle Lagrangian Integrated Trajectory (HYSPLIT) model input products, NCUM-R regional deterministic model post-processed products, NEPS-G Global Ensemble Model Post Processed products, THORPEX International Grand Global Ensemble (TIGGE) products, IMDAA regional reanalysis post-processed products, etc. This involves converting the outputs into open standard file formats such as GRIB2 or NetCDF4 while ensuring the spatial and temporal resolution meets the requirements for distribution to research purposes and other applications. The software is implemented using the Python programming language to perform post-processing tasks on the outputs of the Unified Model.

8. WRF preprocessing toolbox for IMDAA

Introduction

The WRF pre-processing toolbox designed for IMDAA utility has been created at the NCMRWF to automate the WPS component. Primarily implemented using the shell programming language, this toolbox serves a dual purpose. Firstly, this toolbox ensures that the GRIB attributions in the IMDAA SM data are accurate, thus ensuring its reliability to facilitate subsequent modeling. Secondly, it automates essential WPS tasks, including GEOGRID, UNGRIB, and METGRID processes. This comprehensive utility streamlines the pre-processing stages, ensures the accuracy of SM information, addresses the identified issues, and offers an efficient and accurate solution for preparing data inputs crucial for the WRF model's simulations and analyses.

Upon the successful execution of this utility, users are expected to follow the conventional workflow, engaging with *real.exe* and *wrf.exe* (or *tc.exe* and *ndown.exe*) as needed. It's important to note that the present utility focuses specifically on automating the WPS tasks only, including GEOGRID, UNGRIB, and METGRID processes. While it streamlines the pre-processing stages and enhances the quality of input data for the WRF model, the subsequent steps are not addressed within the scope of this current utility. Users should adhere to established procedures for running these components after the successful completion of the utility to ensure the seamless progression of the modeling process. This utility comprises the following elements:

- *runscript_ncmrwf.sh*
- *user_input.sh*

- *sample_user_input*
- *README file*
- *document*
- *tables*

The primary script for execution is '*runscript_ncmrwf.sh*', intended to be run by the user without the need for any code modifications. The '*user_input.sh*' file serves as a configuration file, requiring user input for essential details to facilitate the utility's proper functioning. A sample input file, '*sample_user_input*', is provided within the repository to guide users on completing '*user_input.sh*' before initiating the script. Additionally, a comprehensive set of instructions is available in the '*README.md*' file, outlining the step-by-step process for running the utility. For technical details, users can refer to the '*document*' folder, while the '*tables*' folder contains vital tables necessary for the script's execution. This organized structure aims to enhance user understanding and facilitate a smooth execution experience.

Code availability

The codes are available in the NCMRWF GitHub repository:

https://github.com/NCMRWF/WRF_IMDAAv2

HPC users

This toolbox is specifically designed for individuals executing their WPS tasks on local systems. Following the completion of WPS tasks, they can subsequently submit the REAL and WRF tasks on high-performance computers (HPC). The HPC users need to modify the code (for execution and submission in queues) as per their requirement (i.e., *runscript_ncmrwf.sh*). Or they might let us know so that we can provide solutions based on their HPC configurations.

Prerequisites

Before running this script, it is assumed that essential dependencies have been installed, including

- WGRIB2
- NETCDF4
- either OPENMPI or MPICH or Intel C compiler or Intel Fortran compiler (if parallel processing is required)
- NCKS
- ECCODES

If the user lacks any of the necessary libraries, the script will detect the absence and prompt an error message. Subsequently, the user will be guided to follow specific instructions to install the required libraries. This approach ensures that users are promptly informed of any missing dependencies and are provided with clear directives on how to address the issue, fostering a streamlined and user-friendly experience. Additionally, the WPS program must be installed to fulfill the prerequisites for the successful execution of this script. The script relies on these pre-installed components to effectively carry out its tasks, emphasizing the importance of having a properly configured environment before initiating the execution of the script.

User input

There are several prerequisites that users must address to successfully execute this program. The steps include:

- **Download Data from IMDAA Portal:**

  Users are required to manually download the necessary data from the IMDAA portal, as the script does not automate this process.

- **Fixing the Simulation Domain:**

  The script relies on the presence of the '*namelist.wps*' file for defining the study domain. Users need to ensure that this file is correctly configured according to their study requirements. The script does not generate this file; users must set it up themselves.

- **Complete User Input in 'user_input.sh':**

  Users need to fill out essential details in the '*user_input.sh*' file. This file contains the parameters and configurations necessary for the script to run successfully. Users are responsible for providing accurate information to tailor the script to their specific needs.

- **Install WPS and WRF Programs, and Required Libraries:**

  Before running the script, users must independently install the WPS and WRF programs, along with any libraries specified in Section 8.3 of the documentation. The script assumes the presence of these programs and libraries, and it does not handle the installation process.

By following these steps, users can prepare the required environment and configurations for the script to function as intended. Two important steps are discussed in the subsequent sub-sections.

Data downloading

Users are requested to download the required data exclusively from the IMDAA portal at https://rds.ncmrwf.gov.in/datasets, the steps have been described in Section 3.1. It is important to maintain the original structure of the downloaded data, without merging it. Please store all downloaded data in a single directory, preserving the organization as received from the IMDAA portal.

Table 1. *List of essential data for WRF (available in IMDAA reanalysis)*

	Name	**IMDAA reanalysis name**
1	U component wind in pressure level	UGRD-prl
2	V component wind in pressure level	VGRD-prl
3	Temperature in pressure level	TMP-prl
4	Surface temperature	TMP-sfc
5	Geopotential height in pressure level	HGT-prl
6	Relative humidity in pressure level	RH-prl
7	2-meter Relative humidity	RH-2m
8	Land mask	LAND-sfc
9	Surface pressure	PRES-sfc
10	Mean sea level pressure	PRMSL-msl
11	Soil temperature level 1	TSOIL-L1
12	Soil temperature level 2	TSOIL-L2
13	Soil temperature level 3	TSOIL-L3
14	Soil temperature level 4	TSOIL-L4
15	10-meter U wind	UGRD-10m
16	10-meter V wind	VGRD-10m
17	Soil moisture level 1	CISOILM-L1
18	Soil moisture level 2	CISOILM-L2
19	Soil moisture level 3	CISOILM-L3
20	Soil moisture level 4	CISOILM-L4
21	Water equivalent accumulated snow depth	WEASD-sfc

There is no need to consolidate or alter the structure of the downloaded data. The following data are essential to run the WRF model, thus users are requested to download only these essential data from the portal. All the essential data required for running the WRF has been mentioned in Table-1.

