

MPAS-Atmosphere Model User's Guide

Version 8.3.0

Last updated: 2 June 2025

Foreword

This user's guide describes the Model for Prediction Across Scales – Atmosphere (MPAS-A) Version 8.3.0. MPAS-A is the non-hydrostatic atmosphere model built within the MPAS framework. Users guides for other MPAS components, such as MPAS-Ocean, are separate from this guide.

The component models and framework that comprise MPAS are being developed collaboratively between Los Alamos National Laboratory (LANL) and the U.S. National Science Foundation National Center for Atmospheric Research (NSF NCAR). Common functionality required by different MPAS component models, such as parallel input/output, time management, block decomposition, etc., is provided by the MPAS framework, while development of specific component models, referred to in MPAS as *cores*, is handled by the individual development groups. Currently, LANL is responsible for the ocean, land-ice, and sea-ice cores, while NSF NCAR is responsible for the atmospheric core, MPAS-A.

MPAS is very much a collaborative development of both the shared architectures and the component models. There are a number of contributors to the developments leading to the MPAS-A solver, and many of these developments are shared with the ocean core. The C-grid Voronoi discretization is based on critical developments from John Thuburn, Todd Ringler, Bill Skamarock, and Joe Klemp. The mesh generation that enables the MPAS-A development received major contributions from Todd Ringler, Doug Jacobson, Max Gunzburger, and Lili Ju. Significant developments in the transport scheme were accomplished by Bill Skamarock and Almut Gassmann. The atmospheric physics has been taken from the Advanced Research WRF model; these physics have benefitted from the work of hundreds of scientists. On the framework side we leverage a number of outside packages that have received extensive development from a wide community, including netCDF (UCAR/Unidata) and PIO (as used in the Community Earth Systems Model, CESM).

The software developed for MPAS is open source, and it has been copyrighted under a BSD license. The simple copyright statement can be found at the beginning of MPAS source files and the complete copyright statement can be found in this user's guide or in the `LICENSE` file accompanying the source code.

We conclude by noting that this user's guide is a work in progress. We welcome suggestions for improvements to this guide, including additions, corrections, clarifications, etc. Updates to MPAS-A, including the most recent code, user's guide, and test cases, may be found at
<https://mpas-dev.github.io/>.

Contributors to this guide:

Michael Duda, Laura Fowler, Bill Skamarock, Conrad Roesch, Doug Jacobsen, and Todd Ringler.

The U.S. National Science Foundation National Center for Atmospheric Research (NSF NCAR) is operated by the University Corporation for Atmospheric Research (UCAR) and is sponsored by the National Science Foundation. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Contents

| | |
|---|-----------|
| 1 MPAS-Atmosphere Overview | 5 |
| 1.1 Features | 5 |
| 1.2 Model components | 6 |
| 2 MPAS-Atmosphere Quick Start Guide | 7 |
| 3 Building MPAS | 8 |
| 3.1 Prerequisites | 8 |
| 3.2 Compiling I/O Libraries | 8 |
| 3.2.1 NetCDF | 9 |
| 3.2.2 Parallel-NetCDF | 9 |
| 3.2.3 PIO | 9 |
| 3.3 Compiling MPAS | 9 |
| 3.4 Selecting a single-precision build | 11 |
| 3.5 Cleaning | 11 |
| 4 Preparing Meshes | 12 |
| 4.1 Graph partitioning with METIS | 12 |
| 4.2 Relocating refinement regions on the sphere | 12 |
| 4.3 Creating limited-area SCVT meshes | 13 |
| 5 Configuring Model Input and Output | 14 |
| 5.1 XML stream configuration files | 14 |
| 5.2 Optional stream attributes | 17 |
| 5.3 Stream definition examples | 18 |
| 5.3.1 Example: a single-precision output stream with one month of data per file | 18 |
| 5.3.2 Example: appending records to existing output files | 18 |
| 5.3.3 Example: referencing filename intervals to a time other than the start time | 19 |
| 6 Physics Suites | 21 |
| 6.1 Suite: mesoscale_reference | 21 |
| 6.2 Suite: convection_permitting | 22 |
| 6.3 Suite: none | 22 |
| 6.4 Selecting individual physics parameterizations | 22 |
| 7 Running the MPAS Non-hydrostatic Atmosphere Model | 24 |
| 7.1 Creating idealized ICs | 24 |
| 7.2 Creating real-data ICs | 25 |

| | | |
|----------|--|-----------|
| 7.2.1 | Static fields | 26 |
| 7.2.2 | Vertical grid generation and initial field interpolation | 27 |
| 7.3 | Running the model | 28 |
| 8 | Model Options | 31 |
| 8.1 | Periodic SST and Sea-ice Updates | 31 |
| 8.2 | Regional Simulation | 32 |
| 8.3 | Separate Stream for Invariant Fields | 33 |
| 8.3.1 | Preparing an Invariant File | 33 |
| 8.3.2 | Activating the Invariant Stream | 34 |
| 9 | Visualization | 35 |
| 9.1 | Meshes | 35 |
| 9.2 | Horizontal contour plots | 35 |
| 9.3 | Horizontal cell-filled plots | 35 |
| 9.4 | Vertical cross-sections | 36 |
| A | Initialization Namelist Options | 38 |
| A.1 | nhyd_model | 38 |
| A.2 | dimensions | 40 |
| A.3 | data_sources | 41 |
| A.4 | vertical_grid | 43 |
| A.5 | interpolation_control | 45 |
| A.6 | preproc_stages | 45 |
| A.7 | physics | 46 |
| A.8 | io | 47 |
| A.9 | decomposition | 47 |
| B | Model Namelist Options | 49 |
| B.1 | nhyd_model | 49 |
| B.2 | damping | 55 |
| B.3 | limited_area | 57 |
| B.4 | io | 57 |
| B.5 | decomposition | 58 |
| B.6 | restart | 58 |
| B.7 | printout | 59 |
| B.8 | IAU | 59 |
| B.9 | assimilation | 60 |
| B.10 | development | 60 |
| B.11 | physics | 60 |
| B.12 | soundings | 71 |
| B.13 | physics_lsm_noahmp | 72 |
| C | Grid Description | 76 |
| C.1 | Horizontal grid | 76 |
| C.2 | Vertical grid | 80 |

| | |
|--------------------------------------|------------|
| D Description of Model Fields | 83 |
| Fields A–E | 83 |
| Fields F–O | 119 |
| Fields P–S | 155 |
| Fields T–Z | 201 |
| E MPAS Copyright | 230 |
| F Revision History | 231 |

Chapter 1

MPAS-Atmosphere Overview

The Model for Prediction Across Scales – Atmosphere (MPAS-A) is a non-hydrostatic atmosphere model that is part of a family of Earth-system component models collectively known as MPAS. All MPAS models have in common their use of centroidal Voronoi tessellations for their horizontal meshes, which has motivated the development of a common software framework that provides a high-level driver program and infrastructure for providing parallel execution, input and output, and other software infrastructure.

1.1 Features

Important features of MPAS-A include:

- Fully-compressible, non-hydrostatic dynamics
- Split-explicit Runge-Kutta time integration
- Exact conservation of dry-air mass and scalar mass
- Positive-definite and monotonic transport options
- Generalized terrain-following height coordinate
- Support for unstructured variable-resolution (horizontal) mesh integrations for the sphere and Cartesian planes
- Support for global and limited-area simulation domains

At present, MPAS-A includes parameterizations of physical processes taken from the Weather Research and Forecasting (WRF) Model¹. Specifically, MPAS-A has support for:

- Radiation: CAM and RRTMG long-wave and short-wave radiation schemes
- Land-surface: NOAH land-surface model
- Surface-layer: Monin-Obukhov and MYNN
- Boundary-layer: YSU and MYNN PBL schemes

¹<https://www.mmm.ucar.edu/weather-research-and-forecasting-model>

- Convection: Kain-Fritsch Tiedtke, New Tiedtke, and Grell-Freitas convection parameterizations
- Cloud microphysics: WSM6, Kessler, and Thompson schemes

1.2 Model components

MPAS-A is comprised of two main components: the model, which includes atmospheric dynamics and physics; and an initialization component for generating initial conditions for the atmospheric and land-surface state, update files for sea-surface temperature and sea ice, and lateral boundary conditions. Both components (model and initialization) are built as *cores* within the MPAS software framework and make use of the same driver program and software infrastructure. However, each component is compiled as a separate executable.

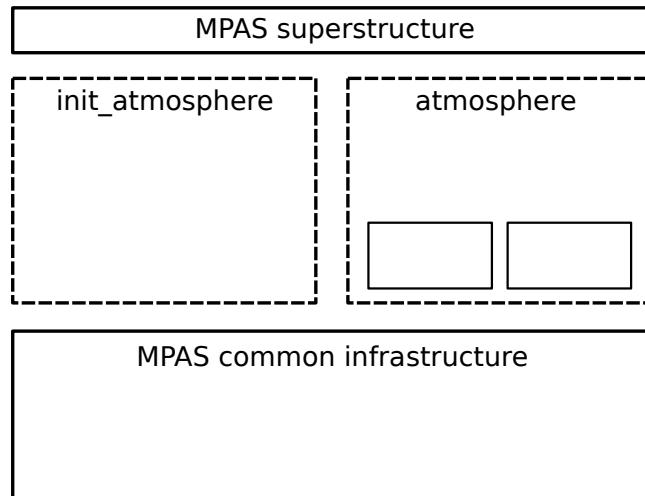


Figure 1.1: The initialization and model components of MPAS-A are built as separate *cores* within the MPAS framework.

A succinct description of building and running MPAS-A is given in Chapter 2, the Quick Start Guide. Detailed instructions for building these components are given in Chapter 3, and the basic steps to create initial conditions and run the MPAS-A model are outlined in Chapter 7.

Chapter 2

MPAS-Atmosphere Quick Start Guide

This chapter provides MPAS-Atmosphere users with a high-level description of the general process of building and running the model. It is meant simply as a brief overview of the process, with more detailed descriptions of each step are provided in later chapters.

In general, the build process follows the steps, below.

1. Build or locate an MPI implementation (MPICH, OpenMPI, MVAPICH2, MPT, etc; Section 3.1).
2. Build the serial netCDF library (Section 3.2.1).
3. Build the parallel-netCDF library (Section 3.2.2).
4. Build the Parallel I/O (PIO) library (Section 3.2.3).
5. (OPTIONAL) build the METIS package (Section 4.1).
6. Obtain the MPAS-Model source code.
7. Build the *init_atmosphere* and *atmosphere* cores (Section 3.3).

After completing these steps, executable files named `init_atmosphere_model`, `atmosphere_model`, and `build_tables` should have been created in the top-level MPAS-Model directory. Once all three executables have been created, a basic global simulation can be performed using the following steps.

1. Create a run directory.
2. Link the `init_atmosphere_model` and `atmosphere_model` executables to the run directory, as well as physics lookup tables (`src/core_atmosphere/physics/physics_wrf/files/*`).
3. Copy the `namelist.*`, `streams.*`, and `stream_list.*` files to the run directory.
4. Edit the namelist files and the stream files appropriately (Chapter 7).
5. (OPTIONAL) prepare meshes for the simulation (Chapter 4).
6. Run `init_atmosphere_model` to create initial conditions (Section 7.1 or Section 7.2).
7. Run `atmosphere_model` to perform model integration (Section 7.3).

Chapter 3

Building MPAS

3.1 Prerequisites

To build MPAS, compatible C and Fortran compilers are required. Additionally, the MPAS software relies on the PIO parallel I/O library to read and write model fields, and the PIO library requires the standard netCDF library as well as the parallel-netCDF library from Argonne National Laboratory. All libraries must be compiled with the same compilers that will be used to build MPAS. Section 3.2 summarizes the basic procedure of installing the required I/O libraries for MPAS.

In order for the MPAS makefiles to find the PIO, parallel-netCDF, and netCDF include files and libraries, the environment variables `PIO`, `PNETCDF`, and `NETCDF` should be set to the root installation directories of the PIO, parallel-netCDF, and netCDF installations, respectively.

An MPI installation such as MPICH or OpenMPI is also required, and there is no option to build a serial version of the MPAS executables. MPAS-Atmosphere v5.0 introduces the capability to use hybrid parallelism using MPI and OpenMP; however, the use of OpenMP *should be considered experimental* and generally does not offer any performance advantage. The primary reason for releasing a shared-memory capability is to make this code available to collaborators for future development.

3.2 Compiling I/O Libraries

IMPORTANT NOTE: *The instructions provided in this section for installing libraries have been successfully used by MPAS developers, but due to differences in library versions, compilers, and system configurations, it is recommended that users consult documentation provided by individual library vendors should problems arise during installation. The MPAS developers cannot assume responsibility for third-party libraries.*

Although most recent versions of the netCDF and parallel-netCDF libraries should work, the most tested versions of these libraries are netCDF 4.4.x and parallel-netCDF 1.8.x. Users are strongly encouraged to use either the latest PIO 2.x version from <https://github.com/NCAR/ParallelIO/>, or PIO versions 1.7.1 or 1.9.23, as other versions have not been tested or are known to not work with MPAS. The netCDF and parallel-netCDF libraries must be installed before building the PIO library.

3.2.1 NetCDF

Version 4.4.x of the netCDF library may be downloaded from <http://www.unidata.ucar.edu/downloads/netcdf/index.jsp>. The Unidata page provides detailed instructions for building the netCDF C and Fortran libraries; both the C and Fortran interfaces are needed by PIO. If netCDF-4 support is desired, the zlib and HDF5 libraries will need to be installed prior to building netCDF. *Before proceeding to compile PIO the NETCDF environment variable should be set to the netCDF root installation directory.*

3.2.2 Parallel-NetCDF

Version 1.8.x of the parallel-netCDF library may be downloaded from <https://trac.mcs.anl.gov/projects/parallel-netcdf/wiki/Download>. *Before proceeding to compile PIO the PNETCDF environment variable should be set to the parallel-netCDF root installation directory.*

3.2.3 PIO

Beginning with the MPAS v5.2 release, the either of the PIO 1.x or 2.x library versions may be used. The two major versions have slightly different APIs; by default, the MPAS build system assumes the PIO 1.x API, but the PIO 2.x library versions may be used by adding the USE_PIO2=true option when compiling MPAS as described in Section 3.3. If compiling with the PIO 1.x library versions, users are strongly encouraged to choose either PIO 1.7.1 or PIO 1.9.23, as other 1.x versions may not work; these two specific versions may be obtained from https://github.com/NCAR/ParallelIO/releases/tag/pio1_7_1 and https://github.com/NCAR/ParallelIO/releases/tag/pio1_9_23.

The PIO 2.x library versions support integrated performance timing with the GPTL library; however, the MPAS infrastructure does not currently provide calls to initialize this library when it is used in PIO 2.x. Therefore, it is recommended to add -DPIO_ENABLE_TIMING=OFF when running the cmake command to build PIO 2.x versions.

After PIO is built and installed the `PIO` environment variable should be set to the directory where PIO was installed. Recent versions of PIO support the specification of an installation prefix, while some older versions do not, in which case the `PIO` environment variable should be set to the directory where PIO was compiled.

3.3 Compiling MPAS

IMPORTANT NOTE: *Before compiling MPAS, the NETCDF, PNETCDF, and PIO environment variables must be set to the library installation directories as described in the previous section.*

The MPAS code uses only the ‘make’ utility for compilation. Rather than employing a separate configuration step before building the code, all information about compilers, compiler flags, etc., is contained in the top-level `Makefile`; each supported combination of compilers (i.e., a configuration) is included in the `Makefile` as a separate make target, and the user selects among these configurations by running `make` with the name of a build target specified on the command-line, e.g.,

```
> make gfortran
```

to build the code using the GNU Fortran and C compilers. Some of the available targets are listed in the table below, and additional targets can be added by simply editing the `Makefile` in the top-level directory.

| Target | Fortran compiler | C compiler | MPI wrappers |
|----------|------------------|------------|---------------------|
| xlf | xlf90 | xlc | mpxlf90 / mpcc |
| pgi | pgf90 | pgcc | mpif90 / mpicc |
| ifort | fort | gcc | mpif90 / mpicc |
| gfortran | gfortran | gcc | mpif90 / mpicc |
| llvm | flang | clang | mpifort / mpicc |
| bluegene | bgxlf95_r | bgxlc_r | mpxlf95_r / mpxlc_r |

The MPAS framework supports multiple *cores* — currently a shallow water model, an ocean model, a land-ice model, a non-hydrostatic atmosphere model, and a non-hydrostatic atmosphere initialization core — so the build process must be told which core to build. This is done by either setting the environment variable **CORE** to the name of the model core to build, or by specifying the core to be built explicitly on the command-line when running `make`. For the atmosphere core, for example, one may run either

```
> setenv CORE atmosphere
> make gfortran
```

or

```
> make gfortran CORE=atmosphere
```

If the **CORE** environment variable is set and a core is specified on the command-line, the command-line value takes precedence; if no core is specified, either on the command line or via the **CORE** environment variable, the build process will stop with an error message stating such. Assuming compilation is successful, the model executable, named `${CORE}_model` (e.g., `atmosphere_model`), should be created in the top-level MPAS directory.

In order to get a list of available cores, one can simply run the top-level `Makefile` without setting the **CORE** environment variable or passing the core via the command-line. An example of the output from this can be seen below.

```
> make
( make error )
make[1]: Entering directory '/scratch/MPAS-Release'

Usage: make target CORE=[core] [options]

Example targets:
  ifort
  gfortran
  xlf
  pgi

Available Cores:
  atmosphere
  init_atmosphere
  landice
  ocean
  seaice
```

```

sw
test

Available Options:
  DEBUG=true      - builds debug version. Default is optimized version.
  USE_PAPI=true   - builds version using PAPI for timers. Default is off.
  TAU=true        - builds version using TAU hooks for profiling. Default is off.
  AUTOCLEAN=true  - forces a clean of infrastructure prior to build new core.
  GEN_F90=true    - Generates intermediate .f90 files through CPP, and builds with them.
  TIMER_LIB=opt   - Selects the timer library interface to be used for profiling the model. Options are
                    TIMER_LIB=native - Uses native built-in timers in MPAS
                    TIMER_LIB=gptl - Uses gptl for the timer interface instead of the native interface
                    TIMER_LIB=tau - Uses TAU for the timer interface instead of the native interface
  OPENMP=true     - builds and links with OpenMP flags. Default is to not use OpenMP.
  USE PIO2=true   - links with the PIO 2 library. Default is to use the PIO 1.x library.
  PRECISION=single - builds with default single-precision real kind. Default is to use double-precision

```

Ensure that NETCDF, PNETCDF, PIO, and PAPI (if USE_PAPI=true) are environment variables that point to the absolute paths for the libraries.

```

***** ERROR *****
No CORE specified. Quitting.
***** ERROR *****

```

3.4 Selecting a single-precision build

Beginning with version 2.0, MPAS-Atmosphere can be compiled and run in single-precision, offering faster model execution and smaller input and output files. Beginning with version 5.0, the selection of the model precision can be made on the command-line, with no need to edit the `Makefile`. To compile a single-precision MPAS-Atmosphere executable, add `PRECISION=single` to the build command, e.g.,

```
> make gfortran CORE=atmosphere PRECISION=single
```

Regardless of which precision the MPAS-Atmosphere `init_atmosphere` and `atmosphere` cores were compiled with, either single- or double-precision input files may be used. In general, the MPAS infrastructure should correctly detect the precision of input files, but one may also explicitly specify the precision of files in an input stream by adding the `precision` attribute to the stream definition as described in Section 5.2.

3.5 Cleaning

To remove all files that were created when the model was built, including the model executable itself, `make` may be run for the ‘clean’ target:

```
> make clean
```

As with compiling, the core to be cleaned is specified by the `CORE` environment variable, or by specifying a core explicitly on the command-line with `CORE=`.

Chapter 4

Preparing Meshes

This chapter describes the steps used to prepare SCVT meshes for use in MPAS-Atmosphere. For quasi-uniform meshes, very little preparation is actually needed, and generally, one only needs to prepare mesh decomposition files — files that describe the decomposition of the SCVT mesh across processors — when running MPAS-Atmosphere using multiple MPI tasks. The procedure for creating these mesh decomposition files is described in the first section.

For variable-resolution SCVT meshes, the area of mesh refinement may be rotated to any part of the sphere using a program, `grid_rotate`, described in the second section. This utility program may be obtained from the MPAS-Atmosphere download page.

When running limited-area simulations, the limited-area mesh may be produced by subsetting the elements in an existing SCVT mesh as described in the third section.

4.1 Graph partitioning with METIS

Before MPAS can be run in parallel, a mesh decomposition file with an appropriate number of partitions (equal to the number of MPI tasks that will be used) is required. A limited number of mesh decomposition files, named `graph.info.part.*`, are provided with each mesh, as is the mesh connectivity file, named `graph.info`. If the number of MPI tasks to be used when running MPAS matches one of the pre-computed decomposition files, then there is no need to run METIS.

In order to create new mesh decomposition files for some particular number of MPI tasks, only the `graph.info` file is required. The currently supported method for partitioning a `graph.info` file uses the METIS software (<http://glaros.dtc.umn.edu/gkhome/views/metis>). The serial graph partitioning program, METIS (rather than ParMETIS or hMETIS) should be sufficient for quickly partitioning any mesh usable by MPAS.

After installing METIS, a `graph.info` file may be partitioned into N partitions by running

```
> gmetis -minconn -contig -niter=200 graph.info N
```

where N is the required number of partitions. The resulting file, `graph.info.part.N`, can then be copied into the MPAS run directory before running the model with N MPI tasks.

4.2 Relocating refinement regions on the sphere

The purpose of the `grid_rotate` program is simply to rotate an MPAS mesh file, moving a refinement region from one geographic location to another, so that the mesh can be re-used for different applications. This utility was developed out of the need to save computational resources, since generating an SCVT — particularly one with a large number of generating points or a high degree of refinement — can take considerable time.

To build the `grid_rotate` program, the Makefile should first be edited to set the Fortran compiler to be used; if the netCDF installation pointed to by the NETCDF environment variable was build with a separate Fortran interface library, it will also be necessary to add `-lncdf` just before `-lncdf` in the Makefile. After editing the Makefile, running ‘make’ should result in a `grid_rotate` executable file.

Besides the MPAS grid file to be rotated, `grid_rotate` requires a namelist file, `namelist.input`, which specifies the rotation to be applied to the mesh. The namelist variables are summarized in the table below

| | |
|---|---|
| <code>config_original_latitude_degrees</code> | original latitude of any point on the sphere |
| <code>config_original_longitude_degrees</code> | original longitude of any point on the sphere |
| <code>config_new_latitude_degrees</code> | latitude to which the original point should be shifted |
| <code>config_new_longitude_degrees</code> | longitude to which the original point should be shifted |
| <code>config_birdseye_rotation_counter_clockwise_degrees</code> | rotation about a vector from the sphere center through the original point |

Essentially, one chooses any point on the sphere, decides where that point should be shifted to, and specifies any change to the orientation (i.e., rotation) of the mesh about that point.

Having set the rotation parameters in the `namelist.input` file, the `grid_rotate` program should be run with two command-line options specifying the original grid file name and the name of the rotated grid file to be produced, e.g.,

```
> grid_rotate grid.nc grid_SE_Asia_refinement.nc
```

The original grid file will not be altered, and a new, rotated grid file will be created. The NCL script `mesh.ncl` may be used to plot either of the original or rotated grid files after suitable setting the name of the grid file in the script.

Note: The `grid_rotate` program initializes the new, rotated grid file to a copy of the original grid file. If the original grid file has only read permission (i.e., no write permission), then so will the copy, and consequently, the `grid_rotate` program will fail when attempting to update the fields in the copy.

4.3 Creating limited-area SCVT meshes

The process of creating a limited-area (regional) mesh for MPAS-Atmosphere involves selecting any existing mesh – either a quasi-uniform mesh, or a variable-resolution mesh that has been rotated as in section 4.2 – describing the geographical region to be extracted from that mesh, and running the limited-area Python program to extract all cells, edges, and vertices in the designated region. The result is a new netCDF mesh file that can be used to make a limited-area simulation as described in Section 8.2.

The limited-area Python program may be obtained from the MPAS-Atmosphere download page. Although the set of required Python packages may change over time, the program currently requires the `numpy` and `netCDF4` packages, in addition to other standard packages.

Chapter 5

Configuring Model Input and Output

The reading and writing of model fields in MPAS is handled by user-configurable *streams*. A stream represents a fixed set of model fields, together with dimensions and attributes, that are all written or read together to or from the same file or set of files. Each MPAS model core may define its own set of default streams that it typically uses for reading initial conditions, for writing and reading restart fields, and for writing additional model history fields. Besides these default streams, users may define new streams to, e.g., write certain diagnostic fields at a higher temporal frequency than the usual model history fields.

Streams are defined in XML configuration files that are created at build time for each model core. The name of this XML file is simply ‘streams.’ suffixed with the name of the core. For example, the streams for the *atmosphere* core are defined in a file named ‘streams.atmosphere’, and the streams for the *init_atmosphere* core are defined in a file named ‘streams.init_atmosphere’. An XML stream file may further reference other text files that contain lists of the model fields that are read or written in each of the streams defined in the XML stream file.

Changes to the XML stream configuration file will take effect the next time an MPAS core is run; there is no need to re-compile after making modifications to the XML files. As described in the next section, it is therefore possible, e.g., to change the interval at which a stream is written, the template for the filenames associated with a stream, or the set of fields that are written to a stream, without the need to re-compile any code.

Two classes of streams exist in MPAS: *immutable* streams and *mutable* streams. Immutable streams are those for which the set of fields that belong to the stream may not be modified at model run-time; however, it is possible to modify the interval at which the stream is read or written, the filename template describing the files containing the stream on disk, and several other parameters of the stream. In contrast, all aspects of mutable streams, including the set of fields that belong to the stream, may be modified at run-time. The motivation for the creation of two stream classes is the idea that an MPAS core may not function correctly if certain fields are not read in upon model start-up or written to restart files, and it is therefore not reasonable for users to modify this set of required fields at run-time. An MPAS core developer may choose to implement such streams as immutable streams. Since fields may not be added to an immutable stream at run-time, new immutable streams may not be defined at run-time, and the only type of new stream that may be defined at run-time is the mutable stream type.

5.1 XML stream configuration files

The XML stream configuration file for an MPAS core always has a parent XML *element* named **streams**, within which individual streams are defined:

```
<streams>
  ...
  ... one or more stream definitions ...
</streams>
```

```
</streams>
```

Immutable streams are defined with the `immutable_stream` element, and mutable streams are defined with the `stream` element:

```
<immutable_stream name="initial_conditions"
                  type="input"
                  filename_template="init.nc"
                  input_interval="initial_only"
                  />

<stream name="history"
       type="output"
       filename_template="output.$Y-$M-$D_$h.$m.$s.nc"
       output_interval="6:00:00" >
    ... model fields belonging to this stream ...
</stream>
```

As shown in the example stream definitions, above, both classes of stream have the following required attributes:

- `name` — A unique name used to refer to the stream.
- `type` — The type of stream, either "input", "output", "input;output", or "none". A stream may be both an input and an output stream (i.e., "input;output") if, for example, it is read once at model start-up to provide initial conditions and thereafter written periodically to provide model checkpoints. A stream may be defined as neither input nor output (i.e., "none") for the purposes of defining a set of fields for inclusion other streams. Note that, for immutable streams, the type attribute may not be changed at run-time.
- `filename_template` — The template for files that exist or will be created by the stream. The filename template may include any of the following variables, which are expanded based on the simulated time at which files are first created.
 - `$Y` — Year
 - `$M` — Month
 - `$D` — Day of the month
 - `$d` — Day of the year
 - `$h` — Hour
 - `$m` — Minute
 - `$s` — Second

A filename template may include either a relative or an absolute path, in which case MPAS will attempt to create any directories in the path that do not exist, subject to filesystem permissions.

- `input_interval` — For streams that have type "input" or "input;output", the interval, beginning at the model initial time, at which the stream will be read. Possible values include a time interval specification in the format "YYYY-MM-DD_hh:mm:ss"; the value "initial_only", which specifies that the stream is read only once at the model initial time; or the value "none", which specifies that the stream is not read during a model run.

- **output_interval** — For streams that have type "output" or "input;output", the interval, beginning at the model initial time, at which the stream will be written. Possible values include a time interval specification in the format "YYYY-MM-DD hh:mm:ss"; the value "initial_only", which specifies that the stream is written only once at the model initial time; or the value "none", which specifies that the stream is not written during a model run.

Finally, the set of fields that belong to a mutable stream may be specified with any combination of the following elements. Note that, for immutable streams, no fields are specified at run-time in the XML configuration file.

- **var** — Associates the specified variable with the stream. The variable may be any of those defined in an MPAS core's Registry.xml file, but may not include individual constituent arrays from a var_array.
- **var_array** — Associates all constituent variables in a var_array, defined in an MPAS core's Registry.xml file, with the stream.
- **var_struct** — Associates all variables in a var_struct, defined in an MPAS core's Registry.xml file, with the stream.
- **stream** — Associates all explicitly associated fields in the specified stream with the stream; streams are not recursively included.
- **file** — Associates all variables listed in the specified text file, with one field per line, with the stream.

5.2 Optional stream attributes

Besides the required attributes described in the preceding section, several additional, optional attributes may be added to the definition of a stream.

- **filename_interval** — The interval between the timestamps used in the construction of the names of files associated with a stream. Possible values include a time interval specification in the format "YYYY-MM-DD_hh:mm:ss"; the value "none", indicating that only one file containing all times is associated with the stream; the value "input_interval" that, for input type streams, indicates that each time to be read from the stream will come from a unique file; or the value "output_interval" that, for output type streams, indicates that each time to be written to the stream will go to a unique file whose name is based on the timestamp of the data being written. The default value is "input_interval" for input type streams and "output_interval" for output type streams. For streams of type "input;output", the default filename interval is "input_interval" if the input interval is an interval (i.e., not "initial_only"), or "output_interval" otherwise. Refer to Section 5.3.1 for an example of the use of the filename_interval attribute.
- **reference_time** — A time that is an integral number of filename intervals from the timestamp of any file associated with the stream. The default value is the start time of the model simulation. Refer to Section 5.3.3 for an example of the use of the reference_time attribute.
- **clobber_mode** — Specifies how a stream should handle attempts to write to a file that already exists. Possible values for the mode include:
 - "overwrite" — The stream is allowed to overwrite records in existing files and to append new records to existing files; records not explicitly written to are left untouched.
 - "truncate" or "replace_files" — The stream is allowed to overwrite existing files, which are first truncated to remove any existing records; this is equivalent to replacing any existing files with newly created files of the same name.
 - "append" — The stream is only allowed to append new records to existing files; existing records may not be overwritten.
 - "never_modify" — The stream is not allowed to modify existing files in any way.

The default clobber mode for streams is "never_modify". Refer to Section 5.3.2 for an example of the use of the clobber_mode attribute.

- **precision** — The precision with which real-valued fields will be written or read in a stream. Possible values include "single" for 4-byte real values, "double" for 8-byte real values, or "native", which specifies that real-valued fields will be written or read in whatever precision the MPAS core was compiled. The default value is "native". Refer to Section 5.3.1 for an example of the use of the precision attribute.
- **packages** — A list of packages attached to the stream. A stream will be active (i.e., read or written) only if at least one of the packages attached to it is active, or if no packages at all are attached. Package names are provided as a semi-colon-separated list. Note that packages may only be defined in an MPAS core's Registry.xml file at build time. By default, no packages are attached to a stream.
- **io_type** — The underlying library and file format that will be used to read or write a stream. Possible values include:
 - "pnetcdf" — Read/write the stream with classic large-file NetCDF files (CDF-2) using the ANL Parallel-NetCDF library.
 - "pnetcdf,cdf5" — Read/write the stream with large-variable files (CDF-5) using the ANL Parallel-NetCDF library.
 - "netcdf" — Read/write the stream with classic large-file NetCDF files (CDF-2) using the Unidata serial NetCDF library.

- "netcdf4" — Read/write the stream with HDF-5 files using the Unidata parallel NetCDF-4 library.

Note that the PIO library must have been built with support for the selected `io_type`. By default, all input and output streams are read and written using the "pnetcdf" option.

5.3 Stream definition examples

This section provides several example streams that make use of the optional stream attributes described in Section 5.2. All examples are of output streams, since it is more likely that a user will need to write additional fields than to read additional fields, which a model would need to be aware of; however, the concepts that are illustrated here translate directly to input streams as well.