Providing relevant information in user_input.sh

The user_input.sh contains the information as described in Figure 1. To facilitate the smooth execution of the program, kindly adhere to the following recommended directory structure and input configurations as mentioned in the upcoming sub-sections.

```
# Path where you kept all downloaded IMDAA data together
imdaa_data_path=path_to_downloaded_imdaa_data_directory

# This is the path where you have installed the WPS package
wps_path=path_to_installed_WPS_directory

# Static data WPS_GEOG path
wps_geog_path=path_to_WPS_GEOG_static_data_directory

# Put your namelist.wps in this path
wps_namelist=full_path_of_namelist.wps

# number of processors if opting for parallel run
nproc=1

# set false if you do not want to repeat the processes from the start
SORT_IMDAA=true
RUN_GEOGRID=true
RUN_UNGRIB=true
```

Figure 1. *Screenshot of the user_input.sh file*

Assumed Directory Structure

- Home Directory: '*/home/workstation11*'
- WPS Installation: '*/home/workstation11/model/WPS-4.4*'
- Static Data (WPS_GEOG): '*/home/workstation11/model/WPS_GEOG*'

- Downloaded IMDAA reanalysis: '*/home/workstation11/imdaadata*'
- Git Repository: '*/home/workstation11/WRF_IMDAA*'
- Configure details in '*user_input.sh*' and execute '*runscript_ncmrwf.sh*' from this directory.

User Input Configuration (user_input.sh)

- '*imdaa_data_path*': Path, where all downloaded IMDAA reanalysis, is stored ('*/home/workstation11/imdaadata*').
- '*wps_path*': Path where the WPS package is installed ('*/home/workstation11/model/WPS-4.4*').
- '*wps_geog_path*': Path to static data (WPS_GEOG) ('*/home/workstation11/model/WPS_GEOG*').
- '*wps_namelist*': Path to your customized '*namelist.wps*' file ('*/home/workstation11/myfiles/namelist.wps*').
- '*nproc*': Number of processors for parallel run (set to 1 if not using parallel processing).
- '*SORT_IMDAA*': Set to '*true*' if you want to repeat processes from the start; otherwise, set to '*false*'.
- '*RUN_GEOGRID*': Set to '*true*' to execute the GEOGRID process; set to '*false*' if not needed.
- '*RUN_UNGRIB*': Set to '*true*' to execute the UNGRIB process; set to '*false*' if not needed.

Note on namelist.wps

- Place the original '*namelist.wps*' in a separate directory ('*/home/workstation11/myfiles*'). Avoid keeping it in the run area ('*/home/workstation11/WRF_IMDAA*'). Only reference the NAMELIST file in the '*wps_namelist*' option of '*user_input.sh*' file.

Following these recommendations will ensure proper directory organization and accurate input configurations for successful script execution.

Execution

- Create a directory in a chosen location and place this repository in it. Ensure that no additional files or folders are stored in this directory. Alternatively, clone the repository.
- Fill in all necessary details in the `user_input.sh` file as mentioned in section 8.4.2.
- Retain the integrity of the "tables" folder; no modifications are required.
- Grant executable permissions to the script:

  chmod +x runscript_ncmrwf.sh

- Execute the script by entering:

  ./runscript_ncmrwf.sh

Issue Tracker and Successful Completion

The script includes error tracking and user notifications for the following scenarios during execution:

- If the user mistakenly deletes any file from the repository before running the script.

- If there are alterations to the name/attributions within the '*user_input.sh*' file.
- If any nomenclature other than 'true' or 'false' is set for SORT_IMDAA, RUN_GEOGRID, or RUN_UNGRIB options in the '*user_input.sh*' file.
- If any file is deleted in the '*tables*' folder.
- If the '*namelist.wps*' file does not exist in the specified path.
- If any essential data is missing in the downloaded IMDAA reanalysis.
- If the simulation start date is mistakenly set later than the end date.
- If anything, other than 'yes' or 'no' is entered during the query for serial or parallel run.
- If the WPS directory does not exist or exists but does not have all components properly installed.
- If all the required libraries are not installed properly, as mentioned in Section 8.3.
- If mistakenly downloaded data for simulation has different dates or the other way round.
- If the static data folder does not exist or does not contain the required resolution data as specified in '*namelist.wps*'.

The program will echo the corresponding error message if it stops midway and guides the user through the necessary steps for a successful resolution. Upon successful completion of the script, the user will receive four useful information about the generated data essential for running the subsequent REAL or WRF part (which needs to be mentioned in *namelist.input*), including:

- NUM_METGRID_LEVELS
- NUM_SOIL_LAYERS

- NUM_LAND_CAT
- NUM_METGRID_SOIL_LEVELS

Case Study

A case study is conducted to assess the disparities between the two WRF model runs. The first utilized operational IMDAA reanalysis with the conventional WPS (referred to as IMDAA-oper), while the second employed data generated by this toolbox for WRF simulation (referred to as IMDAA-Github). The WRF-ARW model version 4.4.1, which is an advanced research version, has been utilized to perform numerical simulations (Skamarock et al., 2008). The simulations are conducted using a single domain over the forecast period of 48 hours, with a horizontal resolution of 4 kilometers (574×505 grid points). The detailed model configuration has been illustrated in Table 2. The IMDAA reanalysis, with a grid resolution of $0.12^0 \times 0.12^0$, has been utilized as the IC and BC for the WRF model at intervals of six hours.

For the case study one cyclone event has been taken for consideration, i.e., Super Cyclonic Storm (SuCS) Amphan. In May 2020, SuCS Amphan emerged as an exceptionally devastating tropical cyclone (16th to 21st May 2020). The cyclone inflicted extensive damage across Eastern India, particularly impacting the states of West Bengal and Odisha. Additionally, Bangladesh experienced severe consequences as a result of the cyclone's destructive force. At 12:00 UTC on May 20, the cyclone officially made landfall in West Bengal. During this event, the Joint Typhoon Warning Center (JTWC) approximated Amphan's sustained winds at 175 km/h over one minute. To capture the landfall (role of SM) and allow the spin-up, the event has been simulated from 19th May 00 UTC to 21st May 00 UTC.

Table 2. *WRF model configuration*

Horizontal resolution	4 km
Vertical levels	45
Grid system	Arakawa C-grid
Dynamics	Non hydrostatic (Skamarock et al., 2008)
Forecast length	48 hours
Time step	20 seconds
Projection	Mercator
Model output interval	3 hourlies
PBL scheme	MRF (Hong & Pan, 1996)
Microphysics scheme	WDM6 (Lim & Hong, 2010)
Short wave radiation	RRTMG (Iacono et al., 2008)
Long wave radiation	
Surface layer	Revised MM5 Scheme (Jiménez et al., 2012)
Land surface	Unified Noah land-surface model (Tewari et al., 2013)
Cumulus	Kain–Fritsch Scheme (Kain & Kain, 2004)

Data for the cyclone has been gathered from the Cyclone E-Atlas website (CEAw, www.rmcchennaieatlas.tn.nic.in), which is maintained by the Chennai Regional Meteorological Centre. So, both simulations have been performed identically using a similar model setup; the only difference is the process of creation of the METGRID output files which is required for the REAL run to create the IC/BC for the WRF run.

Soil moisture update

After performing the METGRID, the two processes created different metadata for the following files:

“met_em.d01.2020-05-19_00-00-00.nc”

“met_em.d01.2020-05-19_06-00-00.nc”

“met_em.d01.2020-05-19_12-00-00.nc”

“met_em.d01.2020-05-19_18-00-00.nc”

“met_em.d01.2020-05-20_00-00-00.nc”

“met_em.d01.2020-05-20_06-00-00.nc”

“met_em.d01.2020-05-20_12-00-00.nc”

“met_em.d01.2020-05-20_18-00-00.nc”

“met_em.d01.2020-05-21_00-00-00.nc”

The IMDAA-Github created files have four different SM data (100 – 300 cm, 35 – 100 cm, 10 – 35 cm, and 0 – 10 cm below ground), which is absent in the IMDAA-oper data. These are the extra metadata information obtained from IMDAA-Github data compared to IMDAA-oper after performing the command ‘*ncdump -h met_em.d01.2020-05-19_00-00-00.nc*’:

Table 3. *Metadata information*

```
num_sm_layers = 4;

float SM(Time, num_sm_layers, south_north, west_east) ;
        SM:FieldType = 104 ;
        SM:MemoryOrder = "XYZ" ;
        SM:units = "" ;
        SM:description = "" ;
        SM:stagger = "M" ;
        SM:sr_x = 1 ;
        SM:sr_y = 1 ;

FLAG_SM000010 = 1 ;
FLAG_SM010035 = 1 ;
FLAG_SM035100 = 1 ;
FLAG_SM100300 = 1 ;
```

Table 3 and Table 4 show all the volumetric SM data information for four layers which is missing in the IMDAA-oper output. The SM data for these four levels is also graphically represented in Figure 2, where the blue and red lines depict the simulations of IMDAA-Github and IMDAA-oper, respectively. Notably, the lower-level SM (35 – 300 cm) exhibits a decreasing trend, while the upper-level layer moisture (0 – 35 cm) shows an increasing trend up to 48 hours in the case of IMDAA-Github.

Table 4. *Metadata information*

```
float SM100300(Time, south_north, west_east) ;
        SM100300:FieldType = 104 ;
        SM100300:MemoryOrder = "XY " ;
        SM100300:units = "m3 m-3" ;
        SM100300:description = "Soil Moist 100-300 cm below gr layer" ;
        SM100300:stagger = "M" ;
        SM100300:sr_x = 1 ;
        SM100300:sr_y = 1 ;
float SM035100(Time, south_north, west_east) ;
        SM035100:FieldType = 104 ;
        SM035100:MemoryOrder = "XY " ;
        SM035100:units = "m3 m-3" ;
        SM035100:description = "Soil Moist 35-100 cm below grn layer" ;
        SM035100:stagger = "M" ;
        SM035100:sr_x = 1 ;
        SM035100:sr_y = 1 ;
float SM010035(Time, south_north, west_east) ;
        SM010035:FieldType = 104 ;
        SM010035:MemoryOrder = "XY " ;
        SM010035:units = "m3 m-3" ;
        SM010035:description = "Soil Moist 10-35 cm below grn layer" ;
        SM010035:stagger = "M" ;
        SM010035:sr_x = 1 ;
        SM010035:sr_y = 1 ;
float SM000010(Time, south_north, west_east) ;
        SM000010:FieldType = 104 ;
        SM000010:MemoryOrder = "XY " ;
        SM000010:units = "m3 m-3" ;
        SM000010:description = "Soil Moist 0-10 cm below grn layer (Up)" ;
        SM000010:stagger = "M" ;
        SM000010:sr_x = 1 ;
        SM000010:sr_y = 1 ;
```

Conversely, corresponding values for IMDAA-oper are non-existent. While the scientific implications require further examination, it is evident that SM undergoes notable variations throughout the simulation in the case of IMDAA-Github.

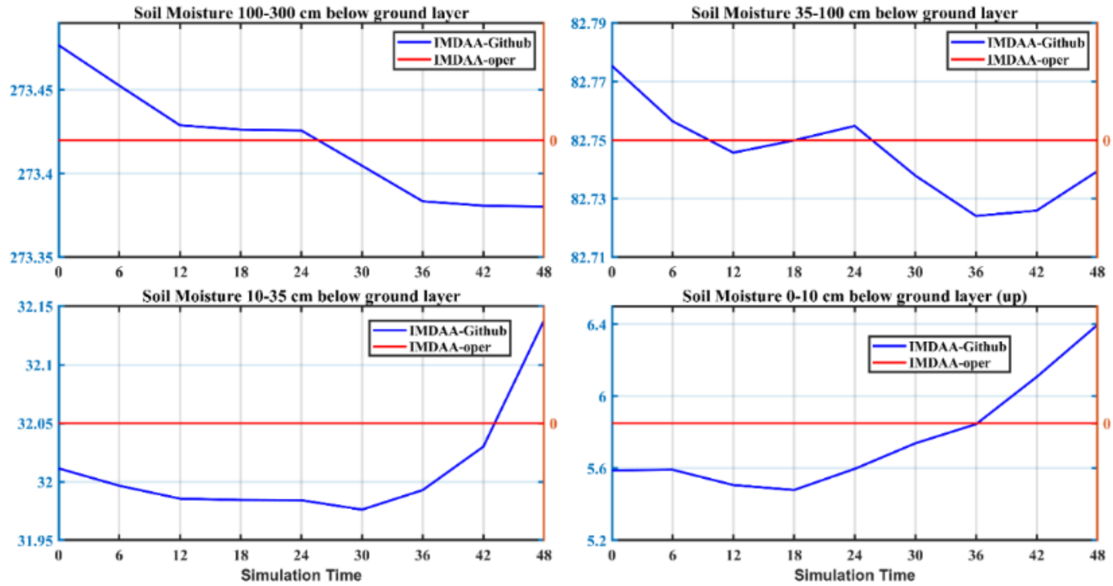

Figure 2. *Four-layer volumetric SM comparison. The red and blue lines represent IMDAA-oper and IMDAA-Github simulations, respectively. Units in* $m^3 m^{-3}$.

Tracks

Figure 3 represents the tracks obtained from these two simulations along with the IMD best track data. From the graphical representation, it is noted that the landfall location is a little closer to the observation for the IMDAA-Github simulation compared to the IMDAA-oper. Since the model configuration is uniform for both simulations, the difference in tracks and landfall location could be attributed to the influence of SM. In Figure 4, the track error is depicted in comparison to the IMD best track data for both simulations. IMDAA-Github holds a slight advantage over IMDAA-oper. Specifically, at the time of landfall (indicated by a dotted vertical line in Figure 4), there is a notable improvement (~17–18 km) in the accuracy of the landfall location for IMDAA-Github when compared to IMDAA-oper.