5.3.1 Example: a single-precision output stream with one month of data per file

In this example, the optional attribute specification `filename_interval="01-00_00:00:00"` is added to force a new output file to be created for the stream every month. Note that the general format for time interval specifications is `YYYY-MM-DD hh:mm:ss`, where any leading terms can be omitted; in this case, the year part of the interval is omitted. To reduce the file size, the specification `precision="single"` is also added to force real-valued fields to be written as 4-byte floating-point values, rather than the default of 8 bytes.

```
<stream name="diagnostics"
    type="output"
    filename_template="diagnostics.$Y-$M.nc"
    filename_interval="01-00_00:00:00"
    precision="single"
    output_interval="6:00:00" >

    <var name="u10"/>
    <var name="v10"/>
    <var name="t2"/>
    <var name="q2"/>

</stream>
```

The only fields that will be written to this stream are the hypothetical 10-m diagnosed wind components, the 2-m temperature, and the 2-m specific humidity variables. Also, note that the filename template only includes the year and month from the model valid time; this can be problematic when the simulation starts in the middle of a month, and a solution for this problem is illustrated in the example of Section 5.3.3.

5.3.2 Example: appending records to existing output files

By default, streams will never modify existing files whose filenames match the name of a file that would otherwise be written during the course of a simulation. However, when restarting a simulation that is expected to add more records to existing output files, it can be useful to instruct the MPAS I/O system to append these records, thereby modifying existing files. This may be accomplished with the `clobber_mode` attribute.

```
<stream name="diagnostics"
    type="output"
    filename_template="diagnostics.$Y-$M.nc"
```

```

    filename_interval="01-00_00:00:00"
    precision="single"
    clobber_mode="append"
    output_interval="6:00:00" >

    <var name="u10"/>
    <var name="v10"/>
    <var name="t2"/>
    <var name="q2"/>

</stream>
```

In general, if MPAS were to attempt to write a record at a time that already existed in an output file, a clobber_mode of ‘append’ would not permit the write to take place, since this would modify existing data; in ‘append’ mode, only new records may be added. However, due to a peculiarity in the implementation of the ‘append’ clobber mode, it may be possible for an output file to contain duplicate times. This can happen when the first record that is appended to an existing file has a timestamp not matching any in the file, after which, any record that is written — regardless of whether its timestamp matches one already in the file — will be appended to the end of the file. This situation may arise, for example, when restarting a model simulation with a shorter output_interval than was used in the original model simulation with an MPAS core that does not write the first output time for restart runs.

5.3.3 Example: referencing filename intervals to a time other than the start time

The example stream of the previous sections creates a new file each month during the simulation, and the filenames contain only the year and month of the timestamp when the file was created. If a simulation begins at 00 UTC on the first day of a month, then each file in the diagnostic stream will contain only output times that fall within the month in the filename. However, if a simulation were to begin in the middle of a month — for example, the month of June, 2014 — the first diagnostics output file would have a filename of ‘diagnostics.2014-06.nc’, but rather than containing only output fields valid in June, it would contain all fields written between the middle of June and the middle of July, at which point one month of simulation would have elapsed, and a new output file, ‘diagnostics.2014-07.nc’, would be created.

In order to ensure that the file ‘diagnostics.2014-06.nc’ contained only data from June 2014, the reference_time attribute may be added such that the day, hour, minute, and second in the date and time represent the first day of the month at 00 UTC. In this example, the year and month of the reference time are not important, since the purpose of the reference time here is to describe to MPAS that the monthly filename interval begins (i.e., is referenced to) the first day of the month.

```

<stream name="diagnostics"
    type="output"
    filename_template="diagnostics.$Y-$M.nc"
    filename_interval="01-00_00:00:00"
    reference_time="2014-01-01_00:00:00"
    precision="single"
    clobber_mode="append"
    output_interval="6:00:00" >

    <var name="u10"/>
    <var name="v10"/>
    <var name="t2"/>
    <var name="q2"/>
```

```
</stream>
```

In general, the components of a timestamp, YYYY-MM-DD hh:mm:ss, that are less significant than (i.e., to the right of) those contained in a filename template are important in a `reference_time`. For example, with a `filename_template` that contained only the year, the month component of the `reference_time` would become important to identify the month of the year on which the yearly basis for filenames would begin.

Chapter 6

Physics Suites

Beginning with version 4.0, MPAS-Atmosphere introduces a new way of selecting the physics schemes to be used in a simulation. Rather than selecting individual parameterization schemes for different processes (e.g., convection, microphysics, etc.), the preferred method is for the user to select a *suite* of parameterization schemes that have been tested together. The selection of a physics suite is made via the namelist option `config_physics_suite` in the `&physics` namelist record. Each of the available suites are described in the sections that follow.

Although the preferred method for selecting the schemes in a simulation is via the choice of a suite, the need to enable or disable individual schemes, or to substitute alternative schemes for the suite default, is recognized. Accordingly, it is possible to override the choice of any individual parameterization scheme through the namelist options described in Appendix B. This is useful, e.g., to disable all parameterizations except for microphysics when running some idealized simulations. The details of selecting individual physics parameterizations are explained in Section 6.4.

6.1 Suite: mesoscale_reference

The default physics suite in MPAS-Atmosphere is the ‘mesoscale_reference’ suite, which contains the schemes listed in Table 6.1. This suite has been tested for mesoscale resolutions (> 10 km cell spacing), and is not appropriate for convective-scale simulations because the Tiedtke scheme will remove convective instability before resolved-scale motions (convective cells) can respond to it.

Table 6.1: The set of parameterization schemes used by the ‘mesoscale_reference’ physics suite.

| Parameterization | Scheme |
|--------------------------------|-----------------------|
| Convection | New Tiedtke |
| Microphysics | WSM6 |
| Land surface | Noah |
| Boundary layer | YSU |
| Surface layer | Revised Monin-Obukhov |
| Radiation, LW | RRTMG |
| Radiation, SW | RRTMG |
| Cloud fraction for radiation | Xu-Randall |
| Gravity wave drag by orography | YSU |

6.2 Suite: convection_permitting

The ‘convection_permitting’ physics suite is appropriate at spatial resolutions allowing for both explicitly resolved hydrostatic and nonhydrostatic motions. It has been tested for mesh spacings from several hundred kilometers down to 3 km in MPAS. The Grell-Freitas convection scheme transitions from a conventional parameterization of deep convection at hydrostatic scales (cell spacings of several tens of kilometers) to a parameterization of precipitating shallow convection at cell spacings less than 10 km. This is the recommended suite for any MPAS applications where convection-permitting meshes ($dx < 10$ km) are employed, including variable-resolution meshes spanning hydrostatic to nonhydrostatic resolutions.

Table 6.2: The set of parameterization schemes used by the ‘convection_permitting’ physics suite.

| Parameterization | Scheme |
|--------------------------------|------------------------------|
| Convection | Grell-Freitas |
| Microphysics | Thompson (non-aerosol aware) |
| Land surface | Noah |
| Boundary layer | MYNN |
| Surface layer | MYNN |
| Radiation, LW | RRTMG |
| Radiation, SW | RRTMG |
| Cloud fraction for radiation | Xu-Randall |
| Gravity wave drag by orography | YSU |

6.3 Suite: none

The only other recognized physics suite in MPAS-Atmosphere is the ‘none’ suite, which sets all physics parameterizations to ‘off’. This suite is primarily intended for use with idealized simulations. For example, the idealized supercell test case makes use of the ‘none’ suite, but with the microphysics scheme explicitly overridden:

```
config_physics_suite = 'none'  
config_microp_scheme = 'kessler'
```

6.4 Selecting individual physics parameterizations

Selecting or disabling an individual physics parameterization may be accomplished by setting the appropriate namelist variable to one of its possible options; possible options for individual parameterizations, along with details of those options, are given in Table 6.3. Note that all parameterization options may be set to ‘off’ to disable the parameterization of the associated process.

Table 6.3: Possible options for individual physics parameterizations. Namelist variables should be added to the &physics namelist record.

| Parameterization | Namelist variable | Possible options | Details |
|--------------------------------|---------------------------------------|------------------------------------|---|
| Convection | <code>config_convection_scheme</code> | <code>cu_tiedtke</code> | Tiedtke (WRF 3.8.1) |
| | | <code>cu_ntiedtke</code> | New Tiedtke (WRF 4.5) |
| | | <code>cu_grell_freitas</code> | Modified version of scale-aware Grell-Freitas (WRF 3.6.1) |
| | | <code>cu_kain_fritsch</code> | Kain-Fritsch (WRF 3.2.1) |
| Microphysics | <code>config_microp_scheme</code> | <code>mp_wsm6</code> | WSM 6-class (WRF 4.5) |
| | | <code>mp_thompson</code> | Thompson non-aerosol aware (WRF 3.8.1) |
| | | <code>mp_thompson_aerosols</code> | Thompson aerosol-aware (WRF 4.1.4) |
| | | <code>mp_kessler</code> | Kessler |
| Land surface | <code>config_lsm_scheme</code> | <code>sf_noah</code> | Noah (WRF 4.5) |
| | | <code>sf_noahmp</code> | Noah-MP 5.0.1 |
| Boundary layer | <code>config_pbl_scheme</code> | <code>bl_ysu</code> | YSU (WRF 4.5) |
| | | <code>bl_myNN</code> | MYNN (WRF 3.6.1) |
| Surface layer | <code>config_sfclayer_scheme</code> | <code>sf_monin_obukhov</code> | Monin-Obukhov (WRF 4.5) |
| | | <code>sf_monin_obukhov_rev</code> | Revised Monin-Obukhov (WRF 4.5) |
| | | <code>sf_myNN</code> | MYNN (WRF 3.6.1) |
| Radiation, LW | <code>config_radt_lw_scheme</code> | <code>rrtmg_lw</code> | RRTMG (WRF 3.8.1) |
| | | <code>cam_lw</code> | CAM (WRF 3.3.1) |
| Radiation, SW | <code>config_radt_sw_scheme</code> | <code>rrtmg_sw</code> | RRTMG (WRF 3.8.1) |
| | | <code>cam_sw</code> | CAM (WRF 3.3.1) |
| Cloud fraction for radiation | <code>config_radt_cld_scheme</code> | <code>cld_fraction</code> | Xu and Randall (1996) |
| | | <code>cld_incidence</code> | 0/1 cloud fraction depending on $q_c + q_i$ |
| | | <code>cld_fraction_thompson</code> | Thompson cloud fraction scheme |
| Gravity wave drag by orography | <code>config_gwdo_scheme</code> | <code>bl_ysu_gwdo</code> | YSU (WRF 4.5) |
| | | <code>bl_ugwp_gwdo</code> | NOAA/GSL orographic gravity wave drag (see also <code>config_ngw_scheme</code>) |

Chapter 7

Running the MPAS Non-hydrostatic Atmosphere Model

Given a CVT mesh, prepared using steps from Chapter 4 as appropriate, this chapter describes the two main steps to running the MPAS-Atmosphere model: creating initial conditions and running the model itself. This chapter makes use of two MPAS cores, `init_atmosphere` and `atmosphere`, which are, respectively, used for initializing and running the non-hydrostatic atmospheric model. Sections 7.1 and 7.2 of this chapter describe the creation of idealized and real-data initial condition files using the `init_atmosphere` core. Section 7.3 describes the basic procedure of running the model itself.

Each section of this chapter follows a familiar pattern of compiling and executing MPAS model components, albeit using different cores depending on its intended use. The compilation will create either an initialization or a model executable, which are named, respectively, `init_atmosphere_model` and `atmosphere_model`. In general, an executable is run with `mpiexec` or `mpirun`, for example:

```
> mpiexec -n 8 atmosphere_model
```

where 8 is the number of MPI tasks to be used. In any case where $n > 1$, there must exist a corresponding graph decomposition file, e.g., `graph.info.part.8`. For more on graph decomposition, see Section 4.1.

7.1 Creating idealized ICs

There are several idealized test cases supported within the `init_atmosphere` model initialization core:

- 1 — Jablonowski and Williamson baroclinic wave, no initial perturbation ¹
- 2 — Jablonowski and Williamson baroclinic wave, with initial perturbation
- 3 — Jablonowski and Williamson baroclinic wave, with normal-mode perturbation
- 4 — squall line
- 5 — super-cell
- 6 — mountain wave

Creating idealized initial conditions is fairly straightforward, as no external data are required and the starting date/time is irrelevant to building the initial conditions file (hereafter referred to as `init.nc`) that will be used to run the model.

The following steps summarize the creation of `init.nc`:

- Include a `grid.nc` file, which contains the CVT mesh, in the working directory
- If running with more than one MPI task, include a `graph.info.part.*` file in the working directory (Section 4.1)

¹Jablonowski, C. and D.L. Williamson, 2006, A baroclinic instability test case for atmospheric model dynamical cores, *QJRMS*, 132, 2943-2975. doi:10.1256/qj.06.12.

- Compile MPAS with the `init_atmosphere` core specified (Section 3.3)
- Edit the `namelist.init_atmosphere` configuration file (described below)
- Edit the `streams.init_atmosphere` I/O configuration file (described below)
- Run `init_atmosphere_model` to create the initial condition file, `init.nc`

When the `init_atmosphere_model` executable is built, a default namelist, `namelist.init_atmosphere`, will have been created. A number of the namelist parameters found in `namelist.init_atmosphere` are irrelevant to creating idealized conditions and can be removed or ignored. The following table outlines the namelist parameters that are required; comments are given on the right for certain key parameters, and formal explanations for all namelist parameters can be found in Appendix A.

| | | |
|--|---|---|
| <code>&nhyd_model</code> | | |
| <code>config_init_case = 2</code> | = | a number between 1 and 6 corresponding to the cases listed at the beginning of this section |
| <code>config_start_time = '0000-01-01_00:00:00'</code> | = | the starting time for the simulation |
| <code>config_theta_adv_order = 3</code> | = | advection order for theta |
| <code>config_coef_3rd_order = 0.25</code> | | |
| <code>/</code> | | |
| <code>&dimensions</code> | | |
| <code>config_nvertlevels = 26</code> | = | the number of vertical levels to be used in the model |
| <code>/</code> | | |
| <code>&decomposition</code> | | |
| <code>config_block_decomp_file_prefix</code> | = | if running in parallel, needs to match the grid decomposition file prefix |
| <code>'graph.info.part.'</code> | | |
| <code>/</code> | | |

After editing the `namelist.init_atmosphere` namelist file, the name of the input grid file, as well as the name of the initial condition file to be created, must be set in the XML I/O configuration file, `streams.init_atmosphere`. For a detailed description of the format of the XML I/O configuration file, refer to Chapter 5. Specifically, the `filename_template` attribute must be set to the name of the grid file in the "input" stream definition, and the `filename_template` attribute must be set to name of the initial condition file to be created in the "output" stream definition.

7.2 Creating real-data ICs

The process of generating real-data initial conditions is similar to that of the idealized case described in the previous section, but is more involved as it requires interpolation of static geographic data (e.g., topography, land cover, soil category, etc.), surface fields such as soil temperature and SST, and the atmospheric initial conditions valid at a specific date and time. The static datasets are the same as those used by the WRF model, and the surface fields and atmospheric initial conditions can be obtained from, e.g., NCEP's GFS data using the WRF Pre-processing System (WPS).

Creating real-data initial conditions requires a single compilation of the `init_atmosphere` core, but the actual generation of the IC files will take place using two separate runs of the `init_atmosphere_model` program, where each of these two runs is described individually in the following sub-sections. While it is possible to condense the two real-data initialization steps into a single run, running each step separately will both improve clarity and, as will become apparent, save a significant amount of time when generating subsequent initial conditions, that is, when making initial conditions using the same mesh but different starting times.

The first step, described in Section 7.2.1, is the interpolation of static fields onto the mesh to create a `static.nc` file. This step cannot be run in parallel and generally takes considerable time to complete;

however, the fields being static, this step need only be run once for a particular mesh, regardless of the number of initial condition files that are ultimately created from the `static.nc` output file. Section 7.2.2 then describes the processing of the atmospheric initial conditions beginning with the `static.nc` file created in Section 7.2.1. Naturally, each of the initialization runs described in the following sub-sections will make use of a `namelist.init_atmosphere` namelist file, and as was the case for idealized initial conditions, the default `namelist.init_atmosphere` file in the MPAS directory may be used as a starting point. Not every variable in this namelist is needed for any particular step, and therefore each section will elaborate only on the namelist variables that are immediately relevant.

7.2.1 Static fields

The generation of a `static.nc` file requires a set of static geographic data. A suitable dataset can be obtained from the WRF model's download page

http://www2.mmm.ucar.edu/wrf/users/download/get_source.html. These static data files should be downloaded to a directory, which will be specified the `namelist.init_atmosphere` file (described below) prior to running this interpolation step. The result of this run will be the creation of a netCDF file (`static.nc`), which is used in Section 7.2.2 to create dynamic initial conditions. Note that `static.nc` can be generated once and then used repeatedly to generate initial condition files for different start times.

The following steps summarize the creation of `static.nc`:

- Download geographic data from the WRF download page (described above)
- Compile MPAS with the `init_atmosphere` core specified (Section 3.3)
- Include a `grid.nc` file in the working directory
- Edit the `namelist.init_atmosphere` configuration file (described below)
- Edit the `streams.init_atmosphere` I/O configuration file (described below)
- Run `init_atmosphere_model` with *only one MPI task specified* to create `static.nc`

Note that it is critical for this step that the initialization core is run serially; afterward, however, the steps described in 7.2.2 may be run with more than one MPI task.

| | |
|--|---|
| <pre>&nhyd_model config_init_case = 7 / &dimensions config_nvertlevels = 1 config_nsoillevels = 1 config_nfglevels = 1 config_nfgoillevels = 1 / &data_sources config_geog_data_path = '/path/to/WPS_GEOG/' config_landuse_data = 'MODI- FIED_IGBP_MODIS_NOAH' config_topo_data = 'GMTED2010' config_vegfrac_data = 'MODIS' config_albedo_data = 'MODIS' config_maxsnowalbedo_data = 'MODIS' config_supersample_factor = 1</pre> | <p>must be 7, the real-data initialization case</p> <p>the following dimensions should be set to 1 now, and their values will become significant in §7.2.2</p> <p>= absolute path to static files obtained from the WRF download page</p> <p>= land use dataset selection</p> <p>= terrain dataset selection</p> <p>= monthly vegetation fraction dataset selection</p> <p>= monthly albedo dataset selection</p> <p>= maximum snow albedo dataset selection</p> <p>= MODIS supersampling factor, generally only needed for meshes with grid distance < 6 km</p> |
|--|---|

```

/
&preproc_stages
config_static_interp = true
config_native_gwd_static = true
config_vertical_grid = false
config_met_interp = false
config_input_sst = false
config_frac_seaice = false
/

```

only the static_interp and native_gwd_static stages should be enabled

After editing the `namelist.init_atmosphere` namelist file, the name of the input CVT grid file, as well as the name of the static file to be created, must be set in the XML I/O configuration file, `streams.init_atmosphere`. For a detailed description of the format of the XML I/O configuration file, refer to Chapter 5. Specifically, the `filename_template` attribute must be set to the name of the SCVT grid file in the "input" stream definition, and the `filename_template` attribute must be set to name of the static file to be created in the "output" stream definition.

7.2.2 Vertical grid generation and initial field interpolation

The second step in creating a real-data initial conditions file (`init.nc`) is to generate a vertical grid, the parameters of which will be specified in the `namelist.init_atmosphere` file, and to obtain an initial conditions dataset and interpolate it onto the model grid. As stated previously, while initial conditions could ultimately be obtained from many different data sources, here we assume the use of WPS "intermediate" data files obtained from GFS data using the WPS ungrb program. Detailed instructions for building and running the WPS, and how to generate intermediate data files from GFS data, can be found in Chapter 3 of the WRF user guide:

http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/v4.1/users_guide_chap3.html.

The following steps summarize the creation of `init.nc`:

- Include a WPS intermediate data file in the working directory
- Include the `static.nc` file in the working directory (Section 7.2.1)
- If running in parallel, include a `graph.info.part.*` file in the working directory (Section 4.1)
- Edit the `namelist.init_atmosphere` configuration file (described below)
- Edit the `streams.init_atmosphere` I/O configuration file (described below)
- Run `init_atmosphere_model` to create `init.nc`

```

&nhyd_model
config_init_case = 7
config_start_time = '2010-10-23_00:00:00'
config_theta_adv_order = 3
config_coef_3rd_order = 0.25
/
&dimensions
config_nvertlevels = 55
config_nsoillevels = 4
config_nfglevels = 38
config_nfgsoillevels = 4
/

```

must be 7
time to process first-guess data
advection order for theta

number of vertical levels to be used in MPAS
number of soil layers to be used in MPAS
number of vertical levels in intermediate file
number of soil layers in intermediate file

| | |
|---|---|
| <pre> &data_sources config_met_prefix = 'FILE' config_use_specchumd = true / &vertical_grid config_ztop = 30000.0 config_nsmterrain = 1 config_smooth_surfaces = true config_blend_boundary_terrain = false / &preproc_stages config_static_interp = false config_native_gwd_static = false config_vertical_grid = true config_met_interp = true config_input_sst = false config_frac_seaice = true / &decomposition config_block_decomp_file_prefix = 'graph.info.part.' / </pre> | the prefix of the intermediate file to be used for initial conditions if available, use specific humidity rather than relative humidity model top height (m) number of smoothing passes for terrain whether to smooth zeta surfaces whether to blend terrain along domain boundaries; only for regional simulations as described in Section 8.2 only these three stages should be enabled = if running in parallel, needs to match the grid decomposition file prefix |
|---|---|

After editing the `namelist.init_atmosphere` namelist file, the name of the static file, as well as the name of the initial condition file to be created, must be set in the XML I/O configuration file, `streams.init_atmosphere`. Specifically, the `filename_template` attribute must be set to the name of the static file in the "input" stream definition, and the `filename_template` attribute must be set to name of the initial condition file to be created in the "output" stream definition.

7.3 Running the model

Having generated the model initial conditions, `init.nc`, as described in either 7.1 or 7.2, we have completed the prerequisites to run the model. The only step remaining before running the model itself is the configuration of `namelist.atmosphere`. When the *atmosphere* core is built, a default `namelist.atmosphere` namelist file will be automatically generated; this namelist can serve as a starting point for any modifications made following the steps below. This section will discuss both running the model from a cold start and restarting the model from some point in a previous run.

The following steps summarize running the model:

- Include an initial condition netCDF file (e.g., `init.nc`) in the working directory(Section 7.1, Section 7.2)
- (OPTIONAL) If the SST and sea-ice fields are to be periodically updated, include a surface netCDF file (e.g., `surface.nc`) in the working directory (Section 8.1)
- (OPTIONAL) If running a regional simulation, include LBC netCDF files (e.g., `lbc.YYYY-MM-DD_HH.mm.ss.nc`) in the working directory (Section 8.2)

- If running in parallel, include a graph decomposition file in the working directory (Section 4.1)
- If the MPAS directory has not been cleaned since running initialization, run `make clean` with the `atmosphere` core specified
- Compile MPAS with the `atmosphere` core specified (Section 3.3)
- Edit the default `namelist.atmosphere` configuration file (described below)
- Edit the `streams.atmosphere` I/O configuration file (described below)
- Run the `atmosphere_model` executable

Below is a list of variables in `namelist.atmosphere` that pertain to model timestepping, explicit horizontal diffusion, and model restarts. A number of namelist variables are not listed here (specifications for dynamical core configuration, physics parameters, etc.) and Appendix B should be consulted for the purpose and acceptable values of these parameters.

| | | |
|---|---|--|
| <code>&nhyd_model</code> | | |
| <code>config_dt</code> = 720.0 | | the model timestep; an appropriate value must be chosen relative to the grid cell spacing |
| <code>config_start_time</code> = '2010-10-23_00:00:00' | | the model start time corresponding to <code>init.nc</code> |
| <code>config_run_duration</code> = '5_00:00:00' | | the duration of the model run; for format rules, see Appendix B |
| <code>config_len_disp</code> = 120000.0 | | the smallest cell-to-cell distance in the mesh, used for computing a dissipation length scale |
| / | | |
| <code>&limited_area</code> | | |
| <code>config_apply_lbcs</code> = false | | must be set to true only if running a regional simulation, as in Section 8.2 |
| / | | |
| <code>&decomposition</code> | | |
| <code>config_block_decomp_file_prefix</code> = 'graph.info.part.' | = | if running in parallel, must match the prefix of the graph decomposition file |
| / | | |
| <code>&restart</code> | | |
| <code>config_do_restart</code> = false | | if true, will select the appropriate <code>restart.nc</code> file generated from a previous run |
| / | | |
| <code>&physics</code> | | |
| <code>config_sst_update</code> = true | | if updating sea-ice and SST with an <code>surface.nc</code> file, set to <code>true</code> , and edit the "surface" stream in the <code>streams.atmosphere</code> file accordingly |
| <code>config_physics_suite</code> = 'mesoscale_reference' | | |
| / | | |

When running the model from a cold start, `config_start_time` should match the time that was used when creating `init.nc`.

Configuration of model input and output is accomplished by editing the `streams.atmosphere` file. The following streams exist by default in the atmosphere core:

| | |
|-------|---|
| input | the stream used to read model initial conditions for cold-start simulations |
|-------|---|

| | |
|-------------|--|
| restart | the stream used to periodically write restart files during model integration, and to read initial conditions when performing a restart model run |
| output | the stream responsible for writing model prognostic and diagnostic fields to history files |
| diagnostics | the stream responsible for writing (mostly) 2-d diagnostic fields, typically at higher temporal frequency than the history files |
| surface | the stream used to read periodic updates of sea-ice and SST from a surface update file created as described in Section 8.1 |
| iau | the stream used to read analysis increments for the Incremental Analysis Update (IAU) scheme |
| lbc_in | the stream used to read lateral boundary updates for limited-area simulations, described in Section 8.2 |

For more information on the options available in the XML I/O configuration file, users are referred to Chapter 5.

During the course of a model run, restart files are created at an interval specified by the `output_interval` attribute in the definition of the "restart" stream. Running the model from a restart file is similar to running the model from `init.nc`. The required changes are that `config_do_restart` must be set to `true` and `config_start_time` must correspond to a restart file existing in the working directory.

Chapter 8

Model Options

Beyond the basic process of running a global simulation with standard output files outlined in Chapter 7, the MPAS-Atmosphere model provides several options that can be described in terms of variations on the basic simulation workflow. In the sections that follow, major model options are described in terms of the deviation from basic global simulation process.

8.1 Periodic SST and Sea-ice Updates

The stand-alone MPAS-Atmosphere model is not coupled to fully prognostic ocean or sea-ice models, and accordingly, the model SST and sea-ice fraction fields will in general not change over the course of a simulation. For simulations shorter than a few days, invariant SST and sea-ice fraction fields will generally not be problematic. However, for longer model simulations, it is generally recommended to periodically update the SST and sea-ice fields from an external file.

The surface data to be used for periodic SST and sea-ice updates could originate from any number of sources, though the most straightforward way to obtain a dataset in a usable format is to process GRIB data (e.g., GFS GRIB data) with the *ungrb* program of the WRF model's pre-processing system(WPS). Detailed instructions for building and running the WPS, and the process of generating intermediate data files from GFS data, can be found in Chapter 3 of the WRF User Guide: http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/v4.1/users_guide_chap3.html.

The following steps summarize the generation of an SST and sea-ice update file, `surface.nc`, using the `init_atmosphere_model` program:

- Include surface data intermediate files in the working directory
- Include a `static.nc` file in the working directory (Section 7.2.1)
- If running in parallel, include a `graph.info.part.*` in the working directory (Section 4.1)
- Edit the `namelist.init_atmosphere` configuration file (see below)
- Edit the `streams.init_atmosphere` I/O configuration file (described below)
- Run `init_atmosphere_model` to create `surface.nc`

```
&nhyd_model  
config_init_case = 8  
config_start_time = '2010-10-23_00:00:00'  
config_stop_time = '2010-10-30_00:00:00'  
/  
&data_sources
```

must be 8, the surface field initialization case
time to begin processing surface data
time to end processing surface data

| | |
|---|---|
| <code>config_sfc_prefix = 'SST'</code> | the prefix of the intermediate data files containing SST and sea-ice |
| <code>config_fg_interval = 86400</code> | interval between intermediate files to use for SST and sea-ice |
| / | |
| | |
| <code>&preproc_stages</code> | |
| <code>config_static_interp = false</code> | only the <code>input_sst</code> and <code>frac_seaice</code> stages should be enabled |
| <code>config_native_gwd_static = false</code> | |
| <code>config_vertical_grid = false</code> | |
| <code>config_met_interp = false</code> | |
| <code>config_input_sst = true</code> | |
| <code>config_frac_seaice = true</code> | |
| / | |
| | |
| <code>&decomposition</code> | |
| <code>config_block_decomp_file_prefix = 'graph.info.part.'</code> | = if running in parallel, needs to match the grid decomposition file prefix |
| / | |

After editing the `namelist.init_atmosphere` namelist file, the name of the static file, as well as the name of the surface update file to be created, must be set in the XML I/O configuration file, `streams.init_atmosphere`. Specifically, the `filename_template` attribute must be set to the name of the static file in the “`input`” stream definition, and the `filename_template` attribute must be set to name of the surface update file to be created in the “`surface`” stream definition. *Also, for the “`surface`” stream, ensure that the “`output_interval`” attribute is set to the interval at which the surface intermediate files are provided.*

8.2 Regional Simulation

New in MPAS v7.0 is the capability to run simulations over regional domains on the surface of the sphere. Setting up and running a limited-area simulation requires as a starting point a limited-area SCVT mesh, described in section 4.3. Given a limited-area mesh, the key differences from a global simulation are:

- the blending of the MPAS terrain field with the “first-guess” terrain data along the boundaries of the limited-area domain;
- the generation of a set of files containing lateral boundary conditions (LBCs); and
- the application of LBCs during the model integration.

The first of these differences – the blending of terrain data – takes place when generating the limited-area initial conditions. Limited-area initial conditions are prepared as in 7.2.2, except that the `config_blend_bdy_terrain` option should be set to `true` in the `namelist.init_atmosphere` file. This option instructs the `init_atmosphere_model` program to perform averaging of the model terrain field from the `static.nc` file with the terrain field from the atmospheric initial conditions dataset along the lateral boundaries of the mesh.

The second difference – the generation of LBC files – requires running the `init_atmosphere_model` program one additional time, with namelist options set as described below.

| | |
|--|-----------------------------------|
| <code>&nhyd_model</code> | |
| <code>config_init_case = 9</code> | the LBCs processing case |
| <code>config_start_time = '2010-10-23_00:00:00'</code> | time to begin processing LBC data |
| <code>config_stop_time = '2010-10-30_00:00:00'</code> | time to end processing LBC data |
| / | |

| | |
|---|---|
| &dimensions | |
| config_nlevels = 38 | number of vertical levels in intermediate file |
| / | |
| &data_sources | |
| config_met_prefix = 'GFS' | |
| config_fg_interval = 10800 | the prefix of intermediate data files to be used for LBCs |
| / | interval between intermediate files |
| &decomposition | |
| config_block_decomp_file_prefix 'graph.info.part.' | = if running in parallel, needs to match the grid decomposition file prefix |
| / | |

When running the LBC processing case, the `output_interval` for the “lbc” stream in the `streams.init_atmosphere` file must be set to match the value of `config_fg_interval` in the `namelist.init_atmosphere` file. Additionally, the file to be read by the “input” stream must contain vertical grid information; typically, the model initial-conditions file can be used as the source for the “input” stream. The end result of the LBC processing case should be a set of netCDF files containing LBCs for the model integration.

The final difference – application of LBCs during the model integration – simply requires setting `config_apply_lbccs` to `true` in the model’s `namelist.atmosphere` file, as well as setting the `input_interval` for the “lbc_in” stream in the `streams.atmosphere` file to match the interval at which the LBC netCDF files were produced.

8.3 Separate Stream for Invariant Fields

By default, the MPAS-Atmosphere model reads time-invariant fields (e.g., `latCell`, `lonCell`, `areaCell`, `zgrid`, `zz`, etc.) from the “input” and “restart” streams (for cold-start and restart runs, respectively), and it writes time-invariant fields to the “restart” stream. In the case of large ensembles, the time-invariant fields that are replicated in the restart files for all ensemble members can account for a substantial amount of storage. In principle, since these time-invariant fields do not change in time or across ensemble members, only one copy of these fields needs to be stored.

MPAS-Atmosphere v8.1.0 introduces a capability to omit time-invariant fields from model restart files. When the model restarts, a new “invariant” stream may be used to read time-invariant fields from a separate file, and many ensemble members can share this file.

In order to make use of the new “invariant” stream, several changes to the standard MPAS-Atmosphere workflow are needed.

8.3.1 Preparing an Invariant File

Through the use of the `init_atmosphere_model` program, a file containing all required time-invariant fields must be prepared. Of course, since the model initial conditions (typically referred to as the `init.nc` file) file contains time-invariant fields, the initial conditions file from any ensemble member may be used.