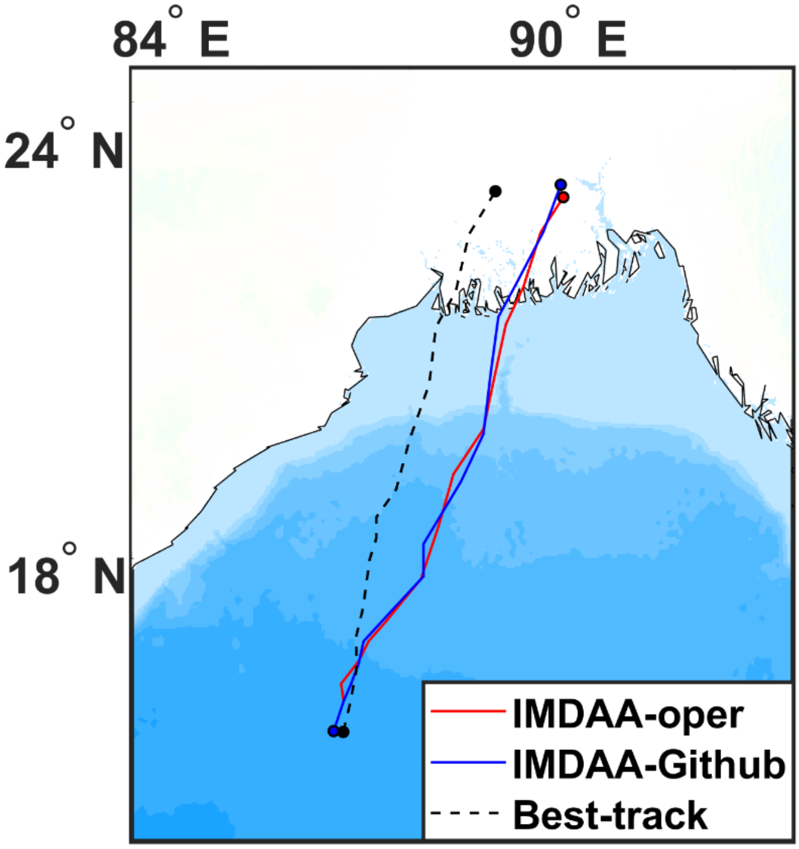

Figure 3. *Model simulated tracks for 48 hours based on initial condition 00 UTC 19 May 2020 with best track data. The red, blue, and dotted lines represent IMDAA-oper, IMDAA-Github, and Best-track, respectively.*

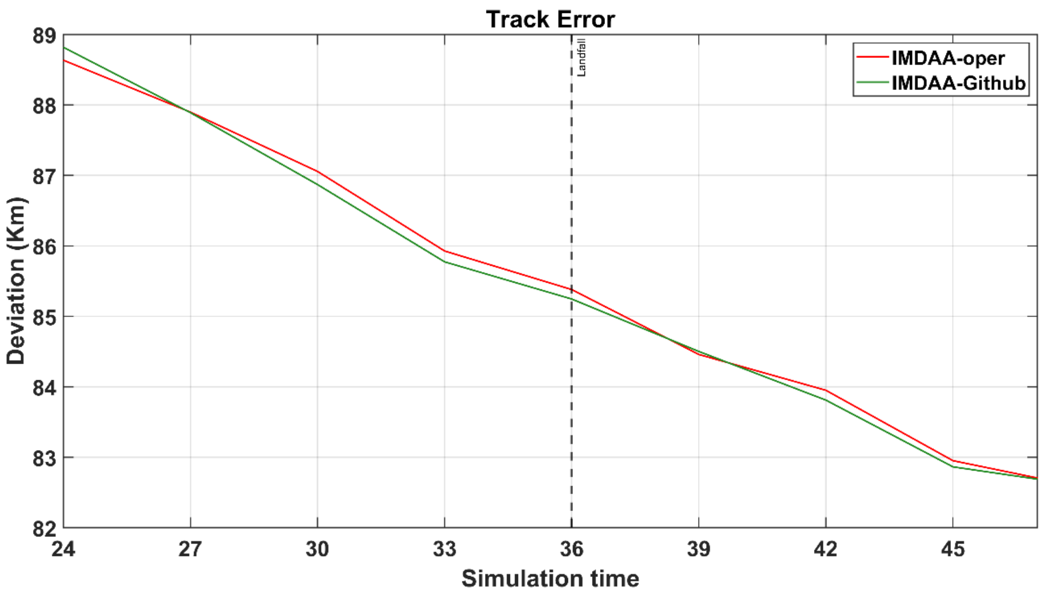

Figure 4. *Track errors where the red and green lines represent IMDAA-oper and IMDAA-Github simulations, respectively.*

Conclusions

In the case study, two WRF simulations have been conducted for a cyclone event (i.e., SuCS Amphan, 19th – 21st May 2020) for a lead time of up to 48 hours with identical model configurations. IMDAA-Github run corresponds to the WRF run, where the data for the REAL operation to create IC/BC was generated by using the toolbox using the same IMDAA reanalysis. On the other hand, IMDAA-oper corresponds to the WRF run, where data for REAL was generated by following the conventional WPS process with operational IMDAA reanalysis (no modifications in GRIB attributions). For the IMDAA-oper run, WRF took the SM data from the default tiles that had been obtained after the GEOGRID process, which is scientifically incorrect for any modeling study. The objective was to investigate whether the toolbox (inclusion of SM attributes) had any influence on WRF simulations.

It is widely recognized that SM plays a significant role in various weather phenomena, including extreme events. Notably, the corrected data from the toolbox has been observed to have a significant influence on model tracks, leading to an approximate 17 km improvement towards the observational landfall location. While other dynamical processes were not extensively explored within the scope of this work, a cursory examination was conducted. Consequently, the toolbox demonstrates a positive impact on WRF simulations, particularly when incorporating operational (web-based) IMDAA reanalysis as IC and BC.

Schematic diagram of the processes

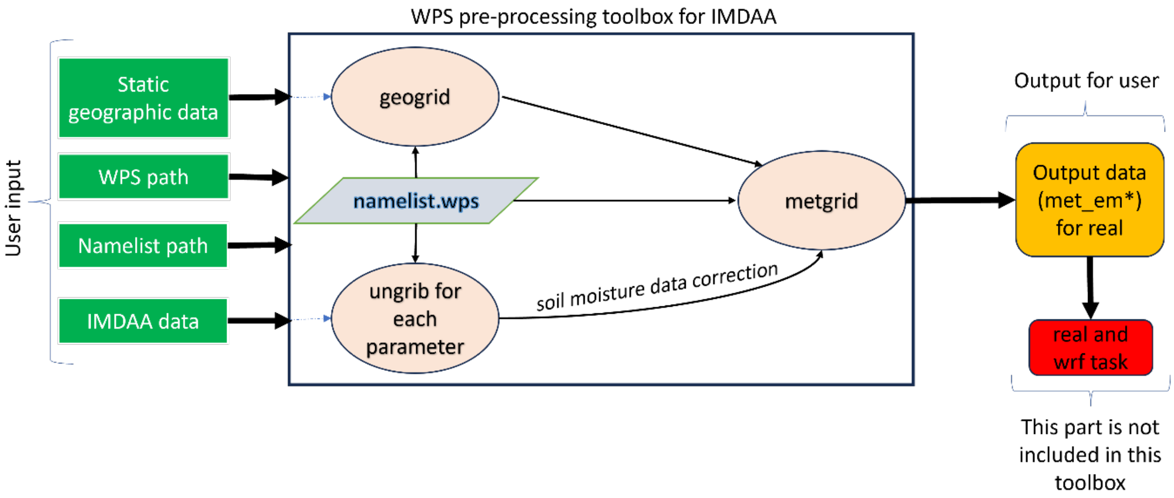

Figure 5. *Flow diagram of the processes involved in this toolbox.*

9. Acknowledgments

The work presented in this study utilized the computational capabilities of the MIHIR supercomputers at the NCMRWF. We extend our sincere gratitude to our colleagues and fellow scientists at NCMRWF, especially Mr. Deepak Singh Bisht, for their invaluable assistance. The authors thankfully acknowledge Dr. Syam Sankar and Dr. Gibies George for their helpful suggestions whenever needed throughout the project. Author VH acknowledges Dr. Ashutosh Kumar Sinha for helping in the Hindi translation of the abstract. Additionally, we express our thanks to the Head, NCMRWF for his encouragement and support, which greatly motivated us during this project. The authors are also thankfully expressing gratitude to the reviewer who evaluated the report.

References

Arulalan, T. (2020). *UMRider – A Parallel Post Processing Utility for Unified Model*. Technical Report. https://doi.org/https://www.ncmrwf.gov.in/Reports-eng/UMRider-Report_2020.pdf

Garg, S., Jena, P., Devi, U., & Azad, S. (2023). Performance evaluation of high-resolution IMDAA and IMERG for detecting cloudburst events over the Northwest Himalayas. *International Journal of Climatology*, *43*(8), 3730–3748. https://doi.org/10.1002/JOC.8055

Hong, S. Y., & Pan, H. L. (1996). Nonlocal Boundary Layer Vertical Diffusion in a Medium-Range Forecast Model. *Monthly Weather Review*, *124*(10), 2322–2339. https://doi.org/10.1175/1520-0493(1996)124<2322:nblvdi>2.0.co

Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., & Collins, W. D. (2008). Radiative forcing by long-lived greenhouse gases: Calculations with the AER radiative transfer models. *Journal of Geophysical Research: Atmospheres*, *113*(D13), 13103. https://doi.org/10.1029/2008JD009944

Indira Rani, S., Arulalan, T., George, J. P., Rajagopal, E. N., Renshaw, R., Maycock, A., Barker, D. M., & Rajeevan, M. (2021). IMDAA: High-Resolution Satellite-Era Reanalysis for the Indian Monsoon Region. *Journal of Climate*, *34*(12), 5109–5133. https://doi.org/10.1175/JCLI-D-20-0412.1

Jiménez, P. A., Dudhia, J., González-Rouco, J. F., Navarro, J., Montávez, J. P., & García-Bustamante, E. (2012). A Revised Scheme for the WRF Surface Layer Formulation. *Monthly Weather Review*, *140*(3), 898–918. https://doi.org/10.1175/MWR-D-11-00056.1

Kain, J. S., & Kain, J. (2004). The Kain - Fritsch convective parameterization: An update.

Journal of Applied Meteorology, *43*(1), 170–181. https://doi.org/10.1175/1520-0450(2004)043<0170:TKCPAU>2.0.CO;2

Lim, K.-S. S., & Hong, S.-Y. (2010). Development of an Effective Double-Moment Cloud Microphysics Scheme with Prognostic Cloud Condensation Nuclei (CCN) for Weather and Climate Models. *Monthly Weather Review*, *138*(5), 1587–1612. https://doi.org/10.1175/2009MWR2968.1

Mondal, U., Panda, S. K., Banerjee, B. K., Kumar, A., & Sharma, D. (2023). Performance evaluation of lightning potential index and flash count using WRF microphysical parameters over Rajasthan and West Bengal, India. *Dynamics of Atmospheres and Oceans*, *104*, 101404. https://doi.org/10.1016/J.DYNATMOCE.2023.101404

Routray, A., Dutta, D., Desamsetti, S., Patel, S. S., George, J. P., & Prasad, V. S. (2023). Study pre- and post-monsoon storms over NIO region using high resolution IMDAA reanalysis dataset. *Climate Dynamics*, *1*, 1–20. https://doi.org/10.1007/S00382-023-06933-1/FIGURES/12

Skamarock, W. C., Klemp, J. B., Dudhia, J. B., Gill, D. O., Barker, D. M., Duda, M. G., Huang, X.-Y., Wang, W., & Powers, J. G. (2008). A description of the Advanced Research WRF Version 3, NCAR Technical Note TN-475+STR. *Technical Report*, *June*, 113. https://doi.org/10.5065/D68S4MVH

Tewari, M., Chen, F., Wang, W., Dudhia, J., LeMone, M. A., Mitchell, K., M.Ek, Gayno, G., Wegiel, J., & Cuenca, R. H. (2013). Data Fusion Techniques for Improving Soil Moisture Deficit Using SMOS Satellite and WRF-NOAH Land Surface Model. *Water Resources Management*, *27*(15), 5069–5087. https://doi.org/10.1007/s11269-013-0452-7

Appendix

GRIB2 Local Table

- 0:1:0:10:29:1:1:192:FOG: Fog Area Cover:%
- 0:1:0:10:29:1:0:205:WETBPT:Wet Bulb Potential Temperature: K
- 0:1:0:10:29:1:1:193:EVARSS:Evaporation Rate From Soil Surface: kg m-2 s-1
- 0:1:0:10:29:1:1:194:EVARCA:Evaporation Rate From Canopy: kg m-2 s-1
- 0:1:0:10:29:1:1:195:EVAROS:Evaporation Rate From Open Sea: kg m-2 s-1
- 0:1:0:10:29:1:1:196:DENRR:Density * R * R:
- 0:1:0:10:29:1:4:192:NSWRFC:Net Short Wave Radiation Flux Corrected: W m-2
- 0:1:0:10:29:1:4:194:DUVB:Direct UV-B Solar Flux: W m-2
- 0:1:0:10:29:1:4:196:CSUSFS:Clear Sky Downward Solar Flux at Surface: W m-2
- 0:1:0:10:29:1:4:197:CSULFS:Clear Sky Upward Solar Flux at Surface: W m-2
- 0:1:0:10:29:1:4:198:CSUSFT:Clear Sky Upward Solar Flux at TOA: W m-2
- 0:1:0:10:29:1:5:195:CSULFT:Clear Sky Upward Long Wave Flux at TOA: W m-2
- 0:1:0:10:29:1:5:192:CSDLFS:Clear Sky Downward Long Wave Flux at Surface: W m-2
- 0:1:0:10:29:1:15:192:RAREF:Radar Reflectivity: dBZ
- 3:1:0:10:29:1:1:192:DAOT038:Dust Aerosol Optical Thickness at 0.38 µm: unitless
- 3:1:0:10:29:1:1:193:DAOT044:Dust Aerosol Optical Thickness at 0.44 µm: unitless
- 3:1:0:10:29:1:1:194:DAOT055:Dust Aerosol Optical Thickness at 0.55 µm: unitless
- 3:1:0:10:29:1:1:195:DAOT067:Dust Aerosol Optical Thickness at 0.67 µm: unitless
- 3:1:0:10:29:1:1:196:DAOT087:Dust Aerosol Optical Thickness at 0.87 µm: unitless
- 3:1:0:10:29:1:1:197:DAOT102:Dust Aerosol Optical Thickness at 1.02 µm: unitless
- 3:1:0:10:29:1:1:198:TCDAM:Total Column Dry Aerosols Mass: kg m-2
- 0:1:0:10:29:1:6:201:VLCDC:Very Low Cloud Cover: %
- 0:1:0:10:29:1:6:202:TCDCRO:Total Cloud Cover Assuming Random Overlap:%
- 0:1:0:10:29:1:6:203:TCDCMRO:Total Cloud Cover Assuming Maximum Random Overlap: %
- 0:1:0:10:29:1:6:204:TCDVFA:Total Cloud Volume Fraction in Atmosphere Layer: %
- 0:1:0:10:29:1:6:205:LCDVFA:Liquid Cloud Volume Fraction in Atmosphere Layer: %
- 0:1:0:10:29:1:6:206:ICDVFA:Ice Cloud Volume Fraction in Atmosphere Layer: %
- 0:1:0:10:29:1:6:207:LIGHTFC:Ligtning Flash Count: unitless
- 2:1:0:10:29:1:0:231:SSFCWROR:Sub Surface Water Runoff Rate : kg m-2 s-1
- 2:1:0:10:29:1:0:232:SUFCWR:Surface Upward Water Rate : kg m-2 s-1
- 2:1:0:10:29:1:0:193:DHFS:Downward Heat Flux in Soil: W m-2
- 2:1:0:10:29:1:0:25:SOILM:Soil Moisture: Kg m-3