If a file containing purely time-invariant fields is desired, the output from the `init_atmosphere_model` program after the following pre-processing stages have been run will suffice:

- `config_static_interp = true`
- `config_native_gwd_static = true`
- `config_vertical_grid = true`

Note that these pre-processing stages do not need to be run all at once. It is possible, for example, to first produce a `static.nc` file using the first two of these pre-processing stages, and to then produce an invariant file (e.g., `invariant.nc`) by running the vertical grid generation stage using the `static.nc` file as input.

8.3.2 Activating the Invariant Stream

When running the model itself (`atmosphere_model`), the use of the new invariant stream may be activated by simply defining the “invariant” immutable stream in the `streams.atmosphere` file as follows:

```
<immutable_stream name="invariant"
                  type="input"
                  filename_template="invariant.nc"
                  input_interval="initial_only" />
```

In the definition of the “invariant” stream, the `filename_template` attribute should be set to the actual name of the invariant file.

By the existence of the “invariant” stream in the `streams.atmosphere` file, the model will omit all time-invariant fields from any restart files that are written. When the model restarts, all time-invariant fields will be read from the “invariant” stream rather than from the “restart” stream.

Chapter 9

Visualization

Since the MPAS input and output files are in netCDF format, a wide variety of software tools may be used to manipulate and visualize fields in these files. As a starting point, several NCL¹ scripts for making the basic types of plots illustrated in Figure 9.1 are provided through the MPAS-Atmosphere download page. Each of these scripts reads the name of the file from which fields should be plotted from the environment variable `FNAME`, and all but the mesh-plotting script read the time frame to be plotted from the environment variable `T`. To plot a field from the first frame (indexed from 0) of the file `output.2010-10-23_00:00:00.nc`, for example, one would set the following environment variables

```
> setenv FNAME output.2010-10-23_00:00:00.nc
> setenv T 0
```

before running one of the scripts. In general, the specific field to be plotted from the netCDF file must be set within a script before running that script.

9.1 Meshes

A plot showing just an MPAS SCVT mesh can be produced using the `atm_mesh.ncl` script, as in Figure 9.1(a). This script reads a subset of the mesh description fields in Appendix C and uses this information to draw the SCVT mesh over a color-filled map background. Parameters in the script can be used to control the type of map projection (e.g., orthographic, cylindrical equidistant, etc.), the colors used to fill land and water points, and the widths of lines used for the Voronoi cells.

9.2 Horizontal contour plots

Contour plots of horizontal fields can be produced with the `atm_contours.ncl` script, as in Figure 9.1(b). The particular field to be plotted is set in the script and can in principle be drawn on any horizontal surface (e.g., a constant pressure surface, a constant height surface, sea-level, etc.) if suitable vertical interpolation code is added to the script. Not shown in the figure are horizontal wind vectors, which can also be added to the plot using example code provided in the script.

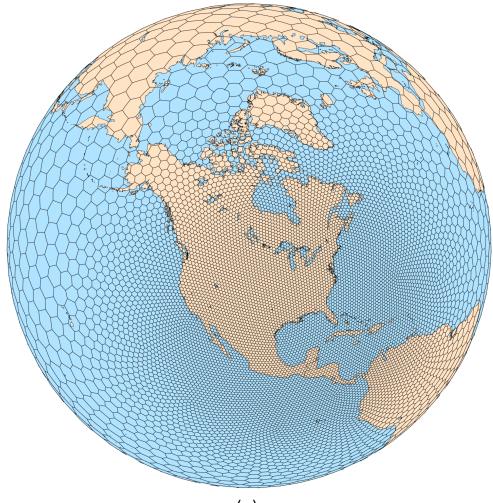
9.3 Horizontal cell-filled plots

For visualizing horizontal fields on their native SCVT grid, the `atm_cells.ncl` script may be used to produce horizontal cell-filled plots, as in Figure 9.1(c). This script draws each MPAS grid cell as a polygon colored according to the value of the field in that cell; the color scale is automatically chosen based on the range of the field and the default NCL color table, though other color tables can be selected instead.

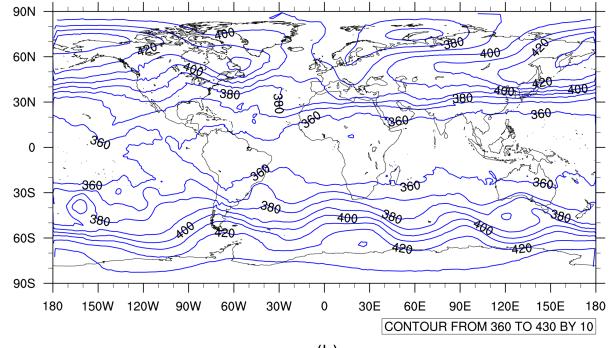
¹NCAR Command Language; <http://ncl.ucar.edu>

9.4 Vertical cross-sections

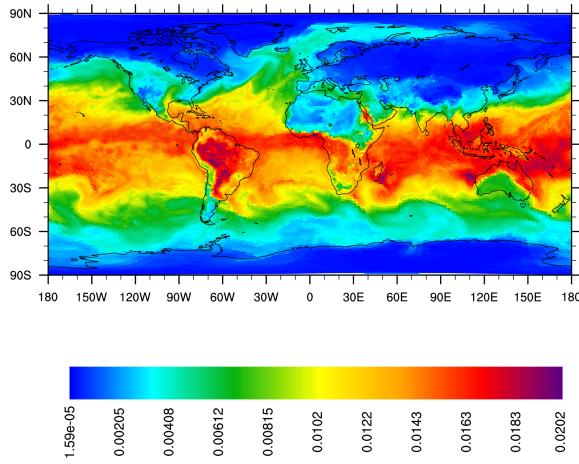
Vertical cross-sections of fields can be created using the `atm_xsec.ncl` script, as in Figure 9.1(d). Before running this script, a starting point and an ending point for the cross section must be given as latitude-longitude pairs near the top of the script, and the number of points along the cross section should be specified. The script evenly distributes the specified number of points along the shortest great-circle arc from the starting point to the ending point, and for each point, the script uses values from the grid cell containing that point (i.e., a nearest-neighbor interpolation to the horizontal cross-section points is performed); no vertical interpolation is performed, and the thicknesses and vertical heights of cells are all drawn according to the MPAS vertical grid.



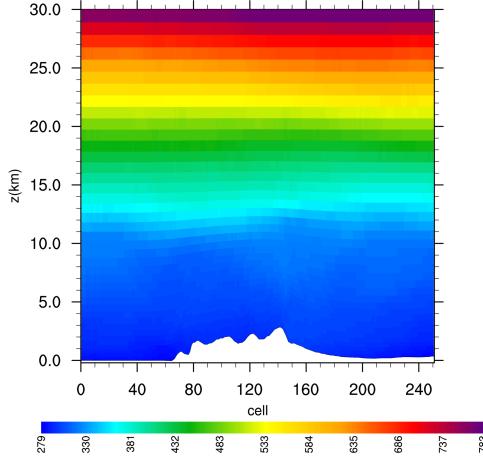
(a)



(b)



(c)



(d)

Figure 9.1: Various plot types that can be produced from MPAS input or output files using example scripts provided with the MPAS code. (a) A plot of an MPAS SCVT mesh against a filled map background. (b) A simple horizontal contour plot. (c) A cell-filled horizontal plot, with individual cells of the mesh drawn as polygons colored according to the field value. (d) A vertical cross-section in height, with areas below the terrain left unshaded.

Appendix A

Initialization Namelist Options

This chapter summarizes the complete set of namelist options available when running the MPAS non-hydrostatic atmosphere initialization core. The applicability of certain options depends on the type of initial conditions to be created — idealized or ‘real-data’ — and such applicability is identified in the description when it exists.

Date-time strings throughout all MPAS namelists assume a common format. Specifically, time intervals are of the form ‘[DDD]HH:MM:SS[.sss]’, where DDD is an integer number of days with any number of digits, HH is a two-digit hour value, MM is a two-digit minute value, SS is a two-digit second value, and sss are fractions of a second with any number of digits; any part of the time interval format in square brackets ([]) may be omitted, and if days are omitted, HH may be either a one- or two-digit hour specification. Time instants (e.g., start time or end time) are of the form ‘YYYY-MM-DD[_HH:MM:SS[.sss]]’, where YYYY is an integer year with any number of digits, MM is a two-digit month value, DD is a two-digit day value, and HH:MM:SS.sss is a time with the same format as in a time interval specification. For both time instants and time intervals, a value of ‘none’ represents ‘no value’.

A.1 nhyd_model

config_init_case (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <p><i>Type of initial conditions to create:</i></p> <p>1 = Jablonowski & Williamson barolinic wave (no initial perturbation), 2 = Jablonowski & Williamson barolinic wave (with initial perturbation), 3 = Jablonowski & Williamson barolinic wave (with normal-mode perturbation), 4 = squall line, 5 = super-cell, 6 = mountain wave, 7 = real-data initial conditions from, e.g., GFS, 8 = surface field (SST, sea-ice) update file for use with real-data simulations 9 = lateral boundary conditions update file for use with real-data simulations 13 = CAM-MPAS 3-d grid with specified topography and zeta levels</p> |
| Possible Values | 1 – 9, or 13 (default: 7) |

config_calendar_type (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Simulation calendar type (hidden by default)</i> |
| Possible Values | ‘gregorian’, ‘gregorian_noleap’ (<i>default: gregorian</i>) |

config_start_time (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Time to begin processing first-guess data (cases 7, 8, and 9 only)</i> |
| Possible Values | ‘YYYY-MM-DD hh:mm:ss’ (<i>default: 2010-10-23_00:00:00</i>) |

config_stop_time (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Time to end processing first-guess data (cases 8 and 9 only)</i> |
| Possible Values | ‘YYYY-MM-DD hh:mm:ss’ (<i>default: 2010-10-23_00:00:00</i>) |

config_theta_adv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Horizontal advection order for theta</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_coef_3rd_order (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Upwinding coefficient in the 3rd order advection scheme</i> |
| Possible Values | $0 \leq \text{config_coef_3rd_order} \leq 1$ (<i>default: 0.25</i>) |

config_num_halos (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Number of halo layers for fields (hidden by default)</i> |
| Possible Values | Integer values, typically 2 or 3; DO NOT CHANGE (<i>default: 2</i>) |

config_interface_projection (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>projecting layer values to the interface, linear vertical interpolation or integral</i> |
| Possible Values | 'linear_interpolation' or 'layer_integral' (layer average value) (<i>default: linear_interpolation</i>) |

A.2 dimensions

config_nvertlevels (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The number of vertical levels to be used in the model</i> |
| Possible Values | Positive integer values (<i>default: 55</i>) |

config_nsoillevels (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The number of vertical soil levels needed by LSM in the model (case 7 only)</i> |
| Possible Values | Positive integer values (<i>default: 4</i>) |

config_nfglevels (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>The number of atmospheric levels (including surface and sea-level) in the first-guess dataset (cases 7 and 9 only)</i> |
| Possible Values | Positive integer values (<i>default: 38</i>) |

config_nfgsoillevels (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The number of vertical soil levels in the first-guess dataset (case 7 only)</i> |
| Possible Values | Positive integer values (<i>default: 4</i>) |

config_gocartlevels (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>The number of vertical gocart levels in the climatological gocart data set</i> |
| Possible Values | Positive integer values (<i>default: 30</i>) |

config_months (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>The number of months in a year (hidden by default)</i> |
| Possible Values | Positive integer values (<i>default: 12</i>) |

A.3 data_sources**config_geog_data_path** (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Path to the WPS static data files (case 7 only)</i> |
| Possible Values | Any valid path (<i>default: /glade/work/wrfhelp/WPS_GEOG/</i>) |

config_met_prefix (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Filename prefix of ungridded intermediate file to use for initial conditions (cases 7 and 9 only)</i> |
| Possible Values | Any alpha-numeric string (<i>default: CFSR</i>) |

config_sfc_prefix (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Filename prefix of ungridded intermediate file to use for SST and sea-ice (cases 7 and 8 only)</i> |
| Possible Values | Any alpha-numeric string (<i>default: SST</i>) |

config_fg_interval (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Interval between intermediate files (cases 8 and 9 only)</i> |
| Possible Values | [DDD_]hh:mm:ss (<i>default: 86400</i>) |

config_landuse_data (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The land use classification to use (case 7 only)</i> |
| Possible Values | 'USGS', 'MODIFIED_IGBP_MODIS_NOAH', or 'MODIFIED_IGBP_MODIS_NOAH_15s' (<i>default: MODIFIED_IGBP_MODIS_NOAH</i>) |

config_soilcat_data (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The soil category classification to use</i> |
| Possible Values | ‘STATSGO’ or ‘BNU’ (<i>default: STATSGO</i>) |

config_topo_data (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>The topography dataset to use for the model terrain and for GWDO static fields (case 7 only)</i> |
| Possible Values | ‘GTOPO30’ or ‘GMTED2010’ (<i>default: GMTED2010</i>) |

config_vegfrac_data (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The climatological monthly vegetation fraction dataset to use (case 7 only)</i> |
| Possible Values | ‘MODIS’ or ‘NCEP’ (<i>default: MODIS</i>) |

config_albedo_data (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>The climatological monthly albedo dataset to use (case 7 only)</i> |
| Possible Values | ‘MODIS’ or ‘NCEP’ (<i>default: MODIS</i>) |

config_maxsnowalbedo_data (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>The maximum snow albedo dataset to use (case 7 only)</i> |
| Possible Values | ‘MODIS’ or ‘NCEP’ (<i>default: MODIS</i>) |

config_supersample_factor (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The supersampling factor to be used for MODIS maximum snow albedo and monthly albedo datasets (case 7 only)</i> |
| Possible Values | Positive integer values (<i>default: 3</i>) |

config_lu_supersample_factor (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The supersampling factor to be used for 30s or 15s MODIS land use, or for 30s USGS land use, as selected by ‘config_landuse_data’ (case 7 only)</i> |
| Possible Values | Positive integer values (<i>default: 1</i>) |

config_30s_supersample_factor (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>The supersampling factor to be used for 30s terrain, soil category, and MODIS FPAR monthly vegetation fraction (case 7 only)</i> |
| Possible Values | Positive integer values (<i>default: 1</i>) |

config_noahmp_static (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to process, read, and write static fields only required by Noah-MP (hidden by default)</i> |
| Possible Values | true or false (<i>default: true</i>) |

config_use_specchumd (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to use specific-humidity as the first-guess moisture variable. If this option is False, relative humidity will be used.</i> |
| Possible Values | true or false (<i>default: false</i>) |

A.4 vertical_grid**config_ztop** (real)

| | |
|-----------------|--|
| Units | <i>m</i> |
| Description | <i>Model top height</i> |
| Possible Values | Positive real values (<i>default: 30000.0</i>) |

config_nsmterrain (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Number of smoothing passes to apply to the interpolated terrain field</i> |
| Possible Values | Non-negative integer values (<i>default:</i> 1) |

config_smooth_surfaces (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to smooth zeta surfaces</i> |
| Possible Values | true or false (<i>default:</i> true) |

config_dzmin (real)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Minimum thickness of layers as a fraction of nominal thickness</i> |
| Possible Values | Real values in the interval (0,1) (<i>default:</i> 0.3) |

config_nsm (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Maximum number of smoothing passes for coordinate surfaces</i> |
| Possible Values | Positive integer values (<i>default:</i> 30) |

config_tc_vertical_grid (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to use the vertical layer profile that was developed for use in real-time TC experiments</i> |
| Possible Values | true or false (<i>default:</i> true) |

config_specified_zeta_levels (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>The name of a text file containing a list of vertical coordinate (zeta) values in increasing order at layer interfaces. (hidden by default)</i> |
| Possible Values | Any valid filename (<i>default:</i>) |

config_blend_bdy_terrain (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to blend terrain along domain boundaries with first-guess terrain. Only useful for limited-area domains.</i> |
| Possible Values | true or false (<i>default: false</i>) |

A.5 interpolation_control**config_extrap_airtemp** (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Method of extrapolation of air temperature above/below first-guess levels.</i> |
| Possible Values | 'constant' (last valid value), 'linear' (linear extrapolation based on last two values), 'lapse-rate' (0.0065 K/m from last valid value) (<i>default: lapse-rate</i>) |

A.6 preproc_stages**config_static_interp** (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to interpolate WPS static data (case 7 only)</i> |
| Possible Values | true or false (<i>default: true</i>) |

config_native_gwd_static (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to recompute subgrid-scale orography statistics directly on the native MPAS mesh (case 7 only)</i> |
| Possible Values | true or false (<i>default: true</i>) |

config_native_gwd_gsl_static (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to recompute subgrid-scale orography statistics for the GSL drag directly on the native MPAS mesh (case 7 only)</i> |
| Possible Values | true or false (<i>default: false</i>) |

config_gwd_cell_scaling (real)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Scaling factor for the effective grid cell diameter used in computation of GWD static fields (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 1.0</i>) |

config_vertical_grid (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to generate vertical grid</i> |
| Possible Values | true or false (<i>default: true</i>) |

config_met_interp (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to interpolate first-guess fields from intermediate file</i> |
| Possible Values | true or false (<i>default: true</i>) |

config_input_sst (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to re-compute SST and sea-ice fields from surface input data set; should be set to .true. when running case 8</i> |
| Possible Values | true or false (<i>default: false</i>) |

config_frac_seaice (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to switch sea-ice threshold from 0.5 to 0.02</i> |
| Possible Values | true or false (<i>default: true</i>) |

A.7 physics

config_tsk_seaice_threshold (real)

| | |
|-----------------|--|
| Units | <i>K</i> |
| Description | <i>surface temperature threshold below which water points are set to seaice points (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 100.</i>) |

A.8 io

config_pio_num_iotasks (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Number of tasks to perform file I/O</i> |
| Possible Values | Integer valued, $0 \leq \text{config_pio_num_iotasks} \leq \# \text{ MPI tasks}$, 0 indicates all tasks perform I/O (<i>default: 0</i>) |

config_pio_stride (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Stride between file I/O tasks</i> |
| Possible Values | Integer valued, $\leq (\# \text{ MPI tasks}) / \text{config_pio_num_iotasks}$ (<i>default: 1</i>) |

A.9 decomposition

config_block_decomp_file_prefix (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Prefix of graph decomposition file, to be suffixed with the MPI task count</i> |
| Possible Values | Any valid filename (<i>default: x1.40962.graph.info.part.</i>) |

config_number_of_blocks (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Number of blocks to assign to each MPI task (hidden by default)</i> |
| Possible Values | Positive integer values (<i>default: 0</i>) |

config_explicit_proc_decomp (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to use an explicit mapping of blocks to MPI tasks (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_proc_decomp_file_prefix (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Prefix of block mapping file (hidden by default)</i> |
| Possible Values | Any valid filename (<i>default: graph.info.part.</i>) |

Appendix B

Model Namelist Options

This chapter summarizes the complete set of namelist options available when running the MPAS non-hydrostatic atmosphere model. All date-time string specifications are of the form described at the beginning of Appendix A.

B.1 nhyd_model

config_time_integration (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Time integration scheme (hidden by default)</i> |
| Possible Values | ‘SRK3’ (<i>default: SRK3</i>) |

config_time_integration_order (integer)

| | |
|-----------------|--------------------------------------|
| Units | - |
| Description | <i>Order for RK time integration</i> |
| Possible Values | 2 or 3 (<i>default: 2</i>) |

config_dt (real)

| | |
|-----------------|--|
| Units | <i>s</i> |
| Description | <i>Model time step, seconds</i> |
| Possible Values | Positive real values (<i>default: 720.0</i>) |

config_calendar_type (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Simulation calendar type (hidden by default)</i> |
| Possible Values | ‘gregorian’, ‘gregorian_noleap’ (<i>default: gregorian</i>) |

config_start_time (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Starting time for model simulation</i> |
| Possible Values | ‘YYYY-MM-DD hh:mm:ss’ (<i>default: 2010-10-23_00:00:00</i>) |

config_stop_time (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Stopping time for model simulation (hidden by default)</i> |
| Possible Values | ‘YYYY-MM-DD hh:mm:ss’ (<i>default: none</i>) |

config_run_duration (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Length of model simulation</i> |
| Possible Values | [DDD_]hh:mm:ss (<i>default: 5_00:00:00</i>) |

config_split_dynamics_transport (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to super-cycle scalar transport</i> |
| Possible Values | Logical values (<i>default: true</i>) |

config_number_of_sub_steps (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Number of acoustic steps per full RK step</i> |
| Possible Values | Positive, even integer values, typically 2 or 6 depending on transport splitting (<i>default: 2</i>) |

config_dynamics_split_steps (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>When config_split_dynamics_transport = T, the number of RK steps per transport step</i> |
| Possible Values | Positive integer values (<i>default: 3</i>) |

config_h_mom_eddy_visc2 (real)

| | |
|-----------------|--|
| Units | $m^2 s^{-1}$ |
| Description | ∇^2 eddy viscosity for horizontal diffusion of momentum (hidden by default) |
| Possible Values | Positive real values (default: 0.0) |

config_h_mom_eddy_visc4 (real)

| | |
|-----------------|--|
| Units | $m^4 s^{-1}$ |
| Description | ∇^4 eddy hyper-viscosity for horizontal diffusion of momentum (hidden by default) |
| Possible Values | Positive real values (default: 0.0) |

config_v_mom_eddy_visc2 (real)

| | |
|-----------------|--|
| Units | $m^2 s^{-1}$ |
| Description | ∇^2 eddy viscosity for vertical diffusion of momentum (hidden by default) |
| Possible Values | Positive real values (default: 0.0) |

config_h_theta_eddy_visc2 (real)

| | |
|-----------------|---|
| Units | $m^2 s^{-1}$ |
| Description | ∇^2 eddy viscosity for horizontal diffusion of theta (hidden by default) |
| Possible Values | Positive real values (default: 0.0) |

config_h_theta_eddy_visc4 (real)

| | |
|-----------------|---|
| Units | $m^4 s^{-1}$ |
| Description | ∇^4 eddy hyper-viscosity for horizontal diffusion of theta (hidden by default) |
| Possible Values | Positive real values (default: 0.0) |

config_v_theta_eddy_visc2 (real)

| | |
|-----------------|---|
| Units | $m^2 s^{-1}$ |
| Description | ∇^2 eddy viscosity for vertical diffusion of theta (hidden by default) |
| Possible Values | Positive real values (default: 0.0) |

config_horiz_mixing (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Formulation of horizontal mixing</i> |
| Possible Values | ‘2d_fixed’ or ‘2d_smagorinsky’ (<i>default: 2d_smagorinsky</i>) |

config_len_disp (real)

| | |
|-----------------|---|
| Units | <i>m</i> |
| Description | <i>Horizontal length scale, used by the Smagorinsky formulation of horizontal diffusion and by 3-d divergence damping (hidden by default)</i> |
| Possible Values | Positive real values. A zero value implies that the length scale is prescribed by the nominalMinDc value in the input file. (<i>default: 0.0</i>) |

config_visc4_2dsmag (real)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Scaling coefficient of δx^3 to obtain ∇^4 diffusion coefficient</i> |
| Possible Values | Non-negative real values (<i>default: 0.05</i>) |

config_del4u_div_factor (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Scaling factor for the divergent component of $\nabla^4 u$ calculation (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 10.0</i>) |

config_w_adv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Horizontal advection order for w (hidden by default)</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_theta_adv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Horizontal advection order for theta (hidden by default)</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_scalar_adv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Horizontal advection order for scalars (hidden by default)</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_u_vadv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Vertical advection order for normal velocities (u) (hidden by default)</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_w_vadv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Vertical advection order for w (hidden by default)</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_theta_vadv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Vertical advection order for theta (hidden by default)</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_scalar_vadv_order (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Vertical advection order for scalars (hidden by default)</i> |
| Possible Values | 2, 3, or 4 (<i>default: 3</i>) |

config_scalar_advection (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to advect scalar fields</i> |
| Possible Values | .true. or .false. (<i>default: true</i>) |

config_positive_definite (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to enable positive-definite advection of scalars (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_monotonic (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to enable monotonic limiter in scalar advection</i> |
| Possible Values | .true. or .false. (<i>default: true</i>) |

config_coef_3rd_order (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Upwinding coefficient in the 3rd order advection scheme</i> |
| Possible Values | $0 \leq \text{config_coef_3rd_order} \leq 1$ (<i>default: 0.25</i>) |

config_smagorinsky_coef (real)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Dimensionless empirical parameter relating the strain tensor to the eddy viscosity in the Smagorinsky turbulence model (hidden by default)</i> |
| Possible Values | Real values typically in the range 0.1 to 0.4 (<i>default: 0.125</i>) |

config_mix_full (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Mix full θ and u fields, or mix perturbation from initial state (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: true</i>) |

config_epssm (real)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Off-centering parameter for the vertically implicit acoustic integration</i> |
| Possible Values | Positive real values (<i>default: 0.1</i>) |

config_smdiv (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>3-d divergence damping coefficient</i> |
| Possible Values | Positive real values (<i>default: 0.1</i>) |

config_apvm_upwinding (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Amount of upwinding in APVM (hidden by default)</i> |
| Possible Values | $0 \leq \text{config_apvm_upwinding} \leq 1$ (<i>default: 0.5</i>) |

config_h_ScaleWithMesh (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Scale eddy viscosities with mesh-density function for horizontal diffusion (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: true</i>) |

config_num_halos (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Number of halo layers for fields (hidden by default)</i> |
| Possible Values | Integer values, typically 2 or 3; DO NOT CHANGE (<i>default: 2</i>) |

config_relax_zone_divdamp_coef (real)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Coefficient for the divergent component of the Laplacian filter of momentum in the relaxation zone (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 6.0</i>) |

B.2 damping

config_zd (real)

| | |
|-----------------|--|
| Units | <i>m</i> |
| Description | <i>Height MSL to begin w-damping profile</i> |
| Possible Values | Positive real values (<i>default: 22000.0</i>) |

config_xnutr (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Maximum w-damping coefficient at model top</i> |
| Possible Values | $0 \leq \text{config_xnutr} \leq 1$ (<i>default: 0.2</i>) |

config_mpas_cam_coef (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Coefficient for scaling the 2nd-order horizontal mixing in the mpas_cam absorbing layer (hidden by default)</i> |
| Possible Values | $0 \leq \text{config_mpas_cam_coef} \leq 1$, standard value is 0.2 (<i>default: 0.0</i>) |

config_number_cam_damping_levels (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Number of layers in which to apply cam 2nd-order horizontal filter top of model; viscosity linearly ramps to zero by layer number from the top (hidden by default)</i> |
| Possible Values | Positive integer values (<i>default: 4</i>) |

config_rayleigh_damp_u (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to apply Rayleigh damping on horizontal velocity in the top-most model levels. The number of levels is specified by the config_number_rayleigh_damp_u_levels option, and the damping timescale is specified by the config_rayleigh_damp_u_timescale_days option. (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_rayleigh_damp_u_timescale_days (real)

| | |
|-----------------|--|
| Units | <i>days</i> |
| Description | <i>Timescale, in days (86400 s), for the Rayleigh damping on horizontal velocity in the top-most model levels. (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 5.0</i>) |

config_number_rayleigh_damp_u_levels (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Number of layers in which to apply Rayleigh damping on horizontal velocity at top of model; damping linearly ramps to zero by layer number from the top (hidden by default)</i> |
| Possible Values | Positive integer values (<i>default: 6</i>) |

B.3 limited_area

config_apply_lbcs (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to apply lateral boundary conditions</i> |
| Possible Values | true or false; this option must be set to true for limited-area simulations and false for global simulations (<i>default: false</i>) |

B.4 io

config_restart_timestamp_name (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Filename used to store most recent restart time stamp (hidden by default)</i> |
| Possible Values | Any valid filename (<i>default: restart_timestamp</i>) |

config_pio_num_iotasks (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Number of tasks to perform file I/O</i> |
| Possible Values | Integer valued, $0 \leq \text{config_pio_num_iotasks} \leq \# \text{ MPI tasks}$, 0 indicates all tasks perform I/O (<i>default: 0</i>) |

config_pio_stride (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Stride between file I/O tasks</i> |
| Possible Values | Integer valued, $\leq (\# \text{ MPI tasks}) / \text{config_pio_num_iotasks}$ (<i>default: 1</i>) |

B.5 decomposition

config_block_decomp_file_prefix (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Prefix of graph decomposition file, to be suffixed with the MPI task count</i> |
| Possible Values | Any valid filename (<i>default: x1.40962.graph.info.part.</i>) |

config_number_of_blocks (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Number of blocks to assign to each MPI task (hidden by default)</i> |
| Possible Values | Positive integer values (<i>default: 0</i>) |

config_explicit_proc_decomp (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to use an explicit mapping of blocks to MPI tasks (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_proc_decomp_file_prefix (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Prefix of block mapping file (hidden by default)</i> |
| Possible Values | Any valid filename (<i>default: graph.info.part.</i>) |

B.6 restart

config_do_restart (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether this run of the model is to restart from a previous restart file or not</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_do_DAcycling (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to re-compute coupled fields θ_m, $\tilde{\rho}$, ρu, etc. from uncoupled fields when restarting the model; used for cycling DA experiments that analyze uncoupled fields in restart files (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

B.7 printout**config_print_global_minmax_vel** (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to print the global min/max of horizontal normal velocity and vertical velocity each timestep</i> |
| Possible Values | .true. or .false. (<i>default: true</i>) |

config_print_detailed_minmax_vel (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to print the global min/max of horizontal normal velocity and vertical velocity each timestep, along with the location in the domain where those extrema occurred</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_print_global_minmax_sca (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Whether to print the global min/max of scalar fields each timestep (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

B.8 IAU**config_IAU_option** (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Incremental Analysis Update scheme</i> |
| Possible Values | ‘off’ or ‘on’; ‘off’ turns off IAU, ‘on’ uses equal weighting of increments across the IAU window (<i>default: off</i>) |

config_IAU_window_length_s (real)

| | |
|-----------------|--|
| Units | <i>s</i> |
| Description | <i>Length of window over which analysis increments are applied</i> |
| Possible Values | Non-negative real values (<i>default: 21600.</i>) |

B.9 assimilation

config_jedi_da (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether this run is within the JEDI data assimilation framework; used to add temperature and specific humidity as diagnostics (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

B.10 development

config_halo_exch_method (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Method to use for exchanging halos (hidden by default)</i> |
| Possible Values | ‘mpas_dmpar’, ‘mpas_halo’ (<i>default: mpas_halo</i>) |

B.11 physics

input_soil_data (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>input soil use classification (hidden by default)</i> |
| Possible Values | ‘STAS’ (<i>default: STAS</i>) |

input_soil_temperature_lag (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>days over which the deep soil temperature is computed using skin temperature (hidden by default)</i> |
| Possible Values | Positive integers (<i>default: 140</i>) |

num_soil_layers (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>number of soil layers in Noah land surface scheme (hidden by default)</i> |
| Possible Values | Positive integers. For Noah LSM, must be set to 4. (<i>default: 4</i>) |

months (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>number of months per year (hidden by default)</i> |
| Possible Values | Positive integer values. DO NOT CHANGE. (<i>default: 12</i>) |

noznlev (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>number of prescribed pressure levels for input climatological ozone volume mixing ratios (hidden by default)</i> |
| Possible Values | Positive integers (<i>default: 59</i>) |

naerlev (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>number of prescribed pressure levels for input climatological aerosol mixing ratio (hidden by default)</i> |
| Possible Values | Positive integers (<i>default: 29</i>) |

camdim1 (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>dimension of CAM radiation absorption save array (hidden by default)</i> |
| Possible Values | Positive integers (<i>default: 4</i>) |

config_frac_seaice (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>logical for configuration of fractional sea-ice (hidden by default)</i> |
| Possible Values | .true. for sea-ice between 0 or 1; .false. for sea-ice equal to 0 or 1 (flag). (<i>default: true</i>) |

config_sfc_albedo (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>logical for configuration of surface albedo (hidden by default)</i> |
| Possible Values | .true. for climatologically varying surface albedo; .false. for fixed input data (<i>default: true</i>) |

config_sfc_snowalbedo (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>logical for configuration of maximum surface albedo for snow (hidden by default)</i> |
| Possible Values | .true. for geographical distribution; .false. for fixed input data (<i>default: true</i>) |

config_sst_update (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>logical for configuration of sea-surface temperature</i> |
| Possible Values | .true. for time-varying sea-surface temperatures; .false., otherwise (<i>default: false</i>) |

config_sstdiurn_update (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>logical for configuration of diurnal cycle of sea-surface temperatures</i> |
| Possible Values | .true. for applying a diurnal cycle to sea-surface temperatures; .false., otherwise (<i>default: false</i>) |

config_deepsoiltemp_update (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>logical for configuration of deep soil temperatures</i> |
| Possible Values | .true. for slowly time-varying deep soil temperatures; .false. otherwise (<i>default: false</i>) |

config_o3climatology (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>logical for configuration of input ozone data in RRMTG long- and short-wave radiation (hidden by default)</i> |
| Possible Values | .true. for using monthly-varying ozone data; .false. for using fixed vertical profile (<i>default: true</i>) |

config_microp_re (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>logical for calculation of the effective radii for cloud water, cloud ice, and snow (hidden by default)</i> |
| Possible Values | .true. for calculating effective radii; .false. for using defaults in RRTMG radiation (<i>default: false</i>) |

config_ysu_pblmix (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>logical for turning on/off top-down, radiation_driven mixing (hidden by default)</i> |
| Possible Values | .true. to turn on top-down radiation_driven mixing; .false. otherwise (<i>default: false</i>) |

config_urban_physics (logical)

| | |
|-----------------|---|
| Units | - |
| Description | <i>logical for turn on/off the urban physics parameterization (hidden by default)</i> |
| Possible Values | .true. to turn on the urban physics; .false. otherwise (<i>default: false</i>) |

config_n_microp (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>number of microphysics time-steps per physics time-steps (hidden by default)</i> |
| Possible Values | Positive integers (<i>default: 1</i>) |

config_radtlw_interval (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>time interval between calls to parameterization of long-wave radiation</i> |
| Possible Values | 'DD_HH:MM:SS' or 'none' (<i>default: 00:30:00</i>) |

config_radtsw_interval (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>time interval between calls to parameterization of short-wave radiation</i> |
| Possible Values | 'DD_HH:MM:SS' or 'none' (<i>default: 00:30:00</i>) |

config_conv_interval (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>time interval between calls to parameterization of convection (hidden by default)</i> |
| Possible Values | 'DD_HH:MM:SS' or 'none' (<i>default: none</i>) |

config_pbl_interval (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>time interval between calls to parameterization of planetary boundary layer (hidden by default)</i> |
| Possible Values | 'DD_HH:MM:SS' or 'none' (<i>default: none</i>) |

config_camrad_abs_update (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>time interval between updates of absorption/emission coefficients in CAM radiation (hidden by default)</i> |
| Possible Values | 'DD_HH:MM:SS' or 'none' (<i>default: 06:00:00</i>) |

config_greeness_update (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>time interval between updates of greeness fraction (hidden by default)</i> |
| Possible Values | 'DD_HH:MM:SS' or 'none' (<i>default: 24:00:00</i>) |

config_bucket_update (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>time interval between updates of accumulated rain and radiation diagnostics</i> |
| Possible Values | 'DD_HH:MM:SS' or 'none' (<i>default: none</i>) |

config_physics_suite (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>Choice of physics suite</i> |
| Possible Values | 'mesoscale_reference','convection_permitting','none' <i>(default: mesoscale_reference)</i> |

config_microp_scheme (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>configuration for cloud microphysics schemes (hidden by default)</i> |
| Possible Values | ‘suite’, ‘mp_wsm6’, ‘mp_thompson’, ‘mp_thompson_aerosols’, ‘mp_kessler’, ‘off’ (<i>default: suite</i>) |

config_convection_scheme (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>configuration for convection schemes (hidden by default)</i> |
| Possible Values | ‘suite’, ‘cu_kain_fritsch’, ‘cu_tiedtke’, ‘cu_ntiedtke’, ‘cu_grell_freitas’, ‘off’ (<i>default: suite</i>) |

config_lsm_scheme (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>configuration for land-surface schemes (hidden by default)</i> |
| Possible Values | ‘suite’, ‘sf_noah’, ‘sf_noahmp’, ‘off’ (<i>default: suite</i>) |

config_pbl_scheme (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>configuration for planetary boundary layer schemes (hidden by default)</i> |
| Possible Values | ‘suite’, ‘bl_ysu’, ‘bl_mynn’, ‘off’ (<i>default: suite</i>) |

config_gwdo_scheme (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>configuration of gravity wave drag over orography (hidden by default)</i> |
| Possible Values | ‘suite’, ‘bl_ysu_gwdo’, ‘bl_ugwp_gwdo’, ‘off’ (<i>default: suite</i>) |

config_ngw_scheme (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>logical for non-stationary gravity wave drag scheme (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_knob_ugwp_tauamp (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>non-stationary gravity wave drag absolute momentum flux at launch level (hidden by default)</i> |
| Possible Values | Non-negative real values (<i>default: 0.13e-3</i>) |

config_ugwp_diags (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>logical for outputting ugwp drag suite diagnostic variables (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_radt_cld_scheme (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>configuration for calculation of horizontal cloud fraction (hidden by default)</i> |
| Possible Values | ‘suite’,‘cld_fraction’,‘cld_incidence’ (<i>default: suite</i>) |

config_radt_lw_scheme (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>configuration for long-wave radiation schemes (hidden by default)</i> |
| Possible Values | ‘suite’,‘rrtmg_lw’,‘cam_lw’,‘off’ (<i>default: suite</i>) |

config_radt_sw_scheme (character)

| | |
|-----------------|---|
| Units | - |
| Description | <i>configuration for short-wave radiation schemes (hidden by default)</i> |
| Possible Values | ‘suite’,‘rrtmg_sw’,‘cam_sw’,‘off’ (<i>default: suite</i>) |

config_sfclayer_scheme (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>configuration for surface layer-scheme (hidden by default)</i> |
| Possible Values | ‘suite’,‘sf_monin_obukhov’,‘sf_mynn’,‘off’ (<i>default: suite</i>) |

config_radt_cld_overlap (character)

| | |
|-----------------|---|
| Units | |
| Description | <i>cloud overlapping option in the RRTMG lw and sw radiation schemes (hidden by default)</i> |
| Possible Values | 'none','random','maximum_random','maximum','exponential','exponential_random' <i>(default: maximum_random)</i> |

config_radt_cld_dcorrlen (character)

| | |
|-----------------|---|
| Units | |
| Description | <i>decorrelation length for cloud overlapping option in the RRTMG lw and sw radiation schemes (hidden by default)</i> |
| Possible Values | 'constant','latitude_varying' <i>(default: constant)</i> |

config_gfconv_closure_deep (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>closure option for deep convection in Grell-Freitas convection scheme (hidden by default)</i> |
| Possible Values | 0 <i>(default: 0)</i> |

config_gfconv_closure_shallow (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>closure option for shallow convection in Grell-Freitas convection scheme (hidden by default)</i> |
| Possible Values | 8 <i>(default: 8)</i> |

config_myNN_tkeadvect (logical)

| | |
|-----------------|---|
| Units | - |
| Description | = true:do MYNN TKE advection <i>(hidden by default)</i> |
| Possible Values | .true. or .false. <i>(default: false)</i> |

config_myNN_tkebudget (integer)

| | |
|-----------------|---|
| Units | - |
| Description | =1:option to add the MYNN TKE budget terms to output <i>(hidden by default)</i> |
| Possible Values | 0,1 <i>(default: 0)</i> |

config_mynn_closure (real)

| | |
|-----------------|---|
| Units | - |
| Description | <i>2.5 level or 3.0 level for the MYNN TKE turbulence closure (hidden by default)</i> |
| Possible Values | 2.0,2.5,3.0 (<i>default: 2.5</i>) |

config_mynn_cloupdf (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>switch to different cloud PDFs to represent subgrid clouds in the MYNN PBL scheme (hidden by default)</i> |
| Possible Values | 0,1,2 (<i>default: 2</i>) |

config_mynn_mixlength (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>mixing length in the MYNN PBL scheme:=0(Nakanishi and Niino 2009);=1(RAP/HRRR);=2(Ito et al. 2015) (hidden by default)</i> |
| Possible Values | 0,1,2 (<i>default: 2</i>) |

config_mynn_stfunc (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>option to switch flux profile relationship for surface (0:Dyer-Hicks, 1:Cheng-Brustaert (hidden by default))</i> |
| Possible Values | 0,1 (<i>default: 1</i>) |

config_mynn_topdown (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>option to add top-down diffusion driven by cloud-top radiative cooling (hidden by default)</i> |
| Possible Values | 0,1 (<i>default: 0</i>) |

config_mynn_scaleaware (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>option to turn on the scale-aware option (hidden by default)</i> |
| Possible Values | 0,1 (<i>default: 1</i>) |

config_mynn_dheat_opt (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>option to activate heating due to dissipation of TKE (hidden by default)</i> |
| Possible Values | 0,1 (<i>default: 1</i>) |

config_mynn_edmf (integer)

| | |
|-----------------|---|
| Units | - |
| Description | =1:activate the EDMF option in the MYNN PBL scheme (=0 otherwise) (hidden by default) |
| Possible Values | 0,1 (<i>default: 1</i>) |

config_mynn_edmf_dd (integer)

| | |
|-----------------|---|
| Units | |
| Description | =1:activate the EDMF downdraft option in the MYNN PBL scheme (=0 otherwise) (hidden by default) |
| Possible Values | 0,1 (<i>default: 0</i>) |

config_mynn_edmf_mom (integer)

| | |
|-----------------|---|
| Units | - |
| Description | =1:activate momentum transport with the EDMF option of the MYNN PBL scheme (=0 otherwise) (hidden by default) |
| Possible Values | 0,1 (<i>default: 0</i>) |

config_mynn_edmf_tke (integer)

| | |
|-----------------|--|
| Units | - |
| Description | =1:activate TKE transport with the EDMF option of the MYNN PBL scheme (=0 otherwise) (hidden by default) |
| Possible Values | 0,1 (<i>default: 0</i>) |

config_mynn_edmf_output (integer)

| | |
|-----------------|--|
| Units | - |
| Description | =0:suppress the allocation of variables needed in the EDMF option of the MYNN PBL scheme (hidden by default) |
| Possible Values | 0,1 (<i>default: 0</i>) |

config_mynn_mixscalars (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>=1: activate mixing of scalars in the MYNN PBL scheme (=0 otherwise) (hidden by default)</i> |
| Possible Values | 0,1 (<i>default: 1</i>) |

config_mynn_mixclouds (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>1: activate mixing of cloud condensates in the MYNN PBL scheme (hidden by default)</i> |
| Possible Values | 0,1 (<i>default: 1</i>) |

config_mynn_mixqt (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>=0: activate mixing of moisture species separately; = 1: activate mixing of total water in the mynn PBL scheme (hidden by default)</i> |
| Possible Values | 0,1 (<i>default: 0</i>) |

config_bucket_radt (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>threshold above which accumulated radiation diagnostics are reset (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 1.0e9</i>) |

config_bucket_rainc (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>threshold above which the accumulated convective precipitation is reset (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 100.0</i>) |

config_bucket_rainnc (real)

| | |
|-----------------|--|
| Units | - |
| Description | <i>threshold above which the accumulated grid-scale precipitation is reset (hidden by default)</i> |
| Possible Values | Positive real values (<i>default: 100.0</i>) |

config_oml1d (logical)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Whether to activate the 1-d ocean mixed-layer model (hidden by default)</i> |
| Possible Values | .true. or .false. (<i>default: false</i>) |

config_oml_hml0 (real)

| | |
|-----------------|---|
| Units | m |
| Description | <i>If greater than zero, provides the constant, initial mixed-layer depth; if zero, initial mixed-layer depth will be taken from the ‘oml_initial’ field. (hidden by default)</i> |
| Possible Values | Non-negative real values (<i>default: 30.0</i>) |

config_oml_gamma (real)

| | |
|-----------------|---|
| Units | $K m^{-1}$ |
| Description | <i>Deep water lapse rate in 1-d OML model (hidden by default)</i> |
| Possible Values | Real values (<i>default: 0.14</i>) |

config_oml_relaxation_time (real)

| | |
|-----------------|---|
| Units | s |
| Description | <i>Relaxation time to initial values in 1-d OML (hidden by default)</i> |
| Possible Values | Non-negative real values (<i>default: 864000.</i>) |

B.12 soundings**config_sounding_interval** (character)

| | |
|-----------------|--|
| Units | - |
| Description | <i>Interval between writing of soundings. A value of ‘none’ disables writing of soundings.</i> |
| Possible Values | ’[DDD_]hh:mm:ss’ or ‘none’ (<i>default: none</i>) |

B.13 physics_lsm_noahmp

config_noahmp_ipt_dveg (integer)

| | |
|-----------------|--------------------------------------|
| Units | - |
| Description | <i>option for dynamic vegetation</i> |
| Possible Values | 1 to 6 (<i>default: 4</i>) |

config_noahmp_ipt_crs (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>option for canopy stomatal resistance</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_btr (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>option for soil moisture in stomatal resistance</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_runsrf (integer)

| | |
|-----------------|----------------------------------|
| Units | - |
| Description | <i>option for surface runoff</i> |
| Possible Values | 0 or 1 (<i>default: 3</i>) |

config_noahmp_ipt_runsub (integer)

| | |
|-----------------|--------------------------------------|
| Units | - |
| Description | <i>option for sub-surface option</i> |
| Possible Values | 0 or 1 (<i>default: 3</i>) |

config_noahmp_ipt_sfcl (integer)

| | |
|-----------------|--------------------------------|
| Units | - |
| Description | <i>option for surface drag</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_frz (integer)

| | |
|-----------------|-------------------------------------|
| Units | - |
| Description | <i>option for supercooled water</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_inf (integer)

| | |
|-----------------|-------------------------------|
| Units | - |
| Description | <i>option for frozen soil</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_rad (integer)

| | |
|-----------------|--------------------------------------|
| Units | - |
| Description | <i>option for radiative transfer</i> |
| Possible Values | 0 or 1 (<i>default: 3</i>) |

config_noahmp_ipt_alb (integer)

| | |
|-----------------|-------------------------------|
| Units | - |
| Description | <i>option for snow albedo</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_snf (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>option for pcp partition option</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_tksno (integer)

| | |
|-----------------|---|
| Units | - |
| Description | <i>option for snow thermal conductivity</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_tbot (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>option for lower boundary of soil temperature</i> |
| Possible Values | 0 or 1 (<i>default: 2</i>) |

config_noahmp_ipt_stc (integer)

| | |
|-----------------|--|
| Units | - |
| Description | <i>option for soil or snow time scheme</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_gla (integer)

| | |
|-----------------|------------------------------|
| Units | - |
| Description | <i>option for glacier</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_rsf (integer)

| | |
|-----------------|---------------------------------------|
| Units | - |
| Description | <i>option for subrface resistance</i> |
| Possible Values | 0 or 1 (<i>default: 4</i>) |

config_noahmp_ipt_soil (integer)

| | |
|-----------------|------------------------------|
| Units | - |
| Description | <i>option for soil data</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_pedo (integer)

| | |
|-----------------|---------------------------------|
| Units | - |
| Description | <i>option for pedo transfer</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_crop (integer)

| | |
|-----------------|------------------------------|
| Units | - |
| Description | <i>option for crop</i> |
| Possible Values | 0 or 1 (<i>default: 0</i>) |

config_noahmp_ipt_irr (integer)

| | |
|-----------------|------------------------------|
| Units | - |
| Description | <i>option for irrigation</i> |
| Possible Values | 0 or 1 (<i>default: 0</i>) |

config_noahmp_ipt_irrm (integer)

| | |
|-----------------|-------------------------------------|
| Units | - |
| Description | <i>option for irrigation method</i> |
| Possible Values | 0 or 1 (<i>default: 0</i>) |

config_noahmp_ipt_infdv (integer)

| | |
|-----------------|-------------------------------------|
| Units | |
| Description | <i>option for dvic infiltration</i> |
| Possible Values | 0 or 1 (<i>default: 1</i>) |

config_noahmp_ipt_tdrn (integer)

| | |
|-----------------|-----------------------------------|
| Units | - |
| Description | <i>option for drainage option</i> |
| Possible Values | 0 or 1 (<i>default: 0</i>) |

Appendix C

Grid Description

This chapter provides a brief introduction to the common types of grids used in the MPAS framework.

C.1 Horizontal grid

The MPAS grid system requires the definition of seven elements. These seven elements are composed of two types of *cells*, two types of *lines*, and three types of *points*. These elements are depicted in Figure C.1 and defined in Table C.1. These elements can be defined on either the plane or the surface of the sphere. The two types of cells form two meshes, a primal mesh composed of Voronoi regions and a dual mesh composed of Delaunay triangles. Each corner of a primal mesh cell is uniquely associated with the “center” of a dual mesh cell and vice versa. So we define the two mesh as either a primal mesh (composed of cells P_i) or a dual mesh (composed of cells D_v). The center of any primal mesh cell, P_i , is denoted by \mathbf{x}_i and the center of any the dual mesh cell, D_v , is denoted by \mathbf{x}_v . The boundary of a given primal mesh cell P_i is composed of the set of lines that connect the \mathbf{x}_v locations of associated dual mesh cells D_v . Similarly, the boundary of a given dual mesh cell D_v is composed of the set of lines that connect the \mathbf{x}_i locations of the associated primal mesh cells P_i . As shown in Figure C.1, a line segment that connects two primal mesh cell centers is uniquely associated with a line segment that connects two dual mesh cell centers. We assume that these two line segments cross and the point of intersection is labeled as \mathbf{x}_e . In addition, we assume that these two line segments are orthogonal as indicated in Figure C.1. Each \mathbf{x}_e is associated with two distances: d_e measures the distance between the primal mesh cells sharing \mathbf{x}_e and l_e measures the distance between the dual mesh cells sharing \mathbf{x}_e .

Since the two line segments crossing at \mathbf{x}_e are orthogonal, these line segments form a convenient local coordinate system for each edge. At each \mathbf{x}_e location a unit vector \mathbf{n}_e is defined to be parallel to the line connecting primal mesh cells. A second unit vector \mathbf{t}_e is defined such that $\mathbf{t}_e = \mathbf{k} \times \mathbf{n}_e$.

Table C.2 provides the names of all *elements* and all *sets of elements* as used in the MPAS framework. Elements appear twice in the table when described in the grid file in more than one way, e.g. points are described with both cartesian and latitude/longitude coordinates. An “ncdump -h” of any MPAS grid, output or restart file will contain all variable names shown in second column of Table C.2.

In addition to these seven element types, we require the definition of *sets of elements*. In all, eight different types of sets are required and these are defined and explained in Table C.3 and Figure C.2. The notation is always of the form of, for example, $i \in CE(e)$, where the LHS indicates the type of element to be gathered (cells) based on the RHS relation to another type of element (edges).

The angle of each edge in an MPAS grid is provided in the variable *angleEdge*. The angle given is the angle between a vector pointing north and a vector pointing in the positive tangential direction of the edge. Referring to Fig. C.3,

$$\text{angleEdge} = \arcsin \|\hat{\mathbf{n}} \times \hat{\mathbf{v}}\|,$$

where $\hat{\mathbf{n}}$ is the unit vector pointing north and $\hat{\mathbf{v}}$ is the unit vector pointing from *verticesOnEdge*(1,iEdge) to *verticesOnEdge*(2,iEdge).

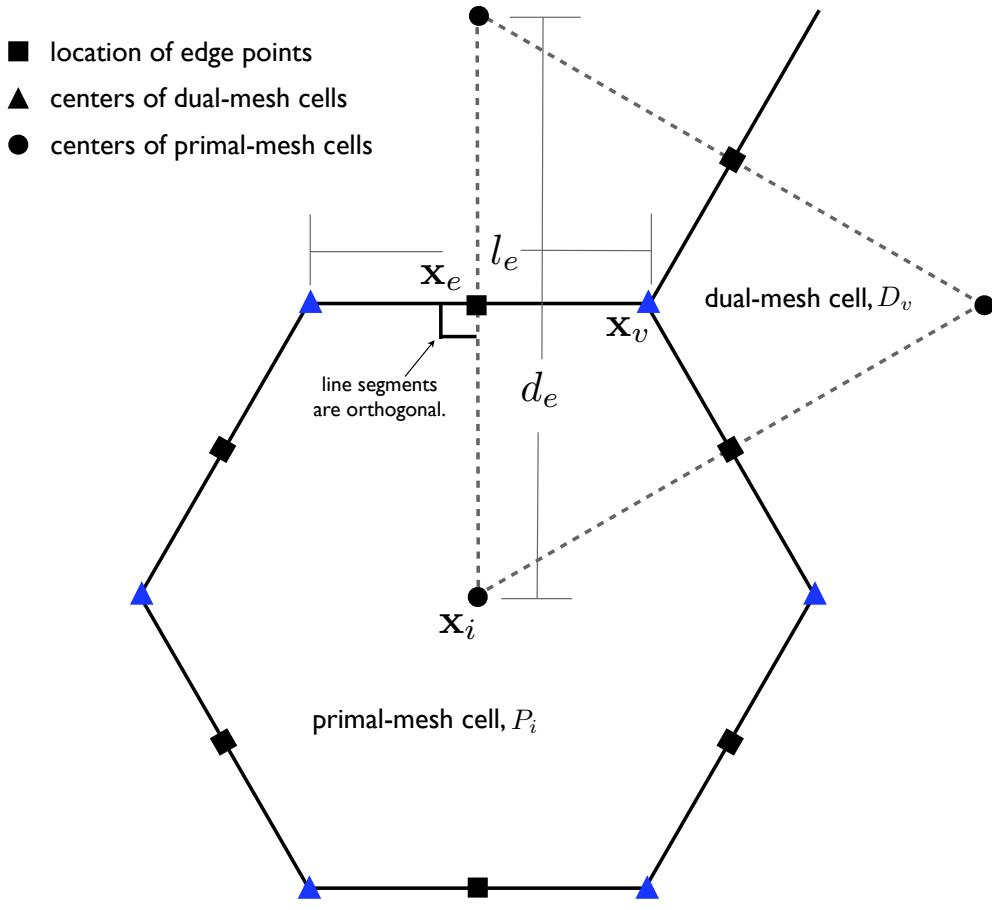


Figure C.1: Definition of elements used to build the MPAS grid. Also see Table C.1.

Table C.1: Definition of elements used to build the MPAS grid.

| <i>Element</i> | <i>Type</i> | <i>Definition</i> |
|----------------|--------------|---|
| \mathbf{x}_i | point | location of center of primal-mesh cells |
| \mathbf{x}_v | point | location of center of dual-mesh cells |
| \mathbf{x}_e | point | location of edge points where velocity is defined |
| d_e | line segment | distance between neighboring \mathbf{x}_i locations |
| l_e | line segment | distance between neighboring \mathbf{x}_v locations |
| P_i | cell | a cell on the primal-mesh |
| D_v | cell | a cell on the dual-mesh |

Given a wind vector $(u_{\perp}, u_{\parallel})$ defined in term of components orthogonal to and parallel to the edge, the earth-relative wind (u, v) may be recovered as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} u_{\perp} \\ u_{\parallel} \end{bmatrix},$$

where $\alpha = \text{angleEdge}$.

Table C.2: Variable names used to describe a MPAS grid.

| <i>Element</i> | <i>Name</i> | <i>Size</i> | <i>Comment</i> |
|----------------|-----------------|--------------------|---|
| \mathbf{x}_i | {x,y,z}Cell | nCells | cartesian location of \mathbf{x}_i |
| \mathbf{x}_i | {lon,lat}Cell | nCells | longitude and latitude of \mathbf{x}_i |
| \mathbf{x}_v | {x,y,z}Vertex | nVertices | cartesian location of \mathbf{x}_v |
| \mathbf{x}_v | {lon,lat}Vertex | nVertices | longitude and latitude of \mathbf{x}_v |
| \mathbf{x}_e | {x,y,z}Edge | nEdges | cartesian location of \mathbf{x}_e |
| \mathbf{x}_e | {lon,lat}Edge | nEdges | longitude and latitude of \mathbf{x}_e |
| d_e | dcEdge | nEdges | distance between \mathbf{x}_i locations |
| l_e | dvEdge | nEdges | distance between \mathbf{x}_v locations |
| $e \in EC(i)$ | edgesOnCell | (nEdgesMax,nCells) | edges that define P_i . |
| $e \in EV(v)$ | edgesOnVertex | (3,nCells) | edges that define D_v . |
| $i \in CE(e)$ | cellsOnEdge | (2,nEdges) | primal-mesh cells that share edge e . |
| $i \in CV(v)$ | cellsOnVertex | (3,nVertices) | primal-mesh cells that define D_v . |
| $v \in VE(e)$ | verticesOnEdge | (2,nEdges) | dual-mesh cells that share edge e . |
| $v \in VI(i)$ | verticesOnCell | (nEdgesMax,nCells) | vertices that define P_i . |

Table C.3: Definition of element groups used to reference connections in the MPAS grid. Examples are provided in Figure C.2.

| Syntax | output |
|-------------------|---|
| $e \in EC(i)$ | set of edges that define the boundary of P_i . |
| $e \in EV(v)$ | set of edges that define the boundary of D_v . |
| $i \in CE(e)$ | two primal-mesh cells that share edge e . |
| $i \in CV(v)$ | set of primal-mesh cells that form the vertices of dual mesh cell D_v . |
| $v \in VE(e)$ | the two dual-mesh cells that share edge e . |
| $v \in VI(i)$ | the set of dual-mesh cells that form the vertices of primal-mesh cell P_i . |
| $e \in ECP(e)$ | edges of cell pair meeting at edge e . |
| $e \in EVC(v, i)$ | edge pair associated with vertex v and mesh cell i . |

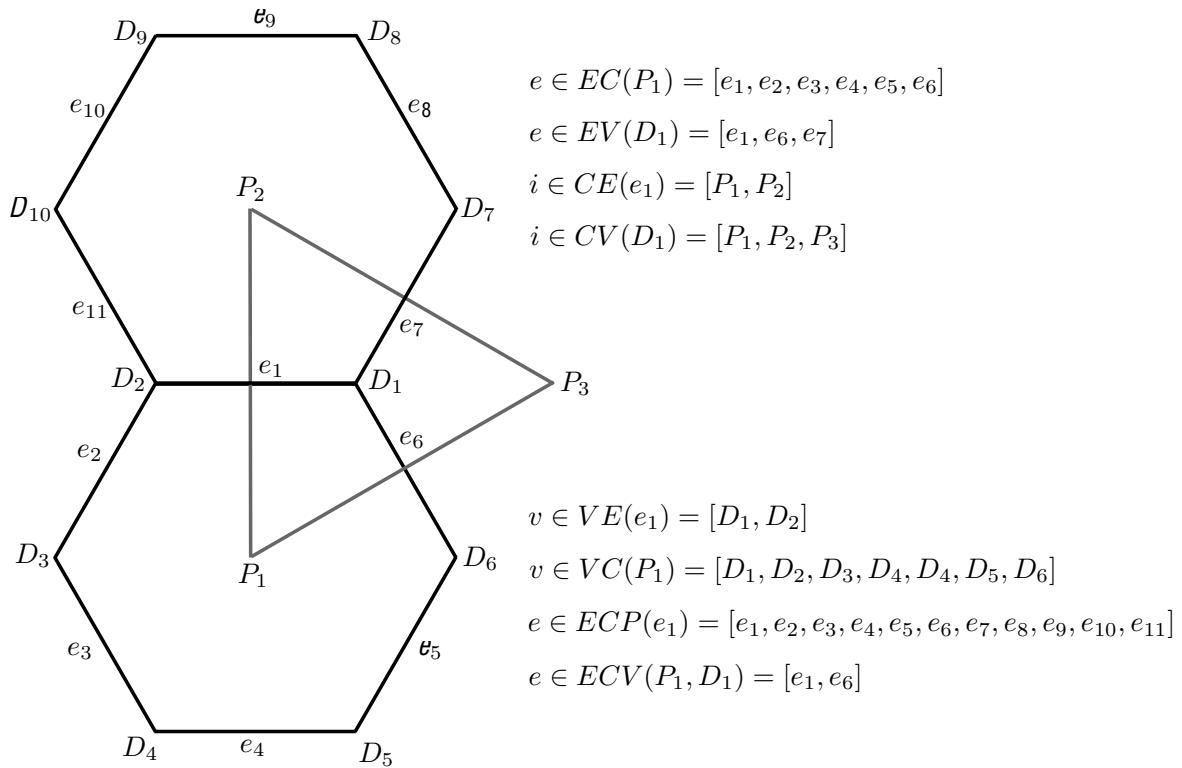


Figure C.2: Definition of element groups used to reference connections in the MPAS grid. Also see Table C.3.

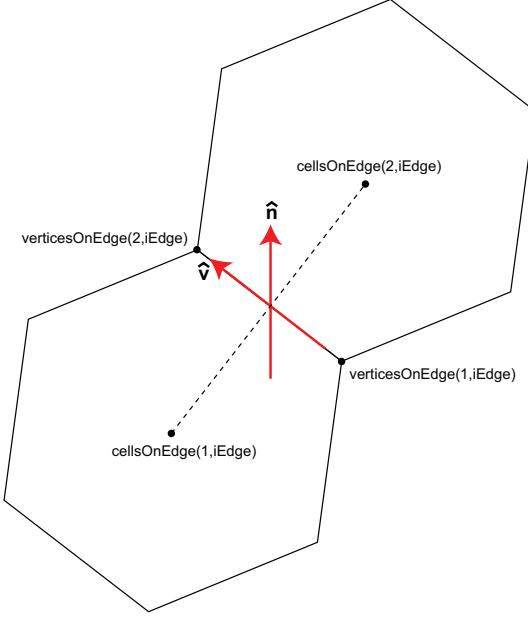


Figure C.3: The angle of an edge refers to the angle between a vector pointing north at an edge location and a vector pointing in the positive tangential velocity direction of the edge.

C.2 Vertical grid

The vertical coordinate in MPAS-Atmosphere is ζ and has units of length, where $0 \leq \zeta \leq z_t$ and z_t is the height of the model top. The relationship between the vertical coordinate and height in the physical domain is given as

$$z = \zeta + A h_s(x, y, \zeta) \quad (\text{C.1})$$

where (x, y) denotes a location on the horizontal mesh and ζ is the vertical coordinate (ζ is directed radially outward from the surface of the sphere, or perpendicular to the horizontal (x, y) plane in a Cartesian coordinate MPAS-A configuration). MPAS-A can be configured with the traditional Gal-Chen and Somerville terrain-following coordinate by setting $h_s(x, y, \zeta) = h(x, y)$ and $A = 1 - \zeta/z_t$, where $h(x, y)$ is the terrain height. Alternatively, A can be modified to allow a more rapid or less rapid transition to the constant-height upper boundary condition. Additionally, a constant-height coordinate can be specified at some intermediate height below h_t .

The influence of the terrain on any coordinate surface ζ can be influenced by the specification of $h_s(x, y, \zeta)$. Specifically, h_s can be set such that $h_s(x, y, 0) = h(x, y)$ (i.e. terrain following at the surface), and progressively filtered fields of $h(x, y)$ can be used at $\zeta > 0$ in $h_s(x, y, \zeta)$, such that the small-scale features in the topography are quickly filtered from the coordinate. Example MPAS-A vertical meshes are given in Figure C.4

On the MPAS-A mesh C-grid staggering, the state variables u , ρ , θ and scalars are located halfway between w levels in both physical height and in the coordinate ζ . Variables associated with the coordinate systems used in the MPAS-A solver, and possibly appearing in its input, output or history files, are defined in Table C.4 and depicted in Figure C.5.

Further information about the vertical coordinate can be found in Klemp, J. B. (2011). A Terrain-Following Coordinate with Smoothed Coordinate Surfaces. *Mon. Wea. Rev.*, **139**, 2163–2169. doi:10.1175/MWR-D-10-05046.1

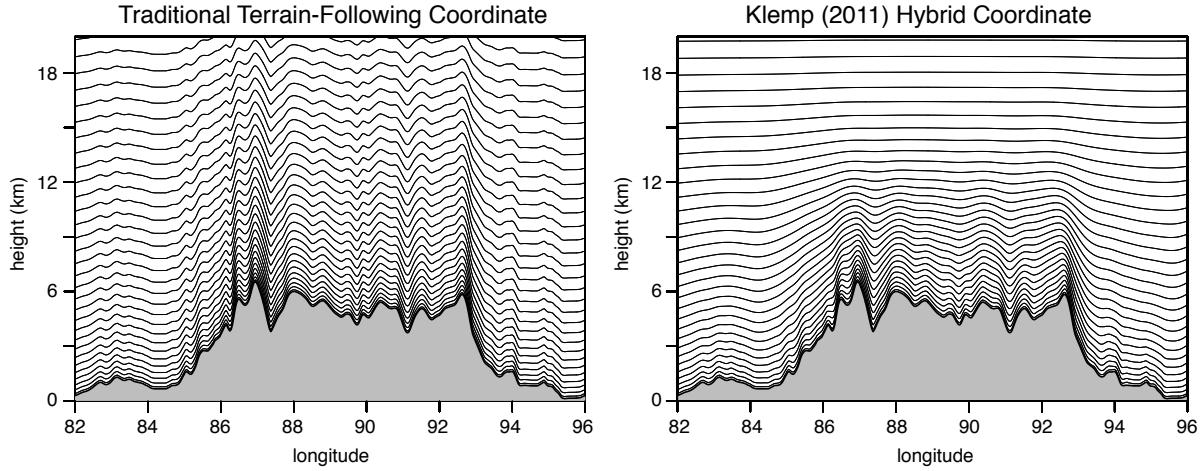


Figure C.4: Example MPAS-A vertical meshes using terrain following (left) and smoothed (right) vertical coordinates.