Variable Table

GRIB1 Param	Level Type	From Level1	To Level2	METGRID Name	METGRID Units	METGRID Description	GRIB2 Discp	GRIB2 Catgy	GRIB2 Param	GRIB2 Level
11	100	*	-	TT	K	Temperature	0	0	0	100
33	100	*	-	UU	m s-1	U	0	2	2	100
34	100	*	-	VV	m s-1	V	0	2	3	100
52	100	*	-	RH	%	Relative Humidity	0	1	1	100
7	100	*	-	HGT	m	Height	0	3	5	100
11	105	1	-	TT	K	Temperature at 2m	0	0	0	103
52	105	1	-	RH	%	Relative Humidity at 2 m	0	1	1	103
33	105	10	-	UU	m s-1	U at 10 m	0	2	2	103
34	105	10	-	VV	m s-1	V at 10 m	0	2	3	103
1	1	0	-	PSFC	Pa	Surface Pressure	0	3	0	1
2	1	0	-	PMSL	Pa	Sea-level Pressure	0	3	1	101
144	112	0	10	SM000010	m3	Soil Moist 0-10 cm below grn layer (Up)	2	0	25	106
144	112	10	35	SM010035	m3	Soil Moist 10-35 cm below grn layer	2	0	25	106
144	112	35	100	SM035100	m3	Soil Moist 35-100 cm below grn layer	2	0	25	106
144	112	100	300	SM100300	m3	Soil Moist 100-300 cm below gr layer	2	0	25	106
11	112	0	10	ST000010	K	T 0-10 cm below ground layer (Upper)	2	0	2	106
11	112	10	35	ST010035	K	T 10-35 cm below ground layer (Upper)	2	0	2	106
11	112	35	100	ST035100	K	T 35-100 cm below ground layer (Upper)	2	0	2	106
11	112	100	300	ST100300	K	T 100-300 cm below ground layer (Bottom)	2	0	2	106
91	1	0	-	SEAICE	proprtn	Ice flag	10	2	0	1
81	1	0	-	LANDSEA	proprtn	Land/Sea flag (1=land, 0 or 2=sea)	2	0	0	1
7	1	0	-	SOILHGT	m	Terrain field of source analysis	2	0	7	1
11	1	0	-	SKINTEMP	K	Skin/Surface temperature	0	0	0	1
13	1	0	-	WEASD	kg m-2	Water equivalent acc. snow depth	0	1	13	1
-	0	0	-	SFCR	m	Surface roughness	2	0	1	1
-	2	0	-	VIS	m	Visibility	0	19	0	103