Table C.4: Vertical coordinate variables in MPAS-Atmosphere. *level* is the integer model level (usually specified with index k where $k = 1$ is the lowest model level and physical height increases with increasing k). Δ denotes a vertical difference between levels, and *cell* is a given mesh cell on the primary mesh.

| <i>Variable</i> | <i>Definition</i> |
|----------------------|---|
| $zgrid(level, cell)$ | physical height of the w points in meters. |
| $zw(level)$ | ζ at w levels. |
| $zu(level)$ | ζ at u levels; $zu(k) = [zw(k+1) + zw(k)]/2$. |
| $dzw(level)$ | $\Delta\zeta$ at u levels; $dzw(k) = zw(k+1) - zw(k)$. |
| $dzu(level)$ | $\Delta\zeta$ at w levels; $dzu(k) = [dzw(k+1) + dzw(k)]/2$. |
| $rdzw(level)$ | $1/dzw$. |
| $rdzu(level)$ | $1/dzu$. |
| $zz(level, cell)$ | $\Delta\zeta/\Delta z$ at u levels; $(zw(k+1) - zw(k))/(zgrid(k+1, cell) - zgrid(k, cell))$. |
| $fzm(level)$ | weight for linear interpolation to $w(k)$ point for $u(k)$ level variable. |
| $fzp(level)$ | weight for linear interpolation to $w(k)$ point for $u(k-1)$ level variable. |

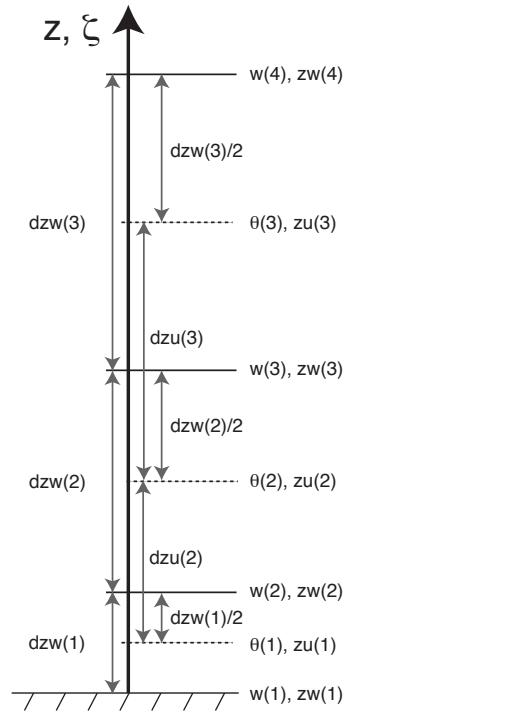


Figure C.5: Vertical distribution of the variables in MPAS-A. Also see Table C.4.

Appendix D

Description of Model Fields

Every field that may be read or written in a NetCDF *stream* (as described in Chapter 5) by the MPAS-Atmosphere model is described in this chapter. The dimensionality of each field is given in Fortran storage order (i.e., the fastest-varying dimension is inner-most).

a_tri (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>implicit tridiagonal solve coefficients</i> |
| Accessed in code | as ‘a_tri’ from the ‘diag’ pool |

ABSMF (real) (lat, days)

| | |
|------------------|--|
| Units | <i>Pa</i> |
| Description | <i>gravity wave absolute momentum flux from NGW file</i> |
| Allocated by | ugwp_ngw_stream |
| Accessed in code | as ‘ABSMF’ from the ‘ngw_input’ pool |

absnxt (real) (nVertLevels, cam_dim1, nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>Total nearest layer absorptivity</i> |
| Accessed in code | as ‘absnxt’ from the ‘diag_physics’ pool |

abstot (real) (nVertLevelsP1, nVertLevelsP1, nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>Total absorptivity</i> |
| Accessed in code | as ‘abstot’ from the ‘diag_physics’ pool |

acc_dwater (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>accumulated canopy,snow,soil water change between dt_soil</i> |
| Accessed in code | as ‘acc_dwater’ from the ‘diag_physics_noahmp’ pool |

acc_dwaterxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>accumulated snow,soil,canopy water change per soil timestep</i> |
| Accessed in code | as ‘acc_dwaterxy’ from the ‘output_noahmp’ pool |

acc_ecan (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>accumulated net canopy evaporation between dt_soil</i> |
| Accessed in code | as ‘acc_ecan’ from the ‘diag_physics_noahmp’ pool |

acc_ecanxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>accumulated net canopy evaporation per soil timestep</i> |
| Accessed in code | as ‘acc_ecanxy’ from the ‘output_noahmp’ pool |

acc_edir (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>accumulated net soil evaporation between dt_soil</i> |
| Accessed in code | as ‘acc_edir’ from the ‘diag_physics_noahmp’ pool |

acc_edirxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>accumulated net ground (soil/snow) evaporation per soil timestep</i> |
| Accessed in code | as ‘acc_edirxy’ from the ‘output_noahmp’ pool |

acc_etran (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>accumulated transpiration between dt_soil</i> |
| Accessed in code | as ‘acc_etran’ from the ‘diag_physics_noahmp’ pool |

acc_etran (real) (nSoilLevels, nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>accumulated etran between dt_soil</i> |
| Accessed in code | as ‘acc_etran’ from the ‘diag_physics_noahmp’ pool |

acc_etranixy (real) (nSoilLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m\ s^{-1}$ |
| Description | <i>accumulated transpiration rate within soil timestep</i> |
| Accessed in code | as ‘acc_etranixy’ from the ‘output_noahmp’ pool |

acc_etranxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | mm |
| Description | <i>accumulated transpiration per soil timestep</i> |
| Accessed in code | as ‘acc_etranxy’ from the ‘output_noahmp’ pool |

acc_prcp (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>accumulated precipitation between dt_soil</i> |
| Accessed in code | as ‘acc_prcp’ from the ‘diag_physics_noahmp’ pool |

acc_prcpxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | mm |
| Description | <i>accumulated precipitation per soil timestep</i> |
| Accessed in code | as ‘acc_prcpxy’ from the ‘output_noahmp’ pool |

acc_qinsur (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>accumulated qinsur between dt_soil</i> |
| Accessed in code | as ‘acc_qinsur’ from the ‘diag_physics_noahmp’ pool |

acc_qinsurxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>accumulated water flux into soil</i> |
| Accessed in code | as ‘acc_qinsurxy’ from the ‘output_noahmp’ pool |

acc_qseva (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>accumulated qsevap between dt_soil</i> |
| Accessed in code | as ‘acc_qseva’ from the ‘diag_physics_noahmp’ pool |

acc_qsevaxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m\ s^{-1}$ |
| Description | <i>accumulated soil surface evaporation</i> |
| Accessed in code | as ‘acc_qsevaxy’ from the ‘output_noahmp’ pool |

acc_ssoil (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated ssoil between dt_soil</i> |
| Accessed in code | as ‘acc_ssoil’ from the ‘diag_physics_noahmp’ pool |

acc_ssoilxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}\ s^{-1}$ |
| Description | <i>accumulated ground heat flux</i> |
| Accessed in code | as ‘acc_ssoilxy’ from the ‘output_noahmp’ pool |

aclwdnb (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky downward surface longwave radiation flux</i> |
| Accessed in code | as ‘aclwdnb’ from the ‘diag_physics’ pool |

aclwdnbc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky downward surface longwave radiation flux</i> |
| Accessed in code | as ‘aclwdnbc’ from the ‘diag_physics’ pool |

aclwdnt (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky downward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘aclwdnt’ from the ‘diag_physics’ pool |

aclwdntc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky downward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘aclwdntc’ from the ‘diag_physics’ pool |

aclwupb (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky upward surface longwave radiation flux</i> |
| Accessed in code | as ‘aclwupb’ from the ‘diag_physics’ pool |

aclwupbc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky upward surface longwave radiation flux</i> |
| Accessed in code | as ‘aclwupbc’ from the ‘diag_physics’ pool |

aclwupt (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky upward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘aclwupt’ from the ‘diag_physics’ pool |

aclwuptc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky upward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘aclwuptc’ from the ‘diag_physics’ pool |

acsnom (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}$ |
| Description | <i>accumulated melted snow</i> |
| Accessed in code | as ‘acsnom’ from the ‘diag_physics’ pool |

acsnow (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}$ |
| Description | <i>accumulated snow</i> |
| Accessed in code | as ‘acsnow’ from the ‘diag_physics’ pool |

acswnb (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky downward surface shortwave radiation flux</i> |
| Accessed in code | as ‘acswnb’ from the ‘diag_physics’ pool |

acswnbc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky downward surface shortwave radiation flux</i> |
| Accessed in code | as ‘acswnbc’ from the ‘diag_physics’ pool |

acswdnt (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky downward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘acswdnt’ from the ‘diag_physics’ pool |

acswdntc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky downward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘acswdntc’ from the ‘diag_physics’ pool |

acsuwupb (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky upward surface shortwave radiation flux</i> |
| Accessed in code | as ‘acsuwupb’ from the ‘diag_physics’ pool |

acsuwupbc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky upward surface shortwave radiation flux</i> |
| Accessed in code | as ‘acsuwupbc’ from the ‘diag_physics’ pool |

acswuupt (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated all-sky upward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘acswuupt’ from the ‘diag_physics’ pool |

acswuuptc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>accumulated clear-sky upward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘acswuuptc’ from the ‘diag_physics’ pool |

adv_coefs (real) (FIFTEEN, nEdges)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Weighting coefficients used for reconstructing cell-based fields at edges</i> |
| Accessed in code | as ‘adv_coefs’ from the ‘mesh’ pool |

adv_coefs_3rd (real) (FIFTEEN, nEdges)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Weighting coefficients used for reconstructing cell-based fields at edges</i> |
| Accessed in code | as ‘adv_coefs_3rd’ from the ‘mesh’ pool |

advCellsForEdge (integer) (FIFTEEN, nEdges)

| | |
|------------------|--|
| Units | - |
| Description | <i>Cells used to reconstruct a cell-based field at an edge</i> |
| Accessed in code | as ‘advCellsForEdge’ from the ‘mesh’ pool |

aerosols (real) (nAerLevels, nCells, Time)

| | |
|------------------|--|
| Description | 'aerosols' is a variable array used in code which is made up of the constituent fields listed below. |
| Accessed in code | as 'aerosols' from the 'state' pool |

Contains constituents:

sul — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Sulfate soluble (SUL) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

sslt — constituent field of var_array **aerosols**

| | |
|-------------|---------------------------------------|
| Description | Sea-salt (SSLT) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

dust1 — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Dust type 1 (DUST1) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

dust2 — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Dust type 2 (DUST2) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

dust3 — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Dust type 3 (DUST3) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

dust4 — constituent field of var_array **aerosols**

| | |
|-------------|--|
| Description | Dust type 4 (<i>DUST4</i>) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

ocpho — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Hydrophobic organic carbon (<i>OCPHO</i>) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

bcpho — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Hydrophobic black carbon (<i>BCPHO</i>) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

ocphi — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Hydrophilic organic carbon (<i>OCPHI</i>) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

bcphi — constituent field of var_array **aerosols**

| | |
|-------------|---|
| Description | Hydrophilic black carbon (<i>BCPHI</i>) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

bg — constituent field of var_array **aerosols**

| | |
|-------------|--|
| Description | Background (<i>BG</i>) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

volc — constituent field of var_array **aerosols**

| | |
|-------------|--|
| Description | Volcanic (<i>VOLC</i>) aerosol concentration |
| Units | $kg\ m^{-3}$ |
| Array Group | aer_cam |

albedo12m (real) (nMonths, nCells)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>monthly-mean climatological surface albedo</i> |
| Accessed in code | as ‘albedo12m’ from the ‘sfc_input’ pool |

alboldxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>snow albedo at last timestep</i> |
| Accessed in code | as ‘alboldxy’ from the ‘diag_physics_noahmp’ pool |

alpha_trí (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>implicit tridiagonal solve coefficients</i> |
| Accessed in code | as ‘alpha_trí’ from the ‘diag’ pool |

angleEdge (real) (nEdges)

| | |
|------------------|---|
| Units | <i>rad</i> |
| Description | <i>Angle between local north and the positive tangential direction of an edge</i> |
| Accessed in code | as ‘angleEdge’ from the ‘mesh’ pool |

aparxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>W m⁻²</i> |
| Description | <i>iphotosynthesis active energy by canopy</i> |
| Accessed in code | as ‘aparxy’ from the ‘output_noahmp’ pool |

areaCell (real) (nCells)

| | |
|------------------|---|
| Units | <i>m²</i> |
| Description | <i>Spherical area of a Voronoi cell</i> |
| Accessed in code | as ‘areaCell’ from the ‘mesh’ pool |

areaTriangle (real) (nVertices)

| | |
|------------------|--|
| Units | <i>m²</i> |
| Description | <i>Spherical area of a Delaunay triangle</i> |
| Accessed in code | as ‘areaTriangle’ from the ‘mesh’ pool |

bcphi — *see ‘aerosols’*

bcpho — *see ‘aerosols’*

bdyMaskCell (integer) (nCells)

| | |
|------------------|---|
| Units | - |
| Description | <i>Limited-area specified/relaxation zone index for cells</i> |
| Accessed in code | as ‘bdyMaskCell’ from the ‘mesh’ pool |

bdyMaskEdge (integer) (nEdges)

| | |
|------------------|---|
| Units | - |
| Description | <i>Limited-area specified/relaxation zone index for edges</i> |
| Accessed in code | as ‘bdyMaskEdge’ from the ‘mesh’ pool |

bdyMaskVertex (integer) (nVertices)

| | |
|------------------|--|
| Units | - |
| Description | <i>Limited-area specified/relaxation zone index for vertices</i> |
| Accessed in code | as ‘bdyMaskVertex’ from the ‘mesh’ pool |

bg — *see ‘aerosols’*

bgapxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>between canopy gap</i> |
| Accessed in code | as ‘bgapxy’ from the ‘output_noahmp’ pool |

br (real) (nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Richardson number</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘br’ from the ‘diag_physics’ pool |

brtemp (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>brightness temperature to calculate cloud top height; usually from geostationary IR window channel</i> |
| Allocated by | jedi_da |
| Accessed in code | as ‘brtemp’ from the ‘diag’ pool |

canhsxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>canopy heat storage change due to canopy temperature change</i> |
| Accessed in code | as ‘canhsxy’ from the ‘output_noahmp’ pool |

canicexy (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>intercepted ice mass</i> |
| Accessed in code | as ‘canicexy’ from the ‘diag_physics_noahmp’ pool |

canliqxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>intercepted liquid water</i> |
| Accessed in code | as ‘canliqxy’ from the ‘diag_physics_noahmp’ pool |

canwat (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg m^{-2}$ |
| Description | <i>water in canopy</i> |
| Accessed in code | as ‘canwat’ from the ‘diag_physics’ pool |

cape (real) (nCells, Time)

| | |
|------------------|--|
| Units | $J kg^{-1}$ |
| Description | <i>Convective available potential energy</i> |
| Accessed in code | as ‘cape’ from the ‘diag’ pool |

cd (real) (nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>drag coefficient at 10-meter</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘cd’ from the ‘diag_physics’ pool |

cda (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>drag coefficient at lowest model level</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘cda’ from the ‘diag_physics’ pool |

cellsOnCell (integer) (maxEdges, nCells)

| | |
|------------------|--|
| Units | - |
| Description | <i>IDs of cells neighboring a cell</i> |
| Accessed in code | as ‘cellsOnCell’ from the ‘mesh’ pool |

cellsOnEdge (integer) (TWO, nEdges)

| | |
|------------------|--|
| Units | - |
| Description | <i>IDs of cells divided by an edge</i> |
| Accessed in code | as ‘cellsOnEdge’ from the ‘mesh’ pool |

cellsOnVertex (integer) (vertexDegree, nVertices)

| | |
|------------------|---|
| Units | - |
| Description | <i>IDs of the cells that meet at a vertex</i> |
| Accessed in code | as ‘cellsOnVertex’ from the ‘mesh’ pool |

cellTangentPlane (real) (R3, TWO, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Components of a pair of vectors defining the tangent plane at a cell</i> |
| Accessed in code | as ‘cellTangentPlane’ from the ‘mesh’ pool |

cf1 (real) ()

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Surface interpolation weight for level k=1 value</i> |
| Accessed in code | as ‘cf1’ from the ‘mesh’ pool |

cf2 (real) ()

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Surface interpolation weight for level k=2 value</i> |
| Accessed in code | as ‘cf2’ from the ‘mesh’ pool |

cf3 (real) ()

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Surface interpolation weight for level k=3 value</i> |
| Accessed in code | as ‘cf3’ from the ‘mesh’ pool |

ch (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m\ s^{-1}$ |
| Description | <i>surface exchange coefficient for heat</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘ch’ from the ‘diag_physics’ pool |

chb2xy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>under canopy exchange coefficient</i> |
| Accessed in code | as ‘chb2xy’ from the ‘output_noahmp’ pool |

chbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $s\ m^{-1}$ |
| Description | <i>bare-ground heat exchange coefficient</i> |
| Accessed in code | as ‘chbxy’ from the ‘output_noahmp’ pool |

chklowq (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>surface saturation flag</i> |
| Accessed in code | as ‘chklowq’ from the ‘diag_physics’ pool |

chleafxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>leaf exchange coefficient</i> |
| Accessed in code | as ‘chleafxy’ from the ‘output_noahmp’ pool |

chs (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>surface exchange coefficient for heat and moisture</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘chs’ from the ‘diag_physics’ pool |

chs2 (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>surface exchange coefficient for heat at 2-meter</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘chs2’ from the ‘diag_physics’ pool |

chucxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>under canopy exchange coefficient</i> |
| Accessed in code | as ‘chucxy’ from the ‘output_noahmp’ pool |

chv2xy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>leaf exchange coefficient</i> |
| Accessed in code | as ‘chv2xy’ from the ‘output_noahmp’ pool |

chvxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $s\ m^{-1}$ |
| Description | <i>vegetation heat exchange coefficient</i> |
| Accessed in code | as ‘chvxy’ from the ‘output_noahmp’ pool |

chxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>surface exchange coefficient for heat</i> |
| Accessed in code | as ‘chxy’ from the ‘diag_physics_noahmp’ pool |

cin (real) (nCells, Time)

| | |
|------------------|-------------------------------|
| Units | $J\ kg^{-1}$ |
| Description | <i>Convective inhibition</i> |
| Accessed in code | as ‘cin’ from the ‘diag’ pool |

circulation (real) (nVertLevels, nVertices, Time)

| | |
|------------------|---|
| Units | $m^2\ s^{-1}$ |
| Description | <i>Horizontal circulation at vertices</i> |
| Accessed in code | as ‘circulation’ from the ‘diag’ pool |

ck (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>enthalpy exchange coeff at 10-meter</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘ck’ from the ‘diag_physics’ pool |

cka (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>enthalpy exchange coefficient at lowest model level</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘cka’ from the ‘diag_physics’ pool |

cldfrac (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>horizontal cloud fraction</i> |
| Accessed in code | as ‘cldfrac’ from the ‘diag_physics’ pool |

cldfrac_bl (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>cloud fraction in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘cldfrac_bl’ from the ‘diag_physics’ pool |

cldfrac_high_UPP (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Cloud fraction at high levels using UPP method</i> |
| Accessed in code | as ‘cldfrac_high_UPP’ from the ‘diag’ pool |

cldfrac_low_UPP (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Cloud fraction at low levels using UPP method</i> |
| Accessed in code | as ‘cldfrac_low_UPP’ from the ‘diag’ pool |

cldfrac_mid_UPP (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Cloud fraction at mid levels using UPP method</i> |
| Accessed in code | as ‘cldfrac_mid_UPP’ from the ‘diag’ pool |

cldfrac_tot_UPP (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Total cloud fraction using UPP column max method</i> |
| Accessed in code | as ‘cldfrac_tot_UPP’ from the ‘diag’ pool |

cldmask (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>cloud mask (1=cloudy ; 0=clear)</i> |
| Allocated by | jedi_da |
| Accessed in code | as ‘cldmask’ from the ‘diag’ pool |

cmxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>surface exchange coefficient for momentum</i> |
| Accessed in code | as ‘cmxy’ from the ‘diag_physics_noahmp’ pool |

coeffs_reconstruct (real) (R3, maxEdges, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Coefficients to reconstruct velocity vectors at cell centers</i> |
| Accessed in code | as ‘coeffs_reconstruct’ from the ‘mesh’ pool |

cofrz (real) (nVertLevels, Time)

| | |
|------------------|---|
| Units | $s m^{-1}$ |
| Description | <i>coefficient for implicit contribution of Omega to density update</i> |
| Accessed in code | as ‘cofrz’ from the ‘diag’ pool |

coftz (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|---|
| Units | $s K$ |
| Description | <i>coefficient for implicit contribution of omega vertical derivative to the theta_m update</i> |
| Accessed in code | as ‘coftz’ from the ‘diag’ pool |

cofwr (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>coefficient for implicit contribution of density to the vertical velocity update</i> |
| Accessed in code | as ‘cofwr’ from the ‘diag’ pool |

cofw (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1} K^{-1}$ |
| Description | <i>coefficient for implicit contribution of density to the vertical velocity update</i> |
| Accessed in code | as ‘cofw’ from the ‘diag’ pool |

cofwz (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1} K^{-1}$ |
| Description | <i>coefficient for implicit contribution of density to the vertical velocity update</i> |
| Accessed in code | as ‘cofwz’ from the ‘diag’ pool |

con (real) (nCells)

| | |
|------------------|------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>orographic convexity</i> |
| Accessed in code | as ‘con’ from the ‘sfc_input’ pool |

conls (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>orographic convexity (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘conls’ from the ‘sfc_input’ pool |

conss (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>orographic convexity (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘conss’ from the ‘sfc_input’ pool |

coszr (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>cosine of zenith solar angle</i> |
| Accessed in code | as ‘coszr’ from the ‘diag_physics’ pool |

cov (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K \text{ kg kg}^{-1}$ |
| Description | <i>liquid water - liquid water potential temperature covariance</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘cov’ from the ‘diag_physics’ pool |

cpm (real) (nCells, Time)

| | |
|------------------|--|
| Units | $J K^{-1} \text{ kg}^{-1}$ |
| Description | <i>specific heat of dry air at constant pressure at lowest model level</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘cpm’ from the ‘diag_physics’ pool |

cqs2 (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>surface exchange coefficient for moisture at 2-meter</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘cqs2’ from the ‘diag_physics’ pool |

cqu (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>rho_d/rho_m at cell edge (u points)</i> |
| Accessed in code | as ‘cqu’ from the ‘diag’ pool |

cqw (real) (nVertLevels, nCells, Time)

| | |
|------------------|--------------------------------|
| Units | <i>unitless</i> |
| Description | <i>rho_d/rho_m at w points</i> |
| Accessed in code | as ‘cqw’ from the ‘diag’ pool |

cubot (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>index of highest level of convection</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in |
| Accessed in code | as ‘cubot’ from the ‘diag_physics’ pool |

cuprec (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>convective precipitation rate</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as ‘cuprec’ from the ‘diag_physics’ pool |

cutop (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>index of lowest level of convection</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in |
| Accessed in code | as ‘cutop’ from the ‘diag_physics’ pool |

DAYs (real) (days)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>day of year in the input ugwp_limb_tau NGW file</i> |
| Allocated by | ugwp_ngw_stream |
| Accessed in code | as ‘DAYs’ from the ‘ngw_input’ pool |

dcEdge (real) (nEdges)

| | |
|------------------|--|
| Units | <i>m</i> |
| Description | <i>Spherical distance between cells separated by an edge</i> |
| Accessed in code | as ‘dcEdge’ from the ‘mesh’ pool |

ddy_j1tau (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>latitude interpolation weight complement for absolute momentum flux due to nonorographic gravity wave drag</i> |
| Allocated by | ugwp_ngw_stream |
| Accessed in code | as ‘ddy_j1tau’ from the ‘ngw_input’ pool |

ddy_j2tau (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>latitude interpolation weight for absolute momentum flux due to nonorographic gravity wave drag</i> |
| Allocated by | ugwp_ngw_stream |
| Accessed in code | as ‘ddy_j2tau’ from the ‘ngw_input’ pool |

deeprechxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>deep water table recharge</i> |
| Accessed in code | as ‘deeprechxy’ from the ‘diag_physics_noahmp’ pool |

defc_a (real) (maxEdges, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Coefficients for computing the off-diagonal components of the horizontal deformation</i> |
| Accessed in code | as ‘defc_a’ from the ‘mesh’ pool |

defc_b (real) (maxEdges, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Coefficients for computing the diagonal components of the horizontal deformation</i> |
| Accessed in code | as ‘defc_b’ from the ‘mesh’ pool |

delta (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>m</i> |
| Description | <i>entrainment layer depth from PBL scheme</i> |
| Allocated by | bl_ysu_in |
| Accessed in code | as ‘delta’ from the ‘diag_physics’ pool |

depv_dt_bl (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | <i>PVU s⁻¹</i> |
| Description | <i>Diabatic EPV tendency from PBL</i> |
| Accessed in code | as ‘depv_dt_bl’ from the ‘diag’ pool |

depv_dt_cu (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Diabatic EPV tendency from convection</i> |
| Accessed in code | as ‘depv_dt_cu’ from the ‘diag’ pool |

depv_dt_diab (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Sum of calculated EPV tendencies from diabatic processes</i> |
| Accessed in code | as ‘depv_dt_diab’ from the ‘diag’ pool |

depv_dt_diab_pv (real) (nCells, Time)

| | |
|------------------|--|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Diabatic EPV tendency on dynamic tropopause</i> |
| Accessed in code | as ‘depv_dt_diab_pv’ from the ‘diag’ pool |

depv_dt_fric (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Sum of calculated EPV tendencies from frictional processes</i> |
| Accessed in code | as ‘depv_dt_fric’ from the ‘diag’ pool |

depv_dt_fric_pv (real) (nCells, Time)

| | |
|------------------|--|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Frictional EPV tendency on dynamic tropopause</i> |
| Accessed in code | as ‘depv_dt_fric_pv’ from the ‘diag’ pool |

depv_dt_lw (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Diabatic EPV tendency from longwave radiation</i> |
| Accessed in code | as ‘depv_dt_lw’ from the ‘diag’ pool |

depv_dt_mix (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Diabatic EPV tendency from explicit numerical mixing</i> |
| Accessed in code | as ‘depv_dt_mix’ from the ‘diag’ pool |

depv_dt_mp (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Diabatic EPV tendency from microphysics</i> |
| Accessed in code | as ‘depv_dt_mp’ from the ‘diag’ pool |

depv_dt_sw (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $PVU\ s^{-1}$ |
| Description | <i>Diabatic EPV tendency from shortwave radiation</i> |
| Accessed in code | as ‘depv_dt_sw’ from the ‘diag’ pool |

deriv_two (real) (FIFTEEN, TWO, nEdges)

| | |
|------------------|--|
| Units | unitless |
| Description | <i>weights for cell-centered second derivative, normal to edge, for transport scheme</i> |
| Accessed in code | as ‘deriv_two’ from the ‘mesh’ pool |

det_qv (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>EDMF detrainment of water vapor mixing ratio in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘det_qv’ from the ‘diag_physics’ pool |

det_thl (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $K\ s^{-1}$ |
| Description | <i>EDMF detrainment of liquid-ice potential temperature in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘det_thl’ from the ‘diag_physics’ pool |

dewpoint_100hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 100 hPa</i> |
| Accessed in code | as ‘dewpoint_100hPa’ from the ‘diag’ pool |

dewpoint_200hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 200 hPa</i> |
| Accessed in code | as ‘dewpoint_200hPa’ from the ‘diag’ pool |

dewpoint_250hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 250 hPa</i> |
| Accessed in code | as ‘dewpoint_250hPa’ from the ‘diag’ pool |

dewpoint_500hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 500 hPa</i> |
| Accessed in code | as ‘dewpoint_500hPa’ from the ‘diag’ pool |

dewpoint_50hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 50 hPa</i> |
| Accessed in code | as ‘dewpoint_50hPa’ from the ‘diag’ pool |

dewpoint_700hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 700 hPa</i> |
| Accessed in code | as ‘dewpoint_700hPa’ from the ‘diag’ pool |

dewpoint_850hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 850 hPa</i> |
| Accessed in code | as ‘dewpoint_850hPa’ from the ‘diag’ pool |

dewpoint_925hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Dewpoint temperature vertically interpolated to 925 hPa</i> |
| Accessed in code | as ‘dewpoint_925hPa’ from the ‘diag’ pool |

dewpoint_surface (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Dewpoint temperature at midpoint of lowest model layer</i> |
| Accessed in code | as ‘dewpoint_surface’ from the ‘diag’ pool |

divergence (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Horizontal velocity divergence at cell center</i> |
| Accessed in code | as ‘divergence’ from the ‘diag’ pool |

dqke (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-2}$ |
| Description | <i>TKE change</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘dqke’ from the ‘diag_physics’ pool |

dss (real) (nVertLevels, nCells)

| | |
|------------------|-------------------------------|
| Units | <i>unitless</i> |
| Description | <i>w-damping coefficient</i> |
| Accessed in code | as ‘dss’ from the ‘mesh’ pool |

dtaux3d (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>gravity wave drag over orography u-stress</i> |
| Accessed in code | as ‘dtaux3d’ from the ‘diag_physics’ pool |

dtaux3d _ bl (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>orog blocking drag u-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtaux3d _ bl’ from the ‘diag_physics’ pool |

dtaux3d _ fd (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>turb orog form drag u-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtaux3d _ fd’ from the ‘diag_physics’ pool |

dtaux3d _ ls (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>mesoscale orog gravity wave drag u-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtaux3d _ ls’ from the ‘diag_physics’ pool |

dtaux3d _ ss (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>small-scale orog gravity wave drag u-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtaux3d _ ss’ from the ‘diag_physics’ pool |

dtauy3d (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>gravity wave drag over orography v-stress</i> |
| Accessed in code | as ‘dtauy3d’ from the ‘diag_physics’ pool |

dtauy3d _ bl (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>orog blocking drag v-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtauy3d _ bl’ from the ‘diag_physics’ pool |

dtauy3d _ fd (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>turb orog form drag v-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtauy3d _ fd’ from the ‘diag_physics’ pool |

dtauy3d _ ls (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>mesoscale orog gravity wave drag v-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtauy3d _ ls’ from the ‘diag_physics’ pool |

dtauy3d _ ss (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>small-scale orog gravity wave drag v-tendency</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dtauy3d _ ss’ from the ‘diag_physics’ pool |

dtdt _ ngw (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K s^{-1}$ |
| Description | <i>temperature tendency due to non-stationary gravity wave drag</i> |
| Allocated by | ugwp_ngw_stream, ugwp_diags_stream |
| Accessed in code | as ‘dtdt _ ngw’ from the ‘diag_physics’ pool |

dtheta _ dt _ mix (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $K s^{-1}$ |
| Description | <i>Potential temperature heating rate from explicit numerical mixing</i> |
| Accessed in code | as ‘dtheta _ dt _ mix’ from the ‘diag’ pool |

dtheta_dt_mp (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K s^{-1}$ |
| Description | <i>Potential temperature heating rate from microphysics</i> |
| Accessed in code | as ‘dtheta_dt_mp’ from the ‘diag’ pool |

dudt_ngw (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>u-momentum tendency due to non-stationary gravity wave drag</i> |
| Allocated by | ugwp_ngw_stream, ugwp_diags_stream |
| Accessed in code | as ‘dudt_ngw’ from the ‘diag_physics’ pool |

dusfc_bl (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>vertically-integrated orog blocking drag u-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dusfc_bl’ from the ‘diag_physics’ pool |

dusfc_fd (real) (nCells, Time)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>vertically-integrated turb orog form drag u-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dusfc_fd’ from the ‘diag_physics’ pool |

dusfc_ls (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>vertically-integrated mesoscale orog gravity wave drag u-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dusfc_ls’ from the ‘diag_physics’ pool |

dusfc_ss (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>vertically-integrated small-scale orog gravity wave drag u-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dusfc_ss’ from the ‘diag_physics’ pool |

dusfcg (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa m s^{-1} |
| Description | <i>vertically-integrated gravity wave drag over orography u-stress</i> |
| Accessed in code | as ‘dusfcg’ from the ‘diag_physics’ pool |

dust1 — see ‘aerosols’**dust2** — see ‘aerosols’**dust3** — see ‘aerosols’**dust4** — see ‘aerosols’**dvdt_ngw** (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | m s^{-2} |
| Description | <i>v-momentum tendency due to non-stationary gravity wave drag</i> |
| Allocated by | ugwp_ngw_stream, ugwp_diags_stream |
| Accessed in code | as ‘dvdt_ngw’ from the ‘diag_physics’ pool |

dvEdge (real) (nEdges)

| | |
|------------------|---|
| Units | m |
| Description | <i>Spherical distance between vertex endpoints of an edge</i> |
| Accessed in code | as ‘dvEdge’ from the ‘mesh’ pool |

dvsfc_bl (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>vertically-integrated orog blocking drag v-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dvsfc_bl’ from the ‘diag_physics’ pool |

dvsfc_fd (real) (nCells, Time)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>vertically-integrated turb orog form drag v-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dvsfc_fd’ from the ‘diag_physics’ pool |

dvsfc_ls (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>vertically-integrated mesoscale orog gravity wave drag v-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dvsfc_ls’ from the ‘diag_physics’ pool |

dvsfc_ss (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>vertically-integrated small-scale orog gravity wave drag v-stress</i> |
| Allocated by | ugwp_diags_stream |
| Accessed in code | as ‘dvsfc_ss’ from the ‘diag_physics’ pool |

dvsfcg (real) (nCells, Time)

| | |
|------------------|--|
| Units | $Pa m s^{-1}$ |
| Description | <i>vertically-integrated gravity wave drag over orography v-stress</i> |
| Accessed in code | as ‘dvsfcg’ from the ‘diag_physics’ pool |

dzs (real) (nSoilLevels, nCells, Time)

| | |
|------------------|------------------------------------|
| Units | m |
| Description | <i>soil layer thickness</i> |
| Accessed in code | as ‘dzs’ from the ‘sfc_input’ pool |

dzu (real) (nVertLevels)

| | |
|------------------|-------------------------------|
| Units | <i>unitless</i> |
| Description | <i>d(zeta) at w levels</i> |
| Accessed in code | as ‘dzu’ from the ‘mesh’ pool |

eahxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>canopy air vapor pressure</i> |
| Accessed in code | as ‘eahxy’ from the ‘diag_physics_noahmp’ pool |

east (real) (R3, nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Cartesian components of the local unit vector pointing east</i> |
| Accessed in code | as ‘east’ from the ‘mesh’ pool |

ecanxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>evaporation of intercepted water</i> |
| Accessed in code | as ‘ecanxy’ from the ‘output_noahmp’ pool |

edgeNormalVectors (real) (R3, nEdges)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Cartesian components of the vector normal to an edge and tangential to the surface of the sphere</i> |
| Accessed in code | as ‘edgeNormalVectors’ from the ‘mesh’ pool |

edgesOnCell (integer) (maxEdges, nCells)

| | |
|------------------|--|
| Units | - |
| Description | <i>IDs of edges forming the boundary of a cell</i> |
| Accessed in code | as ‘edgesOnCell’ from the ‘mesh’ pool |

edgesOnCell_sign (real) (maxEdges, nCells)

| | |
|------------------|---|
| Units | - |
| Description | <i>Sign for edges surrounding a cell: positive for positive outward normal velocity</i> |
| Accessed in code | as ‘edgesOnCell_sign’ from the ‘mesh’ pool |

edgesOnEdge (integer) (maxEdges2, nEdges)

| | |
|------------------|---|
| Units | - |
| Description | <i>IDs of edges involved in reconstruction of tangential velocity for an edge</i> |
| Accessed in code | as ‘edgesOnEdge’ from the ‘mesh’ pool |

edgesOnVertex (integer) (vertexDegree, nVertices)

| | |
|------------------|---|
| Units | - |
| Description | <i>IDs of the edges that meet at a vertex</i> |
| Accessed in code | as ‘edgesOnVertex’ from the ‘mesh’ pool |

edgesOnVertex_sign (real) (vertexDegree, nVertices)

| | |
|------------------|--|
| Units | - |
| Description | <i>Sign for edges incident with a vertex: positive for positive inward tangential velocity</i> |
| Accessed in code | as ‘edgesOnVertex_sign’ from the ‘mesh’ pool |

edirxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>ground surface evaporation rate</i> |
| Accessed in code | as ‘edirxy’ from the ‘output_noahmp’ pool |

edmf_a (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>EDMF relative updraft area of moist updrafts in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘edmf_a’ from the ‘diag_physics’ pool |

edmf_ent (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | m^{-1} |
| Description | <i>EDMF entrainment rate in moist updrafts in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘edmf_ent’ from the ‘diag_physics’ pool |

edmf_qc (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}$ |
| Description | <i>EDMF liquid water mixing ratio in moist updrafts in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘edmf_qc’ from the ‘diag_physics’ pool |

edmf_qt (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>EDMF total water mixing ratio in moist updrafts in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘edmf_qt’ from the ‘diag_physics’ pool |

edmf_thl (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>EDMF liquid-ice water potential temperature in moist updrafts in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘edmf_thl’ from the ‘diag_physics’ pool |

edmf_w (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m\ s^{-1}$ |
| Description | <i>EDMF vertical velocity of moist updrafts in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘edmf_w’ from the ‘diag_physics’ pool |

eflxbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>bottom soil heat flux</i> |
| Accessed in code | as ‘eflxbxy’ from the ‘output_noahmp’ pool |

el_pbl (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>mixing length from PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘el_pbl’ from the ‘diag_physics’ pool |

emstot (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>Total emissivity</i> |
| Accessed in code | as ‘emstot’ from the ‘diag_physics’ pool |

ertel_pv (real) (nVertLevels, nCells, Time)

| | |
|------------------|------------------------------------|
| Units | PVU |
| Description | <i>Ertel’s potential vorticity</i> |
| Accessed in code | as ‘ertel_pv’ from the ‘diag’ pool |

etranxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>transpiration rate</i> |
| Accessed in code | as ‘etranxy’ from the ‘output_noahmp’ pool |

euler_tend_theta (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ K\ m^{-3}\ s^{-1}$ |
| Description | <i>tendency of coupled potential temperature rho*theta_m/zz from dynamics and physics that does not change over a Runge-Kutta timestep (it excludes the flux divergence)</i> |
| Accessed in code | as ‘theta_euler’ from the ‘tend’ pool |

euler_tend_u (real) (nVertLevels, nEdges, Time)

| | |
|------------------|------------------------------------|
| Units | $m\ s^{-2}$ |
| Description | <i>Tendency of u from dynamics</i> |
| Accessed in code | as ‘u_euler’ from the ‘tend’ pool |

euler_tend_w (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|------------------------------------|
| Units | $m s^{-2}$ |
| Description | <i>Tendency of w from dynamics</i> |
| Accessed in code | as ‘w_euler’ from the ‘tend’ pool |

evapprod (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | s^{-1} |
| Description | <i>rain evaporation rate</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘evapprod’ from the ‘diag_physics’ pool |

evbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>latent heat flux: bare ground to atmosphere</i> |
| Accessed in code | as ‘evbxy’ from the ‘output_noahmp’ pool |

evcxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>canopy evaporation</i> |
| Accessed in code | as ‘evcxy’ from the ‘output_noahmp’ pool |

evgxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W m^{-2}$ |
| Description | <i>latent heat flux: ground to canopy</i> |
| Accessed in code | as ‘evgxy’ from the ‘output_noahmp’ pool |

exch_h (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-1}$ |
| Description | <i>exchange coefficient</i> |
| Allocated by | bl_ysu_in |
| Accessed in code | as ‘exch_h’ from the ‘diag_physics’ pool |

exner (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Exner function</i> |
| Accessed in code | as ‘exner’ from the ‘diag’ pool |

exner_base (real) (nVertLevels, nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Base-state Exner function</i> |
| Accessed in code | as ‘exner_base’ from the ‘diag’ pool |

fastcpxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $g\ m^{-2}$ |
| Description | <i>short-lived carbon</i> |
| Accessed in code | as ‘fastcpxy’ from the ‘diag_physics_noahmp’ pool |

fEdge (real) (nEdges)

| | |
|------------------|--------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Coriolis parameter at an edge</i> |
| Accessed in code | as ‘fEdge’ from the ‘mesh’ pool |

fh (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>integrated stability function for heat</i> |
| Allocated by | bl_ysu_in |
| Accessed in code | as ‘fh’ from the ‘diag_physics’ pool |

fifract (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>flood irrigation fraction</i> |
| Accessed in code | as ‘fifract’ from the ‘diag_physics_noahmp’ pool |

firaxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>total net longwave radiation</i> |
| Accessed in code | as ‘firaxy’ from the ‘output_noahmp’ pool |

flhc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}\ K^{-1}$ |
| Description | <i>exchange coefficient for heat</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘flhc’ from the ‘diag_physics’ pool |

flqc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>exchange coefficient for moisture</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘flqc’ from the ‘diag_physics’ pool |

fm (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>integrated stability function for moisture</i> |
| Allocated by | bl_ysu_in |
| Accessed in code | as ‘fm’ from the ‘diag_physics’ pool |

forcplsm (real) (nCells, Time)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>lowest model P into LSM</i> |
| Accessed in code | as ‘forcplsm’ from the ‘output_noahmp’ pool |

forcqlsm (real) (nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>lowest model Q into LSM</i> |
| Accessed in code | as ‘forcqlsm’ from the ‘output_noahmp’ pool |

forctlsm (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>lowest model T into LSM</i> |
| Accessed in code | as ‘forctlsm’ from the ‘output_noahmp’ pool |

forcwlsm (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>lowest model wind speed into LSM</i> |
| Accessed in code | as ‘forcwlsm’ from the ‘output_noahmp’ pool |

forczlsm (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>lowest model Z into LSM</i> |
| Accessed in code | as ‘forczlsm’ from the ‘output_noahmp’ pool |

fpicexy (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>fraction of ice in precipitation</i> |
| Accessed in code | as ‘fpicexy’ from the ‘output_noahmp’ pool |

fsaxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>total absorbed solar radiation</i> |
| Accessed in code | as ‘fsaxy’ from the ‘output_noahmp’ pool |

fvegxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>NOAHMP vegetation fraction</i> |
| Accessed in code | as ‘fvegxy’ from the ‘output_noahmp’ pool |

fVertex (real) (nVertices)

| | |
|------------------|---------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Coriolis parameter at a vertex</i> |
| Accessed in code | as ‘fVertex’ from the ‘mesh’ pool |

fwetxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>wetted or snowed canopy fraction</i> |
| Accessed in code | as ‘fwetxy’ from the ‘diag_physics_noahmp’ pool |

fzm (real) (nVertLevels)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Weight for linear interpolation to w(k) point for u(k) level variable</i> |
| Accessed in code | as ‘fzm’ from the ‘mesh’ pool |

fzp (real) (nVertLevels)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Weight for linear interpolation to w(k) point for u(k-1) level variable</i> |
| Accessed in code | as ‘fzp’ from the ‘mesh’ pool |

gamma_trí (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>implicit tridiagonal solve coefficients</i> |
| Accessed in code | as ‘gamma_trí’ from the ‘diag’ pool |

gddxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>growing degree days</i> |
| Accessed in code | as ‘gddxy’ from the ‘diag_physics_noahmp’ pool |

gecros_state (real) (SIXTY, nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>gecros crop</i> |
| Accessed in code | as ‘gecros_state’ from the ‘diag_physics_noahmp’ pool |

ghbxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W m^{-2}$ |
| Description | <i>heat flux into soil: bare fraction</i> |
| Accessed in code | as ‘ghbxy’ from the ‘output_noahmp’ pool |

ghvxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>heat flux into soil: under canopy</i> |
| Accessed in code | as ‘ghvxy’ from the ‘output_noahmp’ pool |

glw (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky downward surface longwave radiation</i> |
| Accessed in code | as ‘glw’ from the ‘diag_physics’ pool |

gppxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $g\ m^{-2}\ s^{-1}\ C$ |
| Description | <i>gross primary productivity</i> |
| Accessed in code | as ‘gppxy’ from the ‘output_noahmp’ pool |

gradPVn (real) (nVertLevels, nEdges, Time)

| | |
|------------------|---|
| Units | ??? |
| Description | <i>Gradient in the normal direction of PV at edge locations</i> |
| Accessed in code | as ‘gradPVn’ from the ‘diag’ pool |

gradPVt (real) (nVertLevels, nEdges, Time)

| | |
|------------------|---|
| Units | ??? |
| Description | <i>Gradient in the tangential direction of PV at edge locations</i> |
| Accessed in code | as ‘gradPVt’ from the ‘diag’ pool |

grainxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $g\ m^{-2}$ |
| Description | <i>mass of grain</i> |
| Accessed in code | as ‘grainxy’ from the ‘diag_physics_noahmp’ pool |

graupelnc (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>accumulated grid-scale precipitation of graupel</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘graupelnc’ from the ‘diag_physics’ pool |

graupelncv (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>time-step grid-scale precipitation of graupel</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘graupelncv’ from the ‘diag_physics’ pool |

grdflux (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>W m⁻²</i> |
| Description | <i>ground heat flux</i> |
| Accessed in code | as ‘grdflux’ from the ‘diag_physics’ pool |

greenfrac (real) (nMonths, nCells)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>monthly-mean climatological greenness fraction</i> |
| Accessed in code | as ‘greenfrac’ from the ‘sfc_input’ pool |

gsw (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>W m⁻²</i> |
| Description | <i>net surface shortwave radiation flux</i> |
| Accessed in code | as ‘gsw’ from the ‘diag_physics’ pool |

gz1oz0 (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>log(z/z0) where z0 is roughness length</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘gz1oz0’ from the ‘diag_physics’ pool |

h_divergence (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | ??? |
| Description | ??? |
| Accessed in code | as 'h_divergence' from the 'diag' pool |

h_oml (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>ocean mixed layer depth</i> |
| Accessed in code | as 'h_oml' from the 'diag_physics' pool |

h_oml_initial (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>ocean mixed layer depth at initial time</i> |
| Accessed in code | as 'h_oml_initial' from the 'diag_physics' pool |

height_100hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Geometric height interpolated to 100 hPa</i> |
| Accessed in code | as 'height_100hPa' from the 'diag' pool |

height_200hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Geometric height interpolated to 200 hPa</i> |
| Accessed in code | as 'height_200hPa' from the 'diag' pool |

height_250hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Geometric height interpolated to 250 hPa</i> |
| Accessed in code | as 'height_250hPa' from the 'diag' pool |

height _ 500hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Geometric height interpolated to 500 hPa</i> |
| Accessed in code | as ‘height _ 500hPa’ from the ‘diag’ pool |

height _ 50hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>Geometric height interpolated to 50 hPa</i> |
| Accessed in code | as ‘height _ 50hPa’ from the ‘diag’ pool |

height _ 700hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Geometric height interpolated to 700 hPa</i> |
| Accessed in code | as ‘height _ 700hPa’ from the ‘diag’ pool |

height _ 850hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Geometric height interpolated to 850 hPa</i> |
| Accessed in code | as ‘height _ 850hPa’ from the ‘diag’ pool |

height _ 925hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Geometric height interpolated to 925 hPa</i> |
| Accessed in code | as ‘height _ 925hPa’ from the ‘diag’ pool |

hfx (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>upward heat flux at the surface</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘hfx’ from the ‘diag_physics’ pool |

hpbl (real) (nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>Planetary Boundary Layer (PBL) height</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘hpbl’ from the ‘diag_physics’ pool |

hu_oml (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-1}$ |
| Description | <i>ocean mixed layer integrated u (zonal velocity)</i> |
| Accessed in code | as ‘hu_oml’ from the ‘diag_physics’ pool |

hv_oml (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m^2 s^{-1}$ |
| Description | <i>ocean mixed layer integrated v (meridional velocity)</i> |
| Accessed in code | as ‘hv_oml’ from the ‘diag_physics’ pool |

i_aclwdnb (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky downward surface longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwdnb’ from the ‘diag_physics’ pool |

i_aclwdnbc (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky downward surface longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwdnbc’ from the ‘diag_physics’ pool |

i_aclwdnt (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky downward top-of-atmosphere longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwdnt’ from the ‘diag_physics’ pool |

i_aclwdntc (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky downward top-of-atmosphere longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwdntc’ from the ‘diag_physics’ pool |

i_aclwupb (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky upward surface longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwupb’ from the ‘diag_physics’ pool |

i_aclwupbc (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky upward surface longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwupbc’ from the ‘diag_physics’ pool |

i_aclwupt (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky upward top-of-atmosphere longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwupt’ from the ‘diag_physics’ pool |

i_aclwuptc (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky upward top-of-atmosphere longwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_aclwuptc’ from the ‘diag_physics’ pool |

i_acswdn (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky downward surface shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswdn’ from the ‘diag_physics’ pool |

i_acswdnbc (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky downward surface shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswdnbc’ from the ‘diag_physics’ pool |

i_acswdnt (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky downward top-of-atmosphere shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswdnt’ from the ‘diag_physics’ pool |

i_acswdntc (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky downward top-of-atmosphere shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswdntc’ from the ‘diag_physics’ pool |

i_acswupb (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky upward surface shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswupb’ from the ‘diag_physics’ pool |

i_acswupbc (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky upward surface shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswupbc’ from the ‘diag_physics’ pool |

i_acswupt (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated all-sky upward top-of-atmosphere shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswupt’ from the ‘diag_physics’ pool |

i_acswuptc (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated clear-sky upward top-of-atmosphere shortwave radiation greater than config_bucket_radt</i> |
| Accessed in code | as ‘i_acswuptc’ from the ‘diag_physics’ pool |

i_rainc (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated convective precipitation greater than config_bucket_rainc</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as ‘i_rainc’ from the ‘diag_physics’ pool |

i_rainnc (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>incidence of accumulated grid-scale precipitation greater than config_bucket_rainnc</i> |
| Allocated by | mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘i_rainnc’ from the ‘diag_physics’ pool |

iLev_DT (integer) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>Lowest vertical level at or above dynamic tropopause (.lt.1 if 2 PVU below column; .gt.nLevels if 2PVU above column)</i> |
| Accessed in code | as ‘iLev_DT’ from the ‘diag’ pool |

indexToCellID (integer) (nCells)

| | |
|------------------|---|
| Units | - |
| Description | <i>Mapping from local array index to global cell ID</i> |
| Accessed in code | as ‘indexToCellID’ from the ‘mesh’ pool |

indexToEdgeID (integer) (nEdges)

| | |
|------------------|---|
| Units | - |
| Description | <i>Mapping from local array index to global edge ID</i> |
| Accessed in code | as ‘indexToEdgeID’ from the ‘mesh’ pool |

indexToVertexID (integer) (nVertices)

| | |
|------------------|---|
| Units | - |
| Description | <i>Mapping from local array index to global vertex ID</i> |
| Accessed in code | as ‘indexToVertexID’ from the ‘mesh’ pool |

initial_time (text) ()

| | |
|------------------|---|
| Units | YYYY-MM-DD hh:mm:ss |
| Description | <i>Model initialization time</i> |
| Accessed in code | as ‘initial_time’ from the ‘state’ pool |

inTropo (integer) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | 0/1 |
| Description | <i>1 if within troposphere based on EPV flood fill</i> |
| Accessed in code | as ‘inTropo’ from the ‘diag’ pool |

invAreaCell (real) (nCells)

| | |
|------------------|---------------------------------------|
| Units | m^{-2} |
| Description | <i>Inverse of Voronoi cell area</i> |
| Accessed in code | as ‘invAreaCell’ from the ‘mesh’ pool |

invAreaTriangle (real) (nVertices)

| | |
|------------------|--|
| Units | m^{-2} |
| Description | <i>Inverse area of a Delaunay triangle</i> |
| Accessed in code | as ‘invAreaTriangle’ from the ‘mesh’ pool |

invDcEdge (real) (nEdges)

| | |
|------------------|--|
| Units | m^{-1} |
| Description | <i>Inverse distance between cells separated by an edge</i> |
| Accessed in code | as ‘invDcEdge’ from the ‘mesh’ pool |

invDvEdge (real) (nEdges)

| | |
|------------------|---|
| Units | m^{-1} |
| Description | <i>Inverse distance between vertex endpoints of an edge</i> |
| Accessed in code | as ‘invDvEdge’ from the ‘mesh’ pool |

irbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>net longwave at bare fraction surface</i> |
| Accessed in code | as ‘irbxy’ from the ‘output_noahmp’ pool |

ircxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>net longwave in canopy</i> |
| Accessed in code | as ‘ircxy’ from the ‘output_noahmp’ pool |

ireloss (real) (nCells, Time)

| | |
|------------------|--|
| Units | mm |
| Description | <i>spinkler evaporation loss accumulated</i> |
| Accessed in code | as ‘ireloss’ from the ‘diag_physics_noahmp’ pool |

irfivol (real) (nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>flood irrigation water accumulated</i> |
| Accessed in code | as ‘irfivol’ from the ‘diag_physics_noahmp’ pool |

irfract (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>irrigation fraction</i> |
| Accessed in code | as ‘irfract’ from the ‘diag_physics_noahmp’ pool |

irgxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>net longwave at below canopy surface</i> |
| Accessed in code | as ‘irgxy’ from the ‘output_noahmp’ pool |

irmivol (real) (nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>micro irrigation water accumulated</i> |
| Accessed in code | as ‘irmivol’ from the ‘diag_physics_noahmp’ pool |

irnumfi (integer) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>flood irrigation event count</i> |
| Accessed in code | as ‘irnumfi’ from the ‘diag_physics_noahmp’ pool |

irnummi (integer) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>micro irrigation event count</i> |
| Accessed in code | as ‘irnummi’ from the ‘diag_physics_noahmp’ pool |

irnumsi (integer) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>sprinkler irrigation event count</i> |
| Accessed in code | as ‘irnumsi’ from the ‘diag_physics_noahmp’ pool |

irrsplh (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>Joules m⁻²</i> |
| Description | <i>sprinkler evaporation loss accumulated</i> |
| Accessed in code | as ‘irrsplh’ from the ‘diag_physics_noahmp’ pool |

irsivol (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>sprinkler irrigation water accumulated</i> |
| Accessed in code | as ‘irsivol’ from the ‘diag_physics_noahmp’ pool |

irwatfi (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>m</i> |
| Description | <i>flood irrigation amount for the event</i> |
| Accessed in code | as ‘irwatfi’ from the ‘diag_physics_noahmp’ pool |

irwatmi (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>m</i> |
| Description | <i>micro irrigation amount for the event</i> |
| Accessed in code | as ‘irwatmi’ from the ‘diag_physics_noahmp’ pool |

irwatsi (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>m</i> |
| Description | <i>sprinkler irrigation amount for the event</i> |
| Accessed in code | as ‘irwatsi’ from the ‘diag_physics_noahmp’ pool |

isice_lu (integer) ()

| | |
|------------------|--------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Index category for snow/ice</i> |
| Accessed in code | as ‘isice’ from the ‘sfc_input’ pool |

isltyp (integer) (nCells)

| | |
|------------------|---------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>dominant soil category</i> |
| Accessed in code | as ‘isltyp’ from the ‘sfc_input’ pool |

isnowxy (integer) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>number of snow layers</i> |
| Accessed in code | as ‘isnowxy’ from the ‘diag_physics_noahmp’ pool |

iswater_lu (integer) ()

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Index category for water</i> |
| Accessed in code | as ‘iswater’ from the ‘sfc_input’ pool |

ivgtyp (integer) (nCells)

| | |
|------------------|---------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>dominant vegetation category</i> |
| Accessed in code | as ‘ivgtyp’ from the ‘sfc_input’ pool |

jindx1_tau (integer) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>lower latitude index of absolute momentum flux due to nonorographic gravity wave drag for interpolation</i> |
| Allocated by | ugwp_ngw_stream |
| Accessed in code | as ‘jindx1_tau’ from the ‘ngw_input’ pool |

jindx2_tau (integer) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>upper latitude index of absolute momentum flux due to nonorographic gravity wave drag for interpolation</i> |
| Allocated by | ugwp_ngw_stream |
| Accessed in code | as ‘jindx2_tau’ from the ‘ngw_input’ pool |

k22_shallow (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>index of convection at k22 level in Grell-Freitas convection scheme</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘k22_shallow’ from the ‘diag_physics’ pool |

kbcon_shallow (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>index of lowest level of shallow convection in Grell-Freitas convection scheme</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘kbcon_shallow’ from the ‘diag_physics’ pool |

kdiff (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-1}$ |
| Description | <i>Smagorinsky horizontal eddy viscosity</i> |
| Accessed in code | as ‘kdiff’ from the ‘diag’ pool |

ke (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-2}$ |
| Description | <i>Kinetic energy at a cell center</i> |
| Accessed in code | as ‘ke’ from the ‘diag’ pool |

kiteAreasOnVertex (real) (vertexDegree, nVertices)

| | |
|------------------|---|
| Units | m^2 |
| Description | <i>Intersection areas between primal (Voronoi) and dual (triangular) mesh cells</i> |
| Accessed in code | as ‘kiteAreasOnVertex’ from the ‘mesh’ pool |

kiteForCell (integer) (maxEdges, nCells)

| | |
|------------------|--|
| Units | - |
| Description | <i>Index of kite in kiteAreasOnVertex that lies within a cell for each of verticesOnCell</i> |
| Accessed in code | as ‘kiteForCell’ from the ‘mesh’ pool |

kpbl (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>index level of PBL top</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘kpbl’ from the ‘diag_physics’ pool |

ktop_deep (integer) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>index of highest level of deep convection in Grell-Freitas convection scheme</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘ktop_deep’ from the ‘diag_physics’ pool |

ktop_shallow (integer) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>index of highest level of shallow convection in Grell-Freitas convection scheme</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘ktop_shallow’ from the ‘diag_physics’ pool |

kzh (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-1}$ |
| Description | <i>vertical diffusion coefficient of potential temperature</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘kzh’ from the ‘diag_physics’ pool |

kzm (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-1}$ |
| Description | <i>vertical diffusion coefficient of mommentum</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘kzm’ from the ‘diag_physics’ pool |

kzq (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-1}$ |
| Description | <i>vertical diffusion coefficient of water vapor and cloud condensates</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘kzq’ from the ‘diag_physics’ pool |

lai (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $m^{-2} m^{-2}$ |
| Description | <i>leaf area index</i> |
| Accessed in code | as ‘lai’ from the ‘diag_physics’ pool |

landmask (integer) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>land-ocean mask (1=land ; 0=ocean)</i> |
| Accessed in code | as ‘landmask’ from the ‘sfc_input’ pool |

latCell (real) (nCells)

| | |
|------------------|-----------------------------------|
| Units | <i>rad</i> |
| Description | <i>Latitude of cells</i> |
| Accessed in code | as ‘latCell’ from the ‘mesh’ pool |

latEdge (real) (nEdges)

| | |
|------------------|-----------------------------------|
| Units | <i>rad</i> |
| Description | <i>Latitude of edges</i> |
| Accessed in code | as ‘latEdge’ from the ‘mesh’ pool |

LATS (real) (lat)

| | |
|------------------|---|
| Units | <i>degrees</i> |
| Description | <i>latitude in the input ugwp_limb_tau NGW file</i> |
| Allocated by | ugwp_ngw_stream |
| Accessed in code | as ‘LATS’ from the ‘ngw_input’ pool |

latVertex (real) (nVertices)

| | |
|------------------|-------------------------------------|
| Units | <i>rad</i> |
| Description | <i>Latitude of vertices</i> |
| Accessed in code | as ‘latVertex’ from the ‘mesh’ pool |

lbc_nc — see ‘*lbc_scalars*’**lbc_ni** — see ‘*lbc_scalars*’**lbc_nifa** — see ‘*lbc_scalars*’**lbc_nr** — see ‘*lbc_scalars*’**lbc_nwfa** — see ‘*lbc_scalars*’**lbc_qc** — see ‘*lbc_scalars*’

lbc_qg — see ‘*lbc_scalars*’

lbc_qi — see ‘*lbc_scalars*’

lbc_qr — see ‘*lbc_scalars*’

lbc_qs — see ‘*lbc_scalars*’

lbc_qv — see ‘*lbc_scalars*’

lbc_rho (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Description | <i>Lateral boundary tendency of rho</i> |
| Accessed in code | as ‘lbc_rho’ from the ‘lbc’ pool |

lbc_rho_edge (real) (nVertLevels, nEdges, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Description | <i>Lateral boundary tendency of rho_zz averaged from cell centers to cell faces (edges)</i> |
| Accessed in code | as ‘lbc_rho_edge’ from the ‘lbc’ pool |

lbc_rho_zz (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Description | <i>Lateral boundary tendency of rho_zz</i> |
| Accessed in code | as ‘lbc_rho_zz’ from the ‘lbc’ pool |

lbc_rtheta_m (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ K\ m^{-3}\ s^{-2}$ |
| Description | <i>Lateral boundary tendency of rho_zz-coupled theta_m</i> |
| Accessed in code | as ‘lbc_rtheta_m’ from the ‘lbc’ pool |

lbc_ru (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}\ s^{-2}$ |
| Description | <i>Lateral boundary tendency of rho_zz-coupled u</i> |
| Accessed in code | as ‘lbc_ru’ from the ‘lbc’ pool |

lbc_scalars (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Description | <i>‘lbc_scalars’ is a variable array used in code which is made up of the constituent fields listed below.</i> |
| Accessed in code | as ‘lbc_scalars’ from the ‘lbc’ pool |