METGRID table

```
========================================
name=ST
  z_dim_name=num_st_layers
     derived=yes
# IF
     fill_lev =  10 : ST000010(200100)
     fill_lev =  40 : ST010040(200100)
     fill_lev = 100 : ST040100(200100)
     fill_lev = 200 : ST100200(200100)
# ELSE IF
     fill_lev =  10 : ST000010(200100)
     fill_lev =  35 : ST010035(200100)
     fill_lev = 100 : ST035100(200100)
     fill_lev = 300 : ST100300(200100)
# ELSE IF
     fill_lev =  10 : ST000010(200100)
     fill_lev = 200 : ST010200(200100)
# ELSE
     fill_lev =   7 : ST000007(200100)
     fill_lev =  28 : ST007028(200100)
     fill_lev = 100 : ST028100(200100)
     fill_lev = 255 : ST100255(200100)
========================================
name=SM
     z_dim_name=num_sm_layers
     derived=yes
# IF
     fill_lev =  10 : SM000010(200100)
     fill_lev =  40 : SM010040(200100)
     fill_lev = 100 : SM040100(200100)
     fill_lev = 200 : SM100200(200100)
# ELSE IF
     fill_lev =  10 : SM000010(200100)
     fill_lev =  35 : SM010035(200100)
     fill_lev = 100 : SM035100(200100)
     fill_lev = 300 : SM100300(200100)
# ELSE IF
     fill_lev =  10: SM000010(200100)
     fill_lev = 200 : SM010200(200100)
# ELSE
     fill_lev =   7 : SM000007(200100)
     fill_lev =  28 : SM007028(200100)
     fill_lev = 100 : SM028100(200100)
     fill_lev = 255 : SM100255(200100)
========================================
name=SW
     z_dim_name=num_sw_layers
     derived=yes
# IF
     fill_lev = 1 : SW000010(200100)
     fill_lev = 2 : SW010040(200100)
     fill_lev = 3 : SW040100(200100)
     fill_lev = 4 : SW100200(200100)
# ELSE IF
     fill_lev = 1 : SW000010(200100)
     fill_lev = 2 : SW010200(200100)
========================================
```

```
name=SOIL_LAYERS
        derived=yes
        z_dim_name=num_st_layers
        flag_in_output=FLAG_SOIL_LAYERS
        fill_lev=all:vertical_index; level_template=ST
========================================
name=SOILM
        z_dim_name=num_soilm_levels
        derived=yes
        fill_lev =   0 : SOILM000(200100)
        fill_lev =   5 : SOILM005(200100)
        fill_lev =  20 : SOILM020(200100)
        fill_lev =  40 : SOILM040(200100)
        fill_lev = 160 : SOILM160(200100)
        fill_lev = 300 : SOILM300(200100)
========================================
name=SOILT
        z_dim_name=num_soilt_levels
        derived=yes
        fill_lev =   0 : SOILT000(200100)
        fill_lev =   5 : SOILT005(200100)
        fill_lev =  20 : SOILT020(200100)
        fill_lev =  40 : SOILT040(200100)
        fill_lev = 160 : SOILT160(200100)
        fill_lev = 300 : SOILT300(200100)
        fill_lev =  49 : SOILT050(200100)
        fill_lev =  51 : SOILT050(200100)
========================================
name=SOIL_LEVELS
        derived=yes
        z_dim_name=num_soilt_levels
        flag_in_output=FLAG_SOIL_LEVELS
        fill_lev=all:vertical_index; level_template=SOILT
========================================
name=PRES
        z_dim_name=num_metgrid_levels
        derived=yes
        mandatory=yes    # MUST HAVE THIS FIELD
        fill_lev=all:PRESSURE
        fill_lev=200100:PSFC(200100)
        fill_lev=all:vertical_index; level_template=TT
========================================
name=LANDSEA
        interp_option=nearest_neighbor
        fill_missing=-1.
        fill_lev=200100:LANDMASK(1)
========================================
name=XICE ; output_name=SEAICE  # If we get XICE, use entry from SEAICE and
                                #   write the field out as SEAICE
========================================
name=SEAICE
        interp_option=four_pt+average_4pt
        interp_mask=LANDSEA(1)
        masked=land
        missing_value=-1.E30
        fill_missing=0.
========================================
name=SNOW
        interp_option=four_pt+average_4pt
        masked=water
```

```
	fill_missing=0.
	flag_in_output=FLAG_SNOW
========================================
name=SKINTEMP
	interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
	masked=both
	interp_land_mask  = LANDSEA(1)
	interp_water_mask = LANDSEA(0)
	fill_missing=0.
========================================
name=PSFC
	interp_option=four_pt+average_4pt
	fill_lev=200100:const(200100.)
	flag_in_output=FLAG_PSFC
========================================
name=VEGCAT
	interp_option=nearest_neighbor
	fill_missing=0.
	flag_in_output=FLAG_VEGCAT
========================================
name=CANWAT
	interp_option=four_pt
	fill_missing=0.
========================================
name=SOILCAT
	interp_option=nearest_neighbor
	fill_missing=0.
	flag_in_output=FLAG_SOILCAT
========================================
name=SW000010
	interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
	masked=water
	interp_mask=LANDSEA(0)
	missing_value=-1.E30
	fill_missing=1.
	flag_in_output=FLAG_SW000010
========================================
name=SW010040
	interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
	masked=water
	interp_mask=LANDSEA(0)
	missing_value=-1.E30
	fill_missing=1.
	flag_in_output=FLAG_SW010040
========================================
name=SW040100
	interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
	masked=water
	interp_mask=LANDSEA(0)
	missing_value=-1.E30
	fill_missing=1.
	flag_in_output=FLAG_SW040100
========================================
name=SW100200
	interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
	masked=water
	interp_mask=LANDSEA(0)
	missing_value=-1.E30
	fill_missing=1.
	flag_in_output=FLAG_SW100200
```

```
========================================
name=SW010200
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SW010200
========================================
name=SM000010
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SM000010
========================================
name=SM010035
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SM010035
========================================
name=SM010040
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SM010040
========================================
name=SM035100
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SM035100
========================================
name=SM040100
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SM040100
========================================
name=SM100200
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SM100200
========================================
name=SM100300
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
```

```
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=1.
        flag_in_output=FLAG_SM100300
========================================
name=SM010200
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=1.
        flag_in_output=FLAG_SM010200
========================================
name=ST000010
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=285.
        flag_in_output=FLAG_ST000010
========================================
name=ST010035
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=285.
        flag_in_output=FLAG_ST010035
========================================
name=ST010040
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=285.
        flag_in_output=FLAG_ST010040
========================================
name=ST035100
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=285.
        flag_in_output=FLAG_ST035100
========================================
name=ST040100
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=285.
        flag_in_output=FLAG_ST040100
========================================
name=ST100200
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=285.
        flag_in_output=FLAG_ST100200
```

```
=========================================
name=ST100300
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=285.
      flag_in_output=FLAG_ST100300
=========================================
name=ST010200
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=285.
      flag_in_output=FLAG_ST010200
=========================================
name=SM000007
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      fill_missing=1.
      flag_in_output=FLAG_SM000007
=========================================
name=SM007028
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      fill_missing=1.
      flag_in_output=FLAG_SM007028
=========================================
name=SM028100
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      fill_missing=1.
      flag_in_output=FLAG_SM028100
=========================================
name=SM100255
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      missing_value=-1.E30
      fill_missing=1.
      flag_in_output=FLAG_SM100255
=========================================
name=ST000007
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      fill_missing=285.
      flag_in_output=FLAG_ST000007
=========================================
name=ST007028
      interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
      masked=water
      interp_mask=LANDSEA(0)
      fill_missing=285.
      flag_in_output=FLAG_ST007028
=========================================
```

```
name=ST028100
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_ST028100
========================================
name=ST100255
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        missing_value=-1.E30
        fill_missing=285.
        flag_in_output=FLAG_ST100255
========================================
name=SOILM000
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=1.
        flag_in_output=FLAG_SOILM000
========================================
name=SOILM005
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=1.
        flag_in_output=FLAG_SOILM005
========================================
name=SOILM020
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=1.
        flag_in_output=FLAG_SOILM020
========================================
name=SOILM040
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=1.
        flag_in_output=FLAG_SOILM040
========================================
name=SOILM160
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=1.
        flag_in_output=FLAG_SOILM160
========================================
name=SOILM300
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=1.
        flag_in_output=FLAG_SOILM300
========================================
name=SOILT000
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
```

```
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_SOILT000