Contains constituents:

lbc_qv — constituent field of var_array **lbc_scalars**

| | |
|-------------|--|
| Description | <i>Lateral boundary tendency of water vapor mixing ratio</i> |
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Array Group | moist |

lbc_qc — constituent field of var_array **lbc_scalars**

| | |
|--------------|--|
| Description | <i>Lateral boundary tendency of cloud water mixing ratio</i> |
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, cu_ntiedtke_in, mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

lbc_qr — constituent field of var_array **lbc_scalars**

| | |
|--------------|--|
| Description | <i>Lateral boundary tendency of rain water mixing ratio</i> |
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

lbc_qi — constituent field of var_array **lbc_scalars**

| | |
|--------------|--|
| Description | <i>Lateral boundary tendency of ice mixing ratio</i> |
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, cu_ntiedtke_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

lbc_qs — constituent field of var_array **lbc_scalars**

| | |
|--------------|---|
| Description | <i>Lateral boundary tendency of snow mixing ratio</i> |
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

lbc_qg — constituent field of var_array **lbc_scalars**

| | |
|--------------|--|
| Description | <i>Lateral boundary tendency of graupel mixing ratio</i> |
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

lbc_nr — constituent field of var_array **lbc_scalars**

| | |
|--------------|---|
| Description | <i>Lateral boundary tendency of rain number concentration</i> |
| Units | $m^{-3}\ s^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_in, mp_thompson_aers_in |

lbc_ni — constituent field of var_array **lbc_scalars**

| | |
|--------------|--|
| Description | <i>Lateral boundary tendency of ice number concentration</i> |
| Units | $m^{-3}\ s^{-1}$ |
| Array Group | number |
| Allocated by | bl_mynn_in, mp_thompson_in, mp_thompson_aers_in |

lbc_nc — constituent field of var_array **lbc_scalars**

| | |
|--------------|--|
| Description | <i>Lateral boundary tendency of cloud water number concentration</i> |
| Units | $m^{-3} s^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

lbc_nifa — constituent field of var_array **lbc_scalars**

| | |
|--------------|---|
| Description | <i>Lateral boundary tendency of ice-friendly aerosol number concentration</i> |
| Units | $m^{-3} s^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

lbc_nwfa — constituent field of var_array **lbc_scalars**

| | |
|--------------|---|
| Description | <i>Lateral boundary tendency of water-friendly aerosol number concentration</i> |
| Units | $m^{-3} s^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

lbc_theta (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K s^{-2}$ |
| Description | <i>Lateral boundary tendency of theta</i> |
| Accessed in code | as ‘lbc_theta’ from the ‘lbc’ pool |

lbc_u (real) (nVertLevels, nEdges, Time)

| | |
|------------------|---------------------------------------|
| Units | $m s^{-2}$ |
| Description | <i>Lateral boundary tendency of u</i> |
| Accessed in code | as ‘lbc_u’ from the ‘lbc’ pool |

lbc_w (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $m s^{-2}$ |
| Description | <i>Lateral boundary tendency of w</i> |
| Accessed in code | as ‘lbc_w’ from the ‘lbc’ pool |

lcl (real) (nCells, Time)

| | |
|------------------|----------------------------------|
| Units | m |
| Description | <i>Lifted condensation level</i> |
| Accessed in code | as ‘lcl’ from the ‘diag’ pool |

lfc (real) (nCells, Time)

| | |
|------------------|---------------------------------|
| Units | m |
| Description | <i>Level of free convection</i> |
| Accessed in code | as ‘lfc’ from the ‘diag’ pool |

lfmassxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $g m^{-2}$ |
| Description | <i>leaf mass</i> |
| Accessed in code | as ‘lfmassxy’ from the ‘diag_physics_noahmp’ pool |

lh (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>latent heat flux at the surface</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘lh’ from the ‘diag_physics’ pool |

localVerticalUnitVectors (real) (R3, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Cartesian components of the vector pointing in the local vertical direction for a cell</i> |
| Accessed in code | as ‘localVerticalUnitVectors’ from the ‘mesh’ pool |

loctim (real) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>local time</i> |
| Accessed in code | as ‘loctim’ from the ‘diag_physics_noahmp’ pool |

lonCell (real) (nCells)

| | |
|------------------|-----------------------------------|
| Units | <i>rad</i> |
| Description | <i>Longitude of cells</i> |
| Accessed in code | as ‘lonCell’ from the ‘mesh’ pool |

lonEdge (real) (nEdges)

| | |
|------------------|-----------------------------------|
| Units | <i>rad</i> |
| Description | <i>Longitude of edges</i> |
| Accessed in code | as ‘lonEdge’ from the ‘mesh’ pool |

lonVertex (real) (nVertices)

| | |
|------------------|-------------------------------------|
| Units | <i>rad</i> |
| Description | <i>Longitude of vertices</i> |
| Accessed in code | as ‘lonVertex’ from the ‘mesh’ pool |

lwcf (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>top-of-atmosphere cloud longwave radiative forcing</i> |
| Accessed in code | as ‘lwcf’ from the ‘diag_physics’ pool |

lwdnb (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky downward surface longwave radiation flux</i> |
| Accessed in code | as ‘lwdnb’ from the ‘diag_physics’ pool |

lwdnbc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky downward surface longwave radiation flux</i> |
| Accessed in code | as ‘lwdnbc’ from the ‘diag_physics’ pool |

lwdnt (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky downward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘lwdnt’ from the ‘diag_physics’ pool |

lwdntc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky downward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘lwdntc’ from the ‘diag_physics’ pool |

lwupb (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky upward surface longwave radiation flux</i> |
| Accessed in code | as ‘lwupb’ from the ‘diag_physics’ pool |

lwupbc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky upward surface longwave radiation flux</i> |
| Accessed in code | as ‘lwupbc’ from the ‘diag_physics’ pool |

lwupt (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky upward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘lwupt’ from the ‘diag_physics’ pool |

lwuptc (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky upward top-of-the-atmosphere longwave radiation flux</i> |
| Accessed in code | as ‘lwuptc’ from the ‘diag_physics’ pool |

m_hybi (real) (nAerLevels, nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Matched hybi (needs to be re-checked)</i> |
| Accessed in code | as ‘m_hybi’ from the ‘diag_physics’ pool |

m_ps (real) (nCells, Time)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>Surface pressure from match on MPAS grid</i> |
| Accessed in code | as ‘m_ps’ from the ‘diag_physics’ pool |

mavail (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>surface moisture availability (between 0 and 1)</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘mavail’ from the ‘diag_physics’ pool |

meanT_500_300 (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>K</i> |
| Description | <i>Mean temperature in the 300 hPa - 500 hPa layer</i> |
| Accessed in code | as ‘meanT_500_300’ from the ‘diag’ pool |

meshDensity (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Mesh density function (used when generating the mesh) evaluated at a cell</i> |
| Accessed in code | as ‘meshDensity’ from the ‘mesh’ pool |

meshScalingDel2 (real) (nEdges)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Scaling coefficient for ∇^2 eddy viscosity</i> |
| Accessed in code | as ‘meshScalingDel2’ from the ‘mesh’ pool |

meshScalingDel4 (real) (nEdges)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Scaling coefficient for ∇^4 eddy hyper-viscosity</i> |
| Accessed in code | as ‘meshScalingDel4’ from the ‘mesh’ pool |

meshScalingRegionalCell (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Cell-centered Scaling coefficient for relaxation zone</i> |
| Accessed in code | as ‘meshScalingRegionalCell’ from the ‘mesh’ pool |

meshScalingRegionalEdge (real) (nEdges)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Edge-centered Scaling coefficient for relaxation zone</i> |
| Accessed in code | as ‘meshScalingRegionalEdge’ from the ‘mesh’ pool |

mifract (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>micro irrigation fraction</i> |
| Accessed in code | as ‘mifract’ from the ‘diag_physics_noahmp’ pool |

mminlu (text) ()

| | |
|------------------|---------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>land use classification</i> |
| Accessed in code | as ‘mminlu’ from the ‘sfc_input’ pool |

mol (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | K |
| Description | T^* in similarity theory |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘mol’ from the ‘diag_physics’ pool |

mslp (real) (nCells, Time)

| | |
|------------------|--------------------------------|
| Units | Pa |
| Description | <i>Mean sea-level pressure</i> |
| Accessed in code | as ‘mslp’ from the ‘diag’ pool |

mu_c (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>gamma shape parameter</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in |
| Accessed in code | as ‘mu_c’ from the ‘diag_physics’ pool |

nAdvCellsForEdge (integer) (nEdges)

| | |
|------------------|--|
| Units | - |
| Description | <i>Number of cells used to reconstruct a cell-based field at an edge</i> |
| Accessed in code | as ‘nAdvCellsForEdge’ from the ‘mesh’ pool |

nc — see ‘scalars’

nca (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>s</i> |
| Description | <i>relaxation time for KF parameterization of convection</i> |
| Allocated by | cu_kain_fritsch_in |
| Accessed in code | as ‘nca’ from the ‘diag_physics’ pool |

ndays_accum (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>number of accumulated days in a year</i> |
| Accessed in code | as ‘ndays_accum’ from the ‘diag_physics’ pool |

nearestRelaxationCell (integer) (nCells)

| | |
|------------------|--|
| Units | - |
| Description | <i>For cells in the specified zone, gives the index of the nearest cell in the relaxation zone</i> |
| Accessed in code | as ‘nearestRelaxationCell’ from the ‘mesh’ pool |

nEdgesOnCell (integer) (nCells)

| | |
|------------------|---|
| Units | - |
| Description | <i>Number of edges forming the boundary of a cell</i> |
| Accessed in code | as ‘nEdgesOnCell’ from the ‘mesh’ pool |

nEdgesOnEdge (integer) (nEdges)

| | |
|------------------|--|
| Units | - |
| Description | <i>Number of edges involved in reconstruction of tangential velocity for an edge</i> |
| Accessed in code | as ‘nEdgesOnEdge’ from the ‘mesh’ pool |

neexy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $g\ m^{-2}\ s^{-1}\ C$ |
| Description | <i>net ecosystem exchange</i> |
| Accessed in code | as ‘neexy’ from the ‘output_noahmp’ pool |

ni — see ‘scalars’**nifa** — see ‘scalars’**nifa2d** (real) (nCells, Time)

| | |
|------------------|--|
| Units | $nb\ kg^{-1}\ s^{-1}$ |
| Description | <i>surface emission of ice-friendly aerosols</i> |
| Allocated by | mp_thompson_aers_in |
| Accessed in code | as ‘nifa2d’ from the ‘diag_physics’ pool |

nlrad (integer) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>number of layers added above the model-top in the RRTMG lw radiation code</i> |
| Accessed in code | as ‘nlrad’ from the ‘diag_physics’ pool |

noahres (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>residual of the Noah surface energy budget</i> |
| Accessed in code | as ‘noahres’ from the ‘diag_physics’ pool |

nominalMinDc (real) ()

| | |
|------------------|--|
| Units | m |
| Description | <i>Nominal minimum dcEdge value where meshDensity == 1.0</i> |
| Accessed in code | as ‘nominalMinDc’ from the ‘mesh’ pool |

north (real) (R3, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Cartesian components of the unit vector pointing north</i> |
| Accessed in code | as ‘north’ from the ‘mesh’ pool |

nppxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $g\ m^{-2}\ s^{-1}\ C$ |
| Description | <i>net primary productivity</i> |
| Accessed in code | as ‘nppxy’ from the ‘output_noahmp’ pool |

nr — see ‘scalars’**nsteps_accum** (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>number of accumulated time-steps in a day</i> |
| Accessed in code | as ‘nsteps_accum’ from the ‘diag_physics’ pool |

nt_c (real) (nCells)

| | |
|------------------|--|
| Units | $nb\ kg^{-1}$ |
| Description | <i>one-moment constant cloud water droplet concentration</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in |
| Accessed in code | as ‘nt_c’ from the ‘diag_physics’ pool |

nwfa — see ‘scalars’**nwfa2d** (real) (nCells, Time)

| | |
|------------------|--|
| Units | $nb\ kg^{-1}\ s^{-1}$ |
| Description | <i>surface emission of water-friendly aerosols</i> |
| Allocated by | mp_thompson_aers_in |
| Accessed in code | as ‘nwfa2d’ from the ‘diag_physics’ pool |

o3clim (real) (nOznLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>mol mol⁻¹</i> |
| Description | <i>climatological ozone on prescribed pressure levels at current time</i> |
| Accessed in code | as ‘o3clim’ from the ‘diag_physics’ pool |

o3vmr (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>mol mol⁻¹</i> |
| Description | <i>ozone volume mixing ratio</i> |
| Accessed in code | as ‘o3vmr’ from the ‘diag_physics’ pool |

oa1 (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for westerly flow</i> |
| Accessed in code | as ‘oa1’ from the ‘sfc_input’ pool |

oa1ls (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for westerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa1ls’ from the ‘sfc_input’ pool |

oa1ss (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for westerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa1ss’ from the ‘sfc_input’ pool |

oa2 (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for southerly flow</i> |
| Accessed in code | as ‘oa2’ from the ‘sfc_input’ pool |

oa2ls (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for southerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa2ls’ from the ‘sfc_input’ pool |

oa2ss (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for southerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa2ss’ from the ‘sfc_input’ pool |

oa3 (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for south-westerly flow</i> |
| Accessed in code | as ‘oa3’ from the ‘sfc_input’ pool |

oa3ls (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for south-westerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa3ls’ from the ‘sfc_input’ pool |

oa3ss (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for south-westerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa3ss’ from the ‘sfc_input’ pool |

oa4 (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for north-westerly flow</i> |
| Accessed in code | as ‘oa4’ from the ‘sfc_input’ pool |

oa4ls (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for north-westerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa4ls’ from the ‘sfc_input’ pool |

oa4ss (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>asymmetry of subgrid-scale orography for north-westerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘oa4ss’ from the ‘sfc_input’ pool |

ocphi — see ‘aerosols’**ocpho** — see ‘aerosols’**ol1** (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for westerly flow</i> |
| Accessed in code | as ‘ol1’ from the ‘sfc_input’ pool |

ol1ls (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for westerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol1ls’ from the ‘sfc_input’ pool |

ol1ss (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for westerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol1ss’ from the ‘sfc_input’ pool |

ol2 (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for southerly flow</i> |
| Accessed in code | as ‘ol2’ from the ‘sfc_input’ pool |

ol2ls (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for southerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol2ls’ from the ‘sfc_input’ pool |

ol2ss (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for southerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol2ss’ from the ‘sfc_input’ pool |

ol3 (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for south-westerly flow</i> |
| Accessed in code | as ‘ol3’ from the ‘sfc_input’ pool |

ol3ls (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for south-westerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol3ls’ from the ‘sfc_input’ pool |

ol3ss (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for south-westerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol3ss’ from the ‘sfc_input’ pool |

ol4 (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for north-westerly flow</i> |
| Accessed in code | as ‘ol4’ from the ‘sfc_input’ pool |

ol4ls (real) (nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for north-westerly flow (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol4ls’ from the ‘sfc_input’ pool |

ol4ss (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>effective orographic length for north-westerly flow (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘ol4ss’ from the ‘sfc_input’ pool |

olrtoa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky top-of-atmosphere outgoing longwave radiation flux</i> |
| Accessed in code | as ‘olrtoa’ from the ‘diag_physics’ pool |

ozmixm (real) (nMonths, nOznLevels, nCells)

| | |
|------------------|---|
| Units | $mol\ mol^{-1}$ |
| Description | <i>monthly-mean climatological ozone defined at fixed pressure levels</i> |
| Accessed in code | as ‘ozmixm’ from the ‘atm_input’ pool |

pahbxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W m^{-2}$ |
| Description | <i>precipitation advected energy - to bare ground</i> |
| Accessed in code | as ‘pahbxy’ from the ‘output_noahmp’ pool |

pahgxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>precipitation advected energy - to below canopy</i> |
| Accessed in code | as ‘pahgxy’ from the ‘output_noahmp’ pool |

pahvxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>precipitation advected energy - to vegetation</i> |
| Accessed in code | as ‘pahvxy’ from the ‘output_noahmp’ pool |

pahxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W m^{-2}$ |
| Description | <i>precipitation advected energy</i> |
| Accessed in code | as ‘pahxy’ from the ‘output_noahmp’ pool |

pin (real) (nOznLevels)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>fixed pressure levels at which climatological ozone is defined</i> |
| Accessed in code | as ‘pin’ from the ‘atm_input’ pool |

plrad (real) (nCells, Time)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>pressure at model-top</i> |
| Accessed in code | as ‘plrad’ from the ‘diag_physics’ pool |

pondingxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | mm |
| Description | <i>surface ponding from complete pack melt</i> |
| Accessed in code | as ‘pondingxy’ from the ‘output_noahmp’ pool |

potevp (real) (nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>accumulated potential evaporation</i> |
| Accessed in code | as ‘potevp’ from the ‘diag_physics’ pool |

precipw (real) (nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-2}$ |
| Description | <i>precipitable water</i> |
| Accessed in code | as ‘precipw’ from the ‘diag_physics’ pool |

pressure (real) (nVertLevels, nCells, Time)

| | |
|------------------|------------------------------------|
| Units | Pa |
| Description | <i>Pressure</i> |
| Accessed in code | as ‘pressure’ from the ‘diag’ pool |

pressure_base (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>Base state pressure</i> |
| Accessed in code | as ‘pressure_base’ from the ‘diag’ pool |

pressure_p (real) (nVertLevels, nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | Pa |
| Description | <i>Perturbation pressure</i> |
| Accessed in code | as ‘pressure_p’ from the ‘diag’ pool |

psih (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>similarity stability function for heat</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘psih’ from the ‘diag_physics’ pool |

psim (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>similarity stability function for momentum</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘psim’ from the ‘diag_physics’ pool |

psnxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>umol CO₂ m⁻² s⁻¹</i> |
| Description | <i>total photosynthesis</i> |
| Accessed in code | as ‘psnxy’ from the ‘output_noahmp’ pool |

pv_cell (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>s⁻¹</i> |
| Description | <i>absolute vertical vorticity averaged to the cell center from the vertices</i> |
| Accessed in code | as ‘pv_cell’ from the ‘diag’ pool |

pv_edge (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | <i>s⁻¹</i> |
| Description | <i>absolute vertical vorticity averaged to the cell edge from the vertices</i> |
| Accessed in code | as ‘pv_edge’ from the ‘diag’ pool |

pv_vertex (real) (nVertLevels, nVertices, Time)

| | |
|------------------|--|
| Units | <i>s⁻¹</i> |
| Description | <i>absolute vertical vorticity at a vertex</i> |
| Accessed in code | as ‘pv_vertex’ from the ‘diag’ pool |

q2 (real) (nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | <i>kg kg⁻¹</i> |
| Description | <i>2-meter specific humidity</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘q2’ from the ‘diag_physics’ pool |

q2mbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}$ |
| Description | <i>2-meter water vapor mixing ratio over bare ground</i> |
| Accessed in code | as ‘q2mbxy’ from the ‘output_noahmp’ pool |

q2mvxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>2-meter water vapor mixing ratio over canopy</i> |
| Accessed in code | as ‘q2mvxy’ from the ‘output_noahmp’ pool |

q2mxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>grid-mean 2-meter water vapor mixing ratio</i> |
| Accessed in code | as ‘q2mxy’ from the ‘output_noahmp’ pool |

qbouy (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m^2\ s^{-2}$ |
| Description | <i>TKE production - buoyancy</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qbouy’ from the ‘diag_physics’ pool |

qc — see ‘scalars’

qc_amb — see ‘scalars_amb’

qc_bl (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qc_bl’ from the ‘diag_physics’ pool |

qc_cu (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>in-cloud cloud water mixing ratio in Grell-Freitas cloud model</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘qc_cu’ from the ‘diag_physics’ pool |

qcg (real) (nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>cloud water mixing ratio at the ground surface</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qcg’ from the ‘diag_physics’ pool |

qdewcxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy dew rate</i> |
| Accessed in code | as ‘qdewcxy’ from the ‘output_noahmp’ pool |

qdiss (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m^2\ s^{-2}$ |
| Description | <i>TKE dissipation</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qdiss’ from the ‘diag_physics’ pool |

qdriprxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy rain drip rate</i> |
| Accessed in code | as ‘qdriprxy’ from the ‘output_noahmp’ pool |

qdripsxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy snow drip rate</i> |
| Accessed in code | as ‘qdripsxy’ from the ‘output_noahmp’ pool |

qevacxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy evaporation rate</i> |
| Accessed in code | as ‘qevacxy’ from the ‘output_noahmp’ pool |

qfrocxxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy frost rate</i> |
| Accessed in code | as ‘qfrocxxy’ from the ‘output_noahmp’ pool |

qfrzcxxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy liquid freeze rate</i> |
| Accessed in code | as ‘qfrzcxxy’ from the ‘output_noahmp’ pool |

qfx (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>upward moisture flux at the surface</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘qfx’ from the ‘diag_physics’ pool |

qg — see ‘scalars’**qg_amb** — see ‘scalars_amb’**qgh** (real) (nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>lowest level saturation mixing ratio</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘qgh’ from the ‘diag_physics’ pool |

qi — see ‘scalars’**qi_amb** — see ‘scalars_amb’

qi_bl (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qi_bl’ from the ‘diag_physics’ pool |

qi_cu (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}$ |
| Description | <i>in-cloud cloud ice mixing ratio in Grell-Freitas cloud model</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘qi_cu’ from the ‘diag_physics’ pool |

qintrxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy rain interception rate</i> |
| Accessed in code | as ‘qintrxy’ from the ‘output_noahmp’ pool |

qintsxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy snow interception rate</i> |
| Accessed in code | as ‘qintsxy’ from the ‘output_noahmp’ pool |

qke (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $m^2\ s^{-2}$ |
| Description | <i>twice turbulent kinetic energy</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qke’ from the ‘diag_physics’ pool |

qke_adv (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m^2\ s^{-2}$ |
| Description | <i>twice turbulent kinetic energy</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qke_adv’ from the ‘diag_physics’ pool |

qmeltcxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy snow melt rate</i> |
| Accessed in code | as ‘qmeltcxy’ from the ‘output_noahmp’ pool |

qmeltxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>snowpack melting rate due to phase change</i> |
| Accessed in code | as ‘qmeltxy’ from the ‘output_noahmp’ pool |

qr — see ‘scalars’**qr_amb** — see ‘scalars_amb’**qrainxy** (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>rainfall on the ground</i> |
| Accessed in code | as ‘qrainxy’ from the ‘diag_physics_noahmp’ pool |

qs — see ‘scalars’**qs_amb** — see ‘scalars_amb’**qsfc** (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}$ |
| Description | <i>specific humidity at lower boundary</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘qsfc’ from the ‘diag_physics’ pool |

qshear (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2\ s^{-2}$ |
| Description | <i>TKE production - shear</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qshear’ from the ‘diag_physics’ pool |

qsnbotxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>total liquid water (melt + rain through pack) out of snowpack bottom</i> |
| Accessed in code | as ‘qsnbotxy’ from the ‘output_noahmp’ pool |

qsnfroxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>snow pack frost rate</i> |
| Accessed in code | as ‘qsnfroxy’ from the ‘output_noahmp’ pool |

qsnowxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>snowfall on the ground</i> |
| Accessed in code | as ‘qsnowxy’ from the ‘diag_physics_noahmp’ pool |

qsnsubxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>snow pack sublimation rate</i> |
| Accessed in code | as ‘qsnsubxy’ from the ‘output_noahmp’ pool |

qsq (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $kg\ kg^{-12}$ |
| Description | <i>liquid water variance</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qsq’ from the ‘diag_physics’ pool |

qsubcxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>canopy sublimation rate</i> |
| Accessed in code | as ‘qsubcxy’ from the ‘output_noahmp’ pool |

qtdrain (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>tile drainage</i> |
| Accessed in code | as ‘qtdrain’ from the ‘output_noahmp’ pool |

qthrorxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>rain throughfall rate</i> |
| Accessed in code | as ‘qthrorxy’ from the ‘output_noahmp’ pool |

qthrosxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm\ s^{-1}$ |
| Description | <i>snow throughfall rate</i> |
| Accessed in code | as ‘qthrosxy’ from the ‘output_noahmp’ pool |

qv — see ‘scalars’

qv_amb — see ‘scalars_amb’

qv_init (real) (nVertLevels)

| | |
|------------------|-----------------------------------|
| Units | $kg\ kg^{-1}$ |
| Description | <i>qv reference profile</i> |
| Accessed in code | as ‘qv_init’ from the ‘mesh’ pool |

qwt (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $m^2\ s^{-2}$ |
| Description | <i>TKE vertical distribution</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘qwt’ from the ‘diag_physics’ pool |

rainc (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>accumulated convective precipitation</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as ‘rainc’ from the ‘diag_physics’ pool |

raincv (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>time-step convective precipitation</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as ‘raincv’ from the ‘diag_physics’ pool |

rainlsm (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>accumulated liquid precipitation into LSM</i> |
| Accessed in code | as ‘rainlsm’ from the ‘output_noahmp’ pool |

rainnc (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>accumulated total grid-scale precipitation</i> |
| Allocated by | mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘rainnc’ from the ‘diag_physics’ pool |

rainncv (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>time-step total grid-scale precipitation</i> |
| Allocated by | mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘rainncv’ from the ‘diag_physics’ pool |

rainprod (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | s^{-1} |
| Description | <i>rain production rate</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘rainprod’ from the ‘diag_physics’ pool |

rdzu (real) (nVertLevels)

| | |
|------------------|--------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Reciprocal dzu</i> |
| Accessed in code | as ‘rdzu’ from the ‘mesh’ pool |

rdzw (real) (nVertLevels)

| | |
|------------------|--------------------------------|
| Units | <i>unitless</i> |
| Description | <i>Reciprocal dzw</i> |
| Accessed in code | as ‘rdzw’ from the ‘mesh’ pool |

re_cloud (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>effective radius of cloud water droplets</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘re_cloud’ from the ‘diag_physics’ pool |

re_ice (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>effective radius of cloud ice crystals</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘re_ice’ from the ‘diag_physics’ pool |

re_snow (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>effective radius of snow crystals</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘re_snow’ from the ‘diag_physics’ pool |

rechxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>water table recharge</i> |
| Accessed in code | as ‘rechxy’ from the ‘diag_physics_noahmp’ pool |

refl10cm (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>dBZ</i> |
| Description | <i>10 cm radar reflectivity</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘refl10cm’ from the ‘diag_physics’ pool |

refl10cm_1km (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>dBZ</i> |
| Description | <i>diagnosed 10 cm radar reflectivity at 1 km AGL</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘refl10cm_1km’ from the ‘diag_physics’ pool |

refl10cm_1km_max (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>dBZ</i> |
| Description | <i>maximum diagnosed 10 cm radar reflectivity at 1 km AGL since last output time</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘refl10cm_1km_max’ from the ‘diag_physics’ pool |

refl10cm_max (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>dBZ</i> |
| Description | <i>10 cm maximum radar reflectivity</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘refl10cm_max’ from the ‘diag_physics’ pool |

regime (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>flag indicating the PBL regime (stable,unstable,...)</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘regime’ from the ‘diag_physics’ pool |

relhum (real) (nVertLevels, nCells, Time)

| | |
|------------------|----------------------------------|
| Units | <i>percent</i> |
| Description | <i>Relative humidity</i> |
| Accessed in code | as ‘relhum’ from the ‘diag’ pool |

relhum_100hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 100 hPa</i> |
| Accessed in code | as ‘relhum_100hPa’ from the ‘diag’ pool |

relhum_200hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 200 hPa</i> |
| Accessed in code | as ‘relhum_200hPa’ from the ‘diag’ pool |

relhum_250hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 250 hPa</i> |
| Accessed in code | as ‘relhum_250hPa’ from the ‘diag’ pool |

relhum_500hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 500 hPa</i> |
| Accessed in code | as ‘relhum_500hPa’ from the ‘diag’ pool |

relhum_50hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 50 hPa</i> |
| Accessed in code | as ‘relhum_50hPa’ from the ‘diag’ pool |

relhum_700hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 700 hPa</i> |
| Accessed in code | as ‘relhum_700hPa’ from the ‘diag’ pool |

relhum_850hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 850 hPa</i> |
| Accessed in code | as ‘relhum_850hPa’ from the ‘diag’ pool |

relhum_925hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>percent</i> |
| Description | <i>Relative humidity vertically interpolated to 925 hPa</i> |
| Accessed in code | as ‘relhum_925hPa’ from the ‘diag’ pool |

rho (real) (nVertLevels, nCells, Time)

| | |
|------------------|-------------------------------|
| Units | <i>kg m⁻³</i> |
| Description | <i>Dry air density</i> |
| Accessed in code | as ‘rho’ from the ‘diag’ pool |

rho_amb (real) (nVertLevels, nCells, Time)

| | |
|------------------|-----------------------------------|
| Units | <i>kg m⁻³</i> |
| Description | <i>Dry air density increment</i> |
| Accessed in code | as ‘rho’ from the ‘tend_iau’ pool |

rho_base (real) (nVertLevels, nCells, Time)

| | |
|------------------|------------------------------------|
| Units | <i>kg m⁻³</i> |
| Description | <i>Base state dry air density</i> |
| Accessed in code | as ‘rho_base’ from the ‘diag’ pool |

rho_edge (real) (nVertLevels, nEdges, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-3}$ |
| Description | <i>rho_zz averaged from cell centers to the cell edge</i> |
| Accessed in code | as ‘rho_edge’ from the ‘diag’ pool |

rho_p (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-3}$ |
| Description | <i>rho/zz perturbation from the reference state value, advanced over acoustic steps</i> |
| Accessed in code | as ‘rho_p’ from the ‘diag’ pool |

rho_p_save (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-3}$ |
| Description | <i>predicted value rho_p, saved before acoustic steps</i> |
| Accessed in code | as ‘rho_p_save’ from the ‘diag’ pool |

rho_pp (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-3}$ |
| Description | <i>rho/zz perturbation from rho_pp, advanced over acoustic steps</i> |
| Accessed in code | as ‘rho_pp’ from the ‘diag’ pool |

rho_zz (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-3}$ |
| Description | <i>Dry air density divided by d(zeta)/dz</i> |
| Accessed in code | as ‘rho_zz’ from the ‘state’ pool |

rho_zz_old_split (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-3}$ |
| Description | <i>rho/zz</i> |
| Accessed in code | as ‘rho_zz_old_split’ from the ‘diag’ pool |

rmol (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>1./L Monin Obukhov length</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘rmol’ from the ‘diag_physics’ pool |

rncblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>nb kg⁻¹ s⁻¹</i> |
| Description | <i>tendency of cloud liquid water number concentration due to pbl processes</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘rncblten’ from the ‘tend_physics’ pool |

rncmpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>nb kg⁻¹ s⁻¹</i> |
| Description | <i>tendency of cloud water number concentration due to microphysics</i> |
| Allocated by | mp_thompson_aers_in |
| Accessed in code | as ‘rncmpten’ from the ‘tend_physics’ pool |

rniblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>nb kg⁻¹ s⁻¹</i> |
| Description | <i>tendency of cloud ice number concentration due to pbl processes</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘rniblten’ from the ‘tend_physics’ pool |

rnifablten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>nb kg⁻¹ s⁻¹</i> |
| Description | <i>tendency of ice-friendly aerosol number concentration due to pbl processes</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘rnifablten’ from the ‘tend_physics’ pool |

rnifampten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $nb \ kg^{-1} \ s^{-1}$ |
| Description | <i>tendency of ice-friendly aerosol number concentration due to microphysics</i> |
| Allocated by | mp_thompson_aers_in |
| Accessed in code | as ‘rnifampten’ from the ‘tend_physics’ pool |

rnimpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $nb \ kg^{-1} \ s^{-1}$ |
| Description | <i>tendency of cloud ice number concentration due to microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in |
| Accessed in code | as ‘rnimpten’ from the ‘tend_physics’ pool |

rnrnpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $nb \ kg^{-1} \ s^{-1}$ |
| Description | <i>tendency of rain number concentration due to microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in |
| Accessed in code | as ‘rnrnpten’ from the ‘tend_physics’ pool |

rnwfablten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $nb \ kg^{-1} \ s^{-1}$ |
| Description | <i>tendency of water-friendly aerosol number concentration due to pbl processes</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘rnwfablten’ from the ‘tend_physics’ pool |

rnwfampten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $nb \ kg^{-1} \ s^{-1}$ |
| Description | <i>tendency of water-friendly aerosol number concentration due to microphysics</i> |
| Allocated by | mp_thompson_aers_in |
| Accessed in code | as ‘rnwfampten’ from the ‘tend_physics’ pool |

rqcblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of cloud water mixing ratio due to pbl processes</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as 'rqcblten' from the 'tend_physics' pool |

rqccuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of cloud water mixing ratio due to cumulus convection</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as 'rqccuten' from the 'tend_physics' pool |

rqcmpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of cloud water mixing ratio due to cloud microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as 'rqcmpten' from the 'tend_physics' pool |

rqgmpften (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of graupel mixing ratio due to cloud microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as 'rqgmpften' from the 'tend_physics' pool |

rqiblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of cloud ice mixing ratio due to pbl processes</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as 'rqiblten' from the 'tend_physics' pool |

rqicuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of cloud ice mixing ratio due to cumulus convection</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as 'rqicuten' from the 'tend_physics' pool |

rqimpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of cloud ice mixing ratio due to cloud microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as 'rqimpten' from the 'tend_physics' pool |

rqrcuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of rain mixing ratio due to cumulus convection</i> |
| Allocated by | cu_kain_fritsch_in |
| Accessed in code | as 'rqrcuten' from the 'tend_physics' pool |

rqrmpften (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of rain mixing ratio due to cloud microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as 'rqrmpften' from the 'tend_physics' pool |

rqsblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of snow mixing ratio due to pbl processes</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as 'rqsblten' from the 'tend_physics' pool |

rqscuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of snow mixing ratio due to cumulus convection</i> |
| Allocated by | cu_kain_fritsch_in |
| Accessed in code | as ‘rqscuten’ from the ‘tend_physics’ pool |

rqsmpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of snow mixing ratio due to cloud microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘rqsmpten’ from the ‘tend_physics’ pool |

rqvblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of water vapor mixing ratio due to pbl processes</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘rqvblten’ from the ‘tend_physics’ pool |

rqvcuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of water vapor mixing ratio due to cumulus convection</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as ‘rqvcuten’ from the ‘tend_physics’ pool |

rqvdynten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of water vapor due to horizontal and vertical advections</i> |
| Allocated by | cu_grell_freitas_in, cu_ntiedtke_in |
| Accessed in code | as ‘rqvdynten’ from the ‘tend_physics’ pool |

rqvmpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>tendency of water vapor mixing ratio due to cloud microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘rqvmpten’ from the ‘tend_physics’ pool |

rre_cloud (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>microns</i> |
| Description | <i>effective radius of cloud water droplets calculated in RRTMG radiation</i> |
| Accessed in code | as ‘rre_cloud’ from the ‘diag_physics’ pool |

rre_ice (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>microns</i> |
| Description | <i>effective radius of cloud ice crystals calculated in RRTMG radiation</i> |
| Accessed in code | as ‘rre_ice’ from the ‘diag_physics’ pool |

rre_snow (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>microns</i> |
| Description | <i>effective radius of snow crystals calculated in RRTMG radiation</i> |
| Accessed in code | as ‘rre_snow’ from the ‘diag_physics’ pool |

rs (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $s\ m^{-1}$ |
| Description | <i>total stomatal resistance</i> |
| Accessed in code | as ‘rs’ from the ‘output_noahmp’ pool |

rsshaxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $s\ m^{-1}$ |
| Description | <i>shaded stomatal resistance</i> |
| Accessed in code | as ‘rsshaxy’ from the ‘output_noahmp’ pool |

rssunxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $s\ m^{-1}$ |
| Description | <i>sunlit stomatal resistance</i> |
| Accessed in code | as ‘rssunxy’ from the ‘output_noahmp’ pool |

rt_diabatic_tend (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K\ s^{-1}$ |
| Description | <i>Tendency of modified potential temperature due to cloud microphysics</i> |
| Accessed in code | as ‘rt_diabatic_tend’ from the ‘tend’ pool |

rthblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K\ s^{-1}$ |
| Description | <i>tendency of potential temperature due to pbl processes</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘rthblten’ from the ‘tend_physics’ pool |

rthcuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $K\ s^{-1}$ |
| Description | <i>tendency of potential temperature due to cumulus convection</i> |
| Allocated by | cu_grell_freitas_in, cu_kain_fritsch_in, cu_ntiedtke_in |
| Accessed in code | as ‘rthcuten’ from the ‘tend_physics’ pool |