========================================
name=SOILT005
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_SOILT005
========================================
name=SOILT020
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_SOILT020
========================================
name=SOILT040
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_SOILT040
========================================
name=SOILT160
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_SOILT160
========================================
name=SOILT300
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_SOILT300
========================================
name=SOILT050
        interp_option=sixteen_pt+four_pt+wt_average_4pt+wt_average_16pt+search
        masked=water
        interp_mask=LANDSEA(0)
        fill_missing=285.
        flag_in_output=FLAG_SOILT050
========================================
name=PMSL
        interp_option=sixteen_pt+four_pt+average_4pt
        flag_in_output=FLAG_SLP
========================================
name=RH
        interp_option=sixteen_pt+four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(-1.E30)
========================================
name=SPECHUMD
        interp_option=sixteen_pt+four_pt+average_4pt
        fill_missing=0.
        missing_value=-1.E30
        fill_lev=200100:const(-1)
```

```
    flag_in_output=FLAG_SH
===========================================
name=SPECCLDL
    interp_option=sixteen_pt+four_pt+average_4pt
    fill_missing=0.
    missing_value=-1.E30
    fill_lev=200100:const(-1)
    flag_in_output=FLAG_SPECCLDL
===========================================
name=SPECCLDF
    interp_option=sixteen_pt+four_pt+average_4pt
    fill_missing=0.
    missing_value=-1.E30
    fill_lev=200100:const(-1)
    flag_in_output=FLAG_SPECCLDF
===========================================
name=TAVGSFC
    interp_option=four_pt
    fill_missing=0.
    fill_lev=200100:TT(200100)
    flag_in_output=FLAG_TAVGSFC
===========================================
name=T ; output_name=TT  # If we get T, use entry from TT and
                         #   write the field out as TT
===========================================
name=TT
    mandatory=yes    # MUST HAVE THIS FIELD
    interp_option=sixteen_pt+four_pt+average_4pt
    fill_missing=0.
    fill_lev=200100:const(-1.E30)
===========================================
name=U ; output_name=UU  # If we get U, use entry from UU and
                         #   write the field out as UU
===========================================
name=UU
    mandatory=yes    # MUST HAVE THIS FIELD
    interp_option=sixteen_pt+four_pt+average_4pt
    is_u_field=yes
    output_stagger=U
    fill_missing=0.
    fill_lev=200100:const(-1.E30)
===========================================
name=V ; output_name=VV  # If we get V, use entry from VV and
                         #   write the field out as VV
===========================================
name=VV
    mandatory=yes    # MUST HAVE THIS FIELD
    interp_option=sixteen_pt+four_pt+average_4pt
    is_v_field=yes
    output_stagger=V
    fill_missing=0.
    fill_lev=200100:const(-1.E30)
===========================================
name=SST
    interp_option=sixteen_pt+four_pt
    fill_missing=0.
    missing_value=-1.E30
    flag_in_output=FLAG_SST
===========================================
name=QV
```

```
        interp_option=four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
        flag_in_output=FLAG_QV
========================================
name=QR
        interp_option=four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
        flag_in_output=FLAG_QR
========================================
name=QC
        interp_option=four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
        flag_in_output=FLAG_QC
========================================
name=QI
        interp_option=four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
        flag_in_output=FLAG_QI
========================================
name=QS
        interp_option=four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
        flag_in_output=FLAG_QS
========================================
name=QG
        interp_option=four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
        flag_in_output=FLAG_QG
========================================
name=QNI
        interp_option=four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
        flag_in_output=FLAG_QNI
========================================
name=VPTMP
        interp_option=sixteen_pt+four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(0.)
========================================
name=PRESSURE
        interp_option=sixteen_pt+four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:PSFC(200100)
========================================
name=PRHO
        interp_option=sixteen_pt+four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:PSFC(200100)
        flag_in_output=FLAG_PRHO
========================================
name=GHT
        interp_option=sixteen_pt+four_pt+average_4pt
        fill_missing=0.
```

```
        fill_lev=200100:SOILHGT(200100)
        fill_lev=200100:HGT_M(1)
========================================
name=HGTT
        output=no
        interp_option=nearest_neighbor
========================================
name=SNOWH
        interp_option=four_pt+average_4pt
        masked=water
        fill_missing=0.
        flag_in_output=FLAG_SNOWH
========================================
name=SOILHGT
        interp_option=four_pt+average_4pt
        flag_in_output=FLAG_SOILHGT
========================================
name=H0ML
        interp_option=four_pt+average_4pt
        interp_mask=LANDSEA(1)
        masked=land
        fill_missing=0.
========================================
name=TMOML
        interp_option=nearest_neighbor
#        interp_option=sixteen_pt+four_pt+wt_average_4pt+search
        masked=land
        interp_mask=landmask(1)
        missing_value=200.
        fill_missing=-20.
========================================
name=T0
        interp_option=four_pt+average_4pt+search
        masked=land
        interp_mask=landmask(1)
        missing_value=200.
        fill_missing=-20.
========================================
name=QNWFA_JAN
   z_dim_name=num_qnwfa_levels
        interp_option=four_pt+average_4pt
========================================
name=QNWFA_FEB
   z_dim_name=num_qnwfa_levels
        interp_option=four_pt+average_4pt
========================================
name=QNWFA_MAR
   z_dim_name=num_qnwfa_levels
        interp_option=four_pt+average_4pt
========================================
name=QNWFA_APR
   z_dim_name=num_qnwfa_levels
        interp_option=four_pt+average_4pt
========================================
name=QNWFA_MAY
   z_dim_name=num_qnwfa_levels
        interp_option=four_pt+average_4pt
========================================
name=QNWFA_JUN
   z_dim_name=num_qnwfa_levels
```

```
      interp_option=four_pt+average_4pt
========================================
name=QNWFA_JUL
  z_dim_name=num_qnwfa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNWFA_AUG
  z_dim_name=num_qnwfa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNWFA_SEP
  z_dim_name=num_qnwfa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNWFA_OCT
  z_dim_name=num_qnwfa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNWFA_NOV
  z_dim_name=num_qnwfa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNWFA_DEC
  z_dim_name=num_qnwfa_levels
      interp_option=four_pt+average_4pt
      flag_in_output=FLAG_QNWFA
========================================
name=QNIFA_JAN
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_FEB
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_MAR
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_APR
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_MAY
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_JUN
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_JUL
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_AUG
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_SEP
```

```
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_OCT
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_NOV
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
========================================
name=QNIFA_DEC
  z_dim_name=num_qnifa_levels
      interp_option=four_pt+average_4pt
      flag_in_output=FLAG_QNIFA
========================================
name=UTROP
      interp_option=four_pt
      is_u_field=yes
      output_stagger=U
      flag_in_output=FLAG_UTROP
========================================
name=VTROP
      interp_option=four_pt
      is_v_field=yes
      output_stagger=V
      flag_in_output=FLAG_VTROP
========================================
name=TTROP
      interp_option=four_pt
      flag_in_output=FLAG_TTROP
========================================
name=PTROP
      interp_option=four_pt
      flag_in_output=FLAG_PTROP
========================================
name=PTROPNN
      interp_option=nearest_neighbor
      flag_in_output=FLAG_PTROPNN
========================================
name=HGTTROP
      interp_option=four_pt
      flag_in_output=FLAG_HGTTROP
========================================
name=UMAXW
      interp_option=four_pt
      is_u_field=yes
      output_stagger=U
      flag_in_output=FLAG_UMAXW
========================================
name=VMAXW
      interp_option=four_pt
      is_v_field=yes
      output_stagger=V
      flag_in_output=FLAG_VMAXW
========================================
name=TMAXW
      interp_option=four_pt
      flag_in_output=FLAG_TMAXW
========================================
```

```
name=PMAXW
        interp_option=four_pt
        flag_in_output=FLAG_PMAXW
========================================
name=PMAXWNN
        interp_option=nearest_neighbor
        flag_in_output=FLAG_PMAXWNN
========================================
name=HGTMAXW
        interp_option=four_pt
        flag_in_output=FLAG_HGTMAXW
========================================
name=WEASD
        interp_option=four_pt+average_4pt
        masked=water
        fill_missing=0.
        flag_in_output=FLAG_WEASD
========================================
name=DZDT
        interp_option=sixteen_pt+four_pt+average_4pt
        fill_missing=0.
        fill_lev=200100:const(-1)
        flag_in_output=FLAG_DZDT
========================================
```