rthdynten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $K\ s^{-1}$ |
| Description | <i>tendency of temperature due to horizontal and vertical advections</i> |
| Accessed in code | as ‘rthdynten’ from the ‘tend_physics’ pool |

rtheta_base (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $kg\ K\ m^{-3}$ |
| Description | <i>reference state rho*theta/zz</i> |
| Accessed in code | as ‘rtheta_base’ from the ‘diag’ pool |

rtheta_p (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ K\ m^{-3}$ |
| Description | <i>rho*theta_m/zz perturbation from the reference state value</i> |
| Accessed in code | as ‘rtheta_p’ from the ‘diag’ pool |

rtheta_p_save (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ K\ m^{-3}$ |
| Description | <i>predicted value rtheta_p, saved before acoustic steps</i> |
| Accessed in code | as ‘rtheta_p_save’ from the ‘diag’ pool |

rtheta_pp (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ K\ m^{-3}$ |
| Description | <i>rho*theta_m/zz perturbation from rtheta_p</i> |
| Accessed in code | as ‘rtheta_pp’ from the ‘diag’ pool |

rtheta_pp_old (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ K\ m^{-3}$ |
| Description | <i>old time level values of rho*theta_m/zz perturbation from rtheta_p, used in 3D divergence damping</i> |
| Accessed in code | as ‘rtheta_pp_old’ from the ‘diag’ pool |

rthmpten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $K\ s^{-1}$ |
| Description | <i>tendency of potential temperature due to cloud microphysics</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘rthmpten’ from the ‘tend_physics’ pool |

rthratenlw (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K\ s^{-1}$ |
| Description | <i>tendency of potential temperature due to long wave radiation</i> |
| Accessed in code | as ‘rthratenlw’ from the ‘tend_physics’ pool |

rthratensw (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $K \ s^{-1}$ |
| Description | <i>tendency of potential temperature due to short wave radiation</i> |
| Accessed in code | as ‘rthratensw’ from the ‘tend_physics’ pool |

rtmassxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $g \ m^{-2}$ |
| Description | <i>mass of fine roots</i> |
| Accessed in code | as ‘rtmassxy’ from the ‘diag_physics_noahmp’ pool |

ru (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $kg \ m^{-2} \ s^{-1}$ |
| Description | <i>horizontal momentum at cell edge ($\rho * u / zz$)</i> |
| Accessed in code | as ‘ru’ from the ‘diag’ pool |

ru_p (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $kg \ m^{-2} \ s^{-1}$ |
| Description | <i>acoustic perturbation horizontal momentum at cell edge ($\rho * u / zz$)</i> |
| Accessed in code | as ‘ru_p’ from the ‘diag’ pool |

ru_save (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $kg \ m^{-2} \ s^{-1}$ |
| Description | <i>predicted value of horizontal momentum, saved before acoustic steps</i> |
| Accessed in code | as ‘ru_save’ from the ‘diag’ pool |

ruAvg (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $kg \ m^{-2} \ s^{-1}$ |
| Description | <i>time-averaged $\rho * u / zz$ used in scalar transport</i> |
| Accessed in code | as ‘ruAvg’ from the ‘diag’ pool |

ruAvg_split (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $kg \ m^{-2} \ s^{-1}$ |
| Description | <i>time-averaged $\rho * u / zz$ used in scalar transport</i> |
| Accessed in code | as ‘ruAvg_split’ from the ‘diag’ pool |

rubldiff (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>change in PBL zonal wind tendency due to gravity wave drag over orography</i> |
| Accessed in code | as ‘rubldiff’ from the ‘diag_physics’ pool |

rublten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1} s^{-1}$ |
| Description | <i>tendency of zonal wind due to pbl processes</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘rublten’ from the ‘tend_physics’ pool |

rublten_Edge (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | |
| Description | |
| Accessed in code | as ‘rublten_Edge’ from the ‘tend_physics’ pool |

rucuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1} s^{-1}$ |
| Description | <i>tendency of zonal wind due to cumulus convection</i> |
| Allocated by | cu_grell_freitas_in, cu_ntiedtke_in |
| Accessed in code | as ‘rucuten’ from the ‘tend_physics’ pool |

rucuten_Edge (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | |
| Description | |
| Accessed in code | as ‘rucuten_Edge’ from the ‘tend_physics’ pool |

runsbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm s^{-1}$ |
| Description | <i>sub-surface runoff</i> |
| Accessed in code | as ‘runsbxy’ from the ‘output_noahmp’ pool |

runsfxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $mm\ s^{-1}$ |
| Description | <i>surface runoff</i> |
| Accessed in code | as ‘runsfxy’ from the ‘output_noahmp’ pool |

rv (real) (nVertLevels, nEdges, Time)

| | |
|------------------|------------------------------|
| Units | ??? |
| Description | ??? |
| Accessed in code | as ‘rv’ from the ‘diag’ pool |

rvbldiff (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-2}$ |
| Description | <i>change in PBL meridional wind tendency due to gravity wave drag over orography</i> |
| Accessed in code | as ‘rvbldiff’ from the ‘diag_physics’ pool |

rvblten (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}\ s^{-1}$ |
| Description | <i>tendency of meridional wind due to pbl processes</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘rvblten’ from the ‘tend_physics’ pool |

rvcuten (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m\ s^{-1}\ s^{-1}$ |
| Description | <i>tendency of meridional wind due to cumulus convection</i> |
| Allocated by | cu_grell_freitas_in, cu_ntiedtke_in |
| Accessed in code | as ‘rvcuten’ from the ‘tend_physics’ pool |

rw (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>rho*omega/zz carried at w points</i> |
| Accessed in code | as ‘rw’ from the ‘diag’ pool |

rw_p (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>acoustic perturbation rho*omega/zz carried at w points</i> |
| Accessed in code | as ‘rw_p’ from the ‘diag’ pool |

rw_save (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>predicted value of rho*omega/zz, saved before acoustic steps</i> |
| Accessed in code | as ‘rw_save’ from the ‘diag’ pool |

sagxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>solar radiation absorbed by ground</i> |
| Accessed in code | as ‘sagxy’ from the ‘output_noahmp’ pool |

savxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>solar radiation absorbed by vegetation</i> |
| Accessed in code | as ‘savxy’ from the ‘output_noahmp’ pool |

scalars (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Description | <i>‘scalars’ is a variable array used in code which is made up of the constituent fields listed below.</i> |
| Accessed in code | as ‘scalars’ from the ‘state’ pool |

Contains constituents:

qv — constituent field of var_array **scalars**

| | |
|-------------|---------------------------------|
| Description | <i>Water vapor mixing ratio</i> |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |

qc — constituent field of *var_array scalars*

| | |
|--------------|---|
| Description | <i>Cloud water mixing ratio</i> |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, bl_ysu_in, cu_ntiedtke_in, mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qr — constituent field of *var_array scalars*

| | |
|--------------|--|
| Description | <i>Rain water mixing ratio</i> |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qi — constituent field of *var_array scalars*

| | |
|--------------|--|
| Description | <i>Ice mixing ratio</i> |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, bl_ysu_in, cu_ntiedtke_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qs — constituent field of *var_array scalars*

| | |
|--------------|---|
| Description | <i>Snow mixing ratio</i> |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qg — constituent field of *var_array scalars*

| | |
|--------------|---|
| Description | <i>Graupel mixing ratio</i> |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

ni — constituent field of var_array scalars

| | |
|--------------|---|
| Description | <i>Cloud ice number concentration</i> |
| Units | $nb\ kg^{-1}$ |
| Array Group | number |
| Allocated by | bl_mynn_in, mp_thompson_in, mp_thompson_aers_in |

nr — constituent field of var_array scalars

| | |
|--------------|-------------------------------------|
| Description | <i>Rain number concentration</i> |
| Units | $nb\ kg^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_in, mp_thompson_aers_in |

nc — constituent field of var_array scalars

| | |
|--------------|---|
| Description | <i>Cloud water number concentration</i> |
| Units | $nb\ kg^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

nifa — constituent field of var_array scalars

| | |
|--------------|--|
| Description | <i>Ice-friendly aerosol number concentration</i> |
| Units | $nb\ kg^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

nwfia — constituent field of var_array scalars

| | |
|--------------|--|
| Description | <i>Water-friendly aerosol number concentration</i> |
| Units | $nb\ kg^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

scalars_amb (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Description | <i>'scalars_amb' is a variable array used in code which is made up of the constituent fields listed below.</i> |
| Accessed in code | as 'scalars' from the 'tend_iau' pool |

Contains constituents:

qv_amb — constituent field of var_array *scalars_amb*

| | |
|-------------|------------------------------------|
| Description | Water vapor mixing ratio increment |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |

qc_amb — constituent field of var_array *scalars_amb*

| | |
|--------------|---|
| Description | Cloud water mixing ratio increment |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, bl_ysu_in, cu_ntiedtke_in, mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qr_amb — constituent field of var_array *scalars_amb*

| | |
|--------------|--|
| Description | Rain water mixing ratio increment |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qi_amb — constituent field of var_array *scalars_amb*

| | |
|--------------|--|
| Description | Ice mixing ratio increment |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, bl_ysu_in, cu_ntiedtke_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qs_amb — constituent field of var_array *scalars_amb*

| | |
|--------------|---|
| Description | Snow mixing ratio increment |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

qg_amb — constituent field of var_array **scalars_amb**

| | |
|--------------|---|
| Description | <i>Graupel mixing ratio increment</i> |
| Units | $kg\ kg^{-1}$ |
| Array Group | moist |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

scalars_tend (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Description | <i>'scalars_tend' is a variable array used in code which is made up of the constituent fields listed below.</i> |
| Accessed in code | as 'scalars_tend' from the 'tend' pool |

Contains constituents:

tend_qv — constituent field of var_array **scalars_tend**

| | |
|-------------|---|
| Description | <i>Tendency of water vapor mass per unit volume divided by $d(\zeta)/dz$</i> |
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Array Group | moist |

tend_qc — constituent field of var_array **scalars_tend**

| | |
|--------------|---|
| Description | <i>Tendency of cloud water mass per unit volume divided by $d(\zeta)/dz$</i> |
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, bl_ysu_in, cu_ntiedtke_in, mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

tend_qr — constituent field of var_array **scalars_tend**

| | |
|--------------|--|
| Description | <i>Tendency of rain water mass per unit volume divided by $d(\zeta)/dz$</i> |
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | mp_kessler_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

tend_qi — constituent field of var_array **scalars_tend**

| | |
|--------------|--|
| Description | <i>Tendency of ice mass per unit volume divided by $d(\zeta)/dz$</i> |
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, bl_ysu_in, cu_ntiedtke_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

tend_qs — constituent field of var_array **scalars_tend**

| | |
|--------------|--|
| Description | <i>Tendency of snow mass per unit volume divided by $d(\zeta)/dz$</i> |
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | bl_mynn_in, mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

tend_qg — constituent field of var_array **scalars_tend**

| | |
|--------------|---|
| Description | <i>Tendency of graupel mass per unit volume divided by $d(\zeta)/dz$</i> |
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Array Group | moist |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |

tend_ni — constituent field of var_array **scalars_tend**

| | |
|--------------|---|
| Description | <i>Tendency of cloud ice number concentration multiplied by dry air density divided by $d(\zeta)/dz$</i> |
| Units | $nb\ m^{-3}\ s^{-1}$ |
| Array Group | number |
| Allocated by | bl_mynn_in, mp_thompson_in, mp_thompson_aers_in |

tend_nr — constituent field of var_array **scalars_tend**

| | |
|--------------|--|
| Description | <i>Tendency of rain number concentration multiplied by dry air density divided by $d(\zeta)/dz$</i> |
| Units | $nb\ m^{-3}\ s^{-1}$ |
| Array Group | number |
| Allocated by | mp_thompson_in, mp_thompson_aers_in |

tend_nc — constituent field of var_array **scalars_tend**

| | |
|--------------|---|
| Description | <i>Tendency of cloud water number concentration multiplied by dry air density divided by d(zeta)/dz</i> |
| Units | <i>nb m⁻³ s⁻¹</i> |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

tend_nifa — constituent field of var_array **scalars_tend**

| | |
|--------------|--|
| Description | <i>Tendency of ice-friendly aerosol number concentration multiplied by dry air density divided by d(zeta)/dz</i> |
| Units | <i>nb m⁻³ s⁻¹</i> |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

tend_nwfa — constituent field of var_array **scalars_tend**

| | |
|--------------|--|
| Description | <i>Tendency of water-friendly aerosol number concentration multiplied by dry air density divided by d(zeta)/dz</i> |
| Units | <i>nb m⁻³ s⁻¹</i> |
| Array Group | number |
| Allocated by | mp_thompson_aers_in |

scale (real) (nVertLevels, TWO, nCells)

| | |
|------------------|---|
| Units | - |
| Description | <i>Scaling coefficients for monotonic-limited horizontal fluxes</i> |
| Accessed in code | as ‘scale’ from the ‘halo_scratch’ pool |

seoice (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>sea-ice flag (0=no seaice; =1 otherwise)</i> |
| Accessed in code | as ‘seoice’ from the ‘sfc_input’ pool |

sfc_albbck (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>background surface albedo</i> |
| Accessed in code | as ‘sfc_albbck’ from the ‘sfc_input’ pool |

sfc_albedo (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>surface albedo</i> |
| Accessed in code | as ‘sfc_albedo’ from the ‘diag_physics’ pool |

sfc_emibck (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>background surface emissivity</i> |
| Accessed in code | as ‘sfc_emibck’ from the ‘diag_physics’ pool |

sfc_emiss (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>surface emissivity</i> |
| Accessed in code | as ‘sfc_emiss’ from the ‘diag_physics’ pool |

sfcrunoff (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>surface runoff</i> |
| Accessed in code | as ‘sfcrunoff’ from the ‘diag_physics’ pool |

sh2o (real) (nSoilLevels, nCells, Time)

| | |
|------------------|-------------------------------------|
| Units | <i>m3 m^-3</i> |
| Description | <i>soil equivalent liquid water</i> |
| Accessed in code | as ‘sh2o’ from the ‘sfc_input’ pool |

sh3d (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>stability function for heat</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘sh3d’ from the ‘diag_physics’ pool |

shbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>sensible heat flux: bare ground to atmosphere</i> |
| Accessed in code | as ‘shbxy’ from the ‘output_noahmp’ pool |

shcxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>sensible heat flux: leaf to canopy</i> |
| Accessed in code | as ‘shcxy’ from the ‘output_noahmp’ pool |

shdmax (real) (nCells)

| | |
|------------------|--|
| Units | <i>percent</i> |
| Description | <i>maximum fractional coverage of annual green vegetation fraction</i> |
| Accessed in code | as ‘shdmax’ from the ‘sfc_input’ pool |

shdmin (real) (nCells)

| | |
|------------------|--|
| Units | <i>percent</i> |
| Description | <i>minimum fractional coverage of annual green vegetation fraction</i> |
| Accessed in code | as ‘shdmin’ from the ‘sfc_input’ pool |

shgxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>sensible heat flux: ground to canopy</i> |
| Accessed in code | as ‘shgxy’ from the ‘output_noahmp’ pool |

sifract (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>sprinkler irrigation fraction</i> |
| Accessed in code | as ‘sifract’ from the ‘diag_physics_noahmp’ pool |

skintemp (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>ground or water surface temperature</i> |
| Accessed in code | as ‘skintemp’ from the ‘sfc_input’ pool |

sm3d (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>stability function for moisture</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘sm3d’ from the ‘diag_physics’ pool |

smcrel (real) (nSoilLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m^3 m^{-3}$ |
| Description | <i>soil moisture threshold below which transpiration begins to stress</i> |
| Accessed in code | as ‘smcrel’ from the ‘sfc_input’ pool |

sмоis (real) (nSoilLevels, nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | $m^3 m^{-3}$ |
| Description | <i>soil moisture</i> |
| Accessed in code | as ‘smois’ from the ‘sfc_input’ pool |

smstav (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>surface moisture availability</i> |
| Accessed in code | as ‘smstav’ from the ‘diag_physics’ pool |

smstot (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m^3 m^{-3}$ |
| Description | <i>total soil mositure</i> |
| Accessed in code | as ‘smstot’ from the ‘diag_physics’ pool |

sneqvoxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>mm</i> |
| Description | <i>snow mass at last time step</i> |
| Accessed in code | as ‘sneqvoxy’ from the ‘diag_physics_noahmp’ pool |

snicexy (real) (nSnowLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>snow layer ice</i> |
| Accessed in code | as ‘snicexy’ from the ‘diag_physics_noahmp’ pool |

snliqxy (real) (nSnowLevels, nCells, Time)

| | |
|------------------|--|
| Units | <i>mm</i> |
| Description | <i>snow layer liquid</i> |
| Accessed in code | as ‘snliqxy’ from the ‘diag_physics_noahmp’ pool |

snoalb (real) (nCells)

| | |
|------------------|---------------------------------------|
| Units | <i>unitless</i> |
| Description | <i>annual maximum snow albedo</i> |
| Accessed in code | as ‘snoalb’ from the ‘sfc_input’ pool |

snopcx (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>W m⁻²</i> |
| Description | <i>snow phase change heat flux</i> |
| Accessed in code | as ‘snopcx’ from the ‘diag_physics’ pool |

snotime (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>initial number of time-steps since last snow fall</i> |
| Accessed in code | as ‘snotime’ from the ‘diag_physics’ pool |

snow (real) (nCells, Time)

| | |
|------------------|-------------------------------------|
| Units | <i>kg m⁻²</i> |
| Description | <i>snow water equivalent</i> |
| Accessed in code | as ‘snow’ from the ‘sfc_input’ pool |

snowc (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>flag for snow on ground (=0 no snow; =1,otherwise)</i> |
| Accessed in code | as ‘snowc’ from the ‘sfc_input’ pool |

snowenergy (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kJ\ m^{-2}$ |
| Description | <i>energy content in snow relative to 273.16</i> |
| Accessed in code | as ‘snowenergy’ from the ‘output_noahmp’ pool |

snowh (real) (nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | m |
| Description | <i>physical snow depth</i> |
| Accessed in code | as ‘snowh’ from the ‘sfc_input’ pool |

snowlsm (real) (nCells, Time)

| | |
|------------------|--|
| Units | mm |
| Description | <i>accumulated frozen precipitation into LSM</i> |
| Accessed in code | as ‘snowlsm’ from the ‘output_noahmp’ pool |

snownc (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>accumulated grid-scale precipitation of snow</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘snownc’ from the ‘diag_physics’ pool |

snowncv (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>time-step grid-scale precipitation of snow</i> |
| Allocated by | mp_thompson_in, mp_thompson_aers_in, mp_wsm6_in |
| Accessed in code | as ‘snowncv’ from the ‘diag_physics’ pool |

soilcl1 (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>soil texture class level 1 needed as input for the NOAH-MP land surface model</i> |
| Allocated by | sf_noahmp_in |
| Accessed in code | as ‘soilcl1’ from the ‘mesh’ pool |

soilcl2 (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>soil texture class level 2 needed as input for the NOAH-MP land surface model</i> |
| Allocated by | sf_noahmp_in |
| Accessed in code | as ‘soilcl2’ from the ‘mesh’ pool |

soilcl3 (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>soil texture class level 3 needed as input for the NOAH-MP land surface model</i> |
| Allocated by | sf_noahmp_in |
| Accessed in code | as ‘soilcl3’ from the ‘mesh’ pool |

soilcl4 (real) (nCells)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>soil texture class level 4 needed as input for the NOAH-MP land surface model</i> |
| Allocated by | sf_noahmp_in |
| Accessed in code | as ‘soilcl4’ from the ‘mesh’ pool |

soilcomp (real) (nSoilComps, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>soil composition needed as input in the NOAH-MP land surface model</i> |
| Allocated by | sf_noahmp_in |
| Accessed in code | as ‘soilcomp’ from the ‘mesh’ pool |

soilenergy (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>kJ m⁻²</i> |
| Description | <i>energy content in soil relative to 273.16</i> |
| Accessed in code | as ‘soilenergy’ from the ‘output_noahmp’ pool |

spechum (real) (nVertLevels, nCells, Time)

| | |
|------------------|-----------------------------------|
| Units | $kg\ kg^{-1}$ |
| Description | <i>Specific humidity</i> |
| Allocated by | jedi_da |
| Accessed in code | as ‘spechum’ from the ‘diag’ pool |

specZoneMaskCell (real) (nCells)

| | |
|------------------|---|
| Units | - |
| Description | <i>0/1 mask on cells, defined as 1 for cells in the limited-area specified zone</i> |
| Accessed in code | as ‘specZoneMaskCell’ from the ‘mesh’ pool |

specZoneMaskEdge (real) (nEdges)

| | |
|------------------|---|
| Units | - |
| Description | <i>0/1 mask on edges, defined as 1 for edges in the limited-area specified zone</i> |
| Accessed in code | as ‘specZoneMaskEdge’ from the ‘mesh’ pool |

specZoneMaskVertex (real) (nVertices)

| | |
|------------------|---|
| Units | - |
| Description | <i>0/1 mask on vertices, defined as 1 for vertices in the limited-area specified zone</i> |
| Accessed in code | as ‘specZoneMaskVertex’ from the ‘mesh’ pool |

sr (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>time-step ratio of frozen versus total grid-scale precipitation</i> |
| Accessed in code | as ‘sr’ from the ‘diag_physics’ pool |

srh_0_1km (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m^2\ s^{-2}$ |
| Description | <i>Storm relative helicity, 0-1 km AGL</i> |
| Accessed in code | as ‘srh_0_1km’ from the ‘diag’ pool |

srh_0_3km (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m^2 s^{-2}$ |
| Description | <i>Storm relative helicity, 0-3 km AGL</i> |
| Accessed in code | as ‘srh_0_3km’ from the ‘diag’ pool |

sslt — see ‘aerosols’**sst** (real) (nCells, Time)

| | |
|------------------|------------------------------------|
| Units | K |
| Description | <i>sea-surface temperature</i> |
| Accessed in code | as ‘sst’ from the ‘sfc_input’ pool |

sstsk (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>skin sea-surface temperature</i> |
| Accessed in code | as ‘sstsk’ from the ‘diag_physics’ pool |

sstsk_dtc (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>skin sea-surface temperature cooling</i> |
| Accessed in code | as ‘sstsk_dtc’ from the ‘diag_physics’ pool |

sstsk_dtw (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>skin sea-surface temperature warming</i> |
| Accessed in code | as ‘sstsk_dtw’ from the ‘diag_physics’ pool |

stblcpxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $g m^{-2}$ |
| Description | <i>stable carbon pool</i> |
| Accessed in code | as ‘stblcpxy’ from the ‘diag_physics_noahmp’ pool |

stmassxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $g\ m^{-2}$ |
| Description | <i>stem mass</i> |
| Accessed in code | as ‘stmassxy’ from the ‘diag_physics_noahmp’ pool |

stream_function (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m^2\ s^{-1}$ |
| Description | <i>Stream function</i> |
| Allocated by | jedi_da |
| Accessed in code | as ‘stream_function’ from the ‘diag’ pool |

sub_qv (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ kg^{-1}\ s^{-1}$ |
| Description | <i>EDMF subsidence of water vapor mixing ratio in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘sub_qv’ from the ‘diag_physics’ pool |

sub_thl (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $K\ s^{-1}$ |
| Description | <i>EDMF subsidence of liquid-ice potential temperature in the MYNN PBL scheme</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘sub_thl’ from the ‘diag_physics’ pool |

sul — see ‘aerosols’

surface_pressure (real) (nCells, Time)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>Diagnosed surface pressure</i> |
| Accessed in code | as ‘surface_pressure’ from the ‘diag’ pool |

swcf (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>top-of-atmosphere cloud shortwave radiative forcing</i> |
| Accessed in code | as ‘swcf’ from the ‘diag_physics’ pool |

swddif (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>shortwave surface downward diffuse irradiance</i> |
| Accessed in code | as ‘swddif’ from the ‘diag_physics’ pool |

swddir (real) (nCells, Time)

| | |
|------------------|---|
| Units | $W\ m^{-2}$ |
| Description | <i>shortwave surface downward direct irradiance</i> |
| Accessed in code | as ‘swddir’ from the ‘diag_physics’ pool |

swddni (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>shortwave surface downward direct normal irradiance</i> |
| Accessed in code | as ‘swddni’ from the ‘diag_physics’ pool |

swdnb (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky downward surface shortwave radiation flux</i> |
| Accessed in code | as ‘swdnb’ from the ‘diag_physics’ pool |

swdnbc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky downward surface shortwave radiation flux</i> |
| Accessed in code | as ‘swdnbc’ from the ‘diag_physics’ pool |

swdnflux (real) (nVertLevelsP2, nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | - |
| Accessed in code | as ‘swdnflux’ from the ‘diag_physics’ pool |

swdnflxc (real) (nVertLevelsP2, nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | - |
| Accessed in code | as ‘swdnflxc’ from the ‘diag_physics’ pool |

swdnt (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky downward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘swdnt’ from the ‘diag_physics’ pool |

swdntc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky downward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘swdntc’ from the ‘diag_physics’ pool |

swupb (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky upward surface shortwave radiation flux</i> |
| Accessed in code | as ‘swupb’ from the ‘diag_physics’ pool |

swupbc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky upward surface shortwave radiation flux</i> |
| Accessed in code | as ‘swupbc’ from the ‘diag_physics’ pool |

swupflx (real) (nVertLevelsP2, nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | - |
| Accessed in code | as ‘swupflx’ from the ‘diag_physics’ pool |

swupflxc (real) (nVertLevelsP2, nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | - |
| Accessed in code | as ‘swupflxc’ from the ‘diag_physics’ pool |

swupt (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>all-sky upward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘swupt’ from the ‘diag_physics’ pool |

swuptc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $W\ m^{-2}$ |
| Description | <i>clear-sky upward top-of-atmosphere shortwave radiation flux</i> |
| Accessed in code | as ‘swuptc’ from the ‘diag_physics’ pool |

t2m (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | K |
| Description | <i>2-meter temperature</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘t2m’ from the ‘diag_physics’ pool |

t2mbxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>2-meter temperature over bare ground</i> |
| Accessed in code | as ‘t2mbxy’ from the ‘output_noahmp’ pool |

t2mvxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>2-meter temperature over canopy</i> |
| Accessed in code | as ‘t2mvxy’ from the ‘output_noahmp’ pool |

t2mxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>grid-mean 2-meter temperature</i> |
| Accessed in code | as ‘t2mxy’ from the ‘output_noahmp’ pool |

t_init (real) (nVertLevels, nCells)

| | |
|------------------|----------------------------------|
| Units | K |
| Description | <i>theta reference profile</i> |
| Accessed in code | as ‘t_init’ from the ‘mesh’ pool |

t_iso_levels (real) (nIsoLevelsT)

| | |
|------------------|--|
| Units | Pa |
| Description | <i>Levels for vertical interpolation of temperature to isobaric surfaces</i> |
| Accessed in code | as ‘t_iso_levels’ from the ‘diag’ pool |

t_isobaric (real) (nIsoLevelsT, nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Temperature interpolated to isobaric surfaces defined in t_iso_levels</i> |
| Accessed in code | as ‘t_isobaric’ from the ‘diag’ pool |

t_oml (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>ocean mixed layer temperature</i> |
| Accessed in code | as ‘t_oml’ from the ‘diag_physics’ pool |

t_oml_200m_initial (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>ocean mixed layer 200 m mean temperature at initial time</i> |
| Accessed in code | as ‘t_oml_200m_initial’ from the ‘diag_physics’ pool |

t_oml_initial (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>ocean mixed layer temperature at initial time</i> |
| Accessed in code | as ‘t_oml_initial’ from the ‘diag_physics’ pool |

tahxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>canopy air temperature</i> |
| Accessed in code | as ‘tahxy’ from the ‘diag_physics_noahmp’ pool |

taod5502d (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>total aerosol optical depth at 550 nm from the Thompson cloud micro-physics</i> |
| Allocated by | mp_thompson_aers_in |
| Accessed in code | as ‘taod5502d’ from the ‘diag_physics’ pool |

taod5503d (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>aerosol optical depth at 550 nm from the Thompson cloud microphysics</i> |
| Allocated by | mp_thompson_aers_in |
| Accessed in code | as ‘taod5503d’ from the ‘diag_physics’ pool |

taussxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | - |
| Description | <i>non-dimensional snow age</i> |
| Accessed in code | as ‘taussxy’ from the ‘diag_physics_noahmp’ pool |

tday_accum (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>K</i> |
| Description | <i>accumulated daily surface temperature for current day</i> |
| Accessed in code | as ‘tday_accum’ from the ‘diag_physics’ pool |

temperature (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | <i>K</i> |
| Description | <i>temperature</i> |
| Allocated by | jedi_da |
| Accessed in code | as ‘temperature’ from the ‘diag’ pool |

temperature_100hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>K</i> |
| Description | <i>Temperature vertically interpolated to 100 hPa</i> |
| Accessed in code | as ‘temperature_100hPa’ from the ‘diag’ pool |

temperature_200hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Temperature vertically interpolated to 200 hPa</i> |
| Accessed in code | as ‘temperature_200hPa’ from the ‘diag’ pool |

temperature_250hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Temperature vertically interpolated to 250 hPa</i> |
| Accessed in code | as ‘temperature_250hPa’ from the ‘diag’ pool |

temperature_500hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Temperature vertically interpolated to 500 hPa</i> |
| Accessed in code | as ‘temperature_500hPa’ from the ‘diag’ pool |

temperature_50hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Temperature vertically interpolated to 50 hPa</i> |
| Accessed in code | as ‘temperature_50hPa’ from the ‘diag’ pool |

temperature_700hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Temperature vertically interpolated to 700 hPa</i> |
| Accessed in code | as ‘temperature_700hPa’ from the ‘diag’ pool |

temperature_850hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Temperature vertically interpolated to 850 hPa</i> |
| Accessed in code | as ‘temperature_850hPa’ from the ‘diag’ pool |

temperature_925hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Temperature vertically interpolated to 925 hPa</i> |
| Accessed in code | as ‘temperature_925hPa’ from the ‘diag’ pool |

temperature_surface (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Temperature at midpoint of lowest model layer</i> |
| Accessed in code | as ‘temperature_surface’ from the ‘diag’ pool |

tend_nc — see ‘*scalars_tend*’

tend_ni — see ‘*scalars_tend*’

tend_nifa — see ‘*scalars_tend*’

tend_nr — see ‘*scalars_tend*’

tend_nwfa — see ‘*scalars_tend*’

tend_qc — see ‘*scalars_tend*’

tend_qg — see ‘*scalars_tend*’

tend_qi — see ‘*scalars_tend*’

tend_qr — see ‘*scalars_tend*’

tend_qs — see ‘*scalars_tend*’

tend_qv — see ‘*scalars_tend*’

tend_rho (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Description | <i>Tendency of dry density from dynamics</i> |
| Accessed in code | as ‘rho_zz’ from the ‘tend’ pool |

tend_rho_physics (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-3}\ s^{-1}$ |
| Description | <i>Total dry air tendency from physics</i> |
| Accessed in code | as ‘tend_rho_physics’ from the ‘tend_physics’ pool |

tend_rtheta_physics (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ K\ m^{-3}$ |
| Description | <i>Total rho*theta/zz tendency from physics</i> |
| Accessed in code | as ‘tend_rtheta_physics’ from the ‘tend_physics’ pool |

tend_ru_physics (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>Total horizontal momentum tendency from physics</i> |
| Accessed in code | as ‘tend_ru_physics’ from the ‘tend_physics’ pool |

tend_sfc_pressure (real) (nCells, Time)

| | |
|------------------|---|
| Units | $Pa\ s^{-1}$ |
| Description | <i>Tendency of surface pressure</i> |
| Accessed in code | as ‘tend_sfc_pressure’ from the ‘tend’ pool |

tend_theta (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $kg\ K\ m^{-3}\ s^{-1}$ |
| Description | <i>tendency of coupled potential temperature rho*theta_m/zz from dynamics and physics, updated each RK step</i> |
| Accessed in code | as ‘theta_m’ from the ‘tend’ pool |

tend_u (real) (nVertLevels, nEdges, Time)

| | |
|------------------|------------------------------------|
| Units | $m s^{-2}$ |
| Description | <i>Tendency of u from dynamics</i> |
| Accessed in code | as ‘u’ from the ‘tend’ pool |

tend_u_phys (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>Normal wind tendencies from physics parameterizations</i> |
| Accessed in code | as ‘tend_u_phys’ from the ‘diag’ pool |

tend_umerid (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1} s^{-1}$ |
| Description | <i>Total cell-centered meridional wind tendency from physics</i> |
| Accessed in code | as ‘tend_umerid’ from the ‘tend_physics’ pool |

tend_uzonal (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1} s^{-1}$ |
| Description | <i>Total cell-centered zonal wind tendency from physics</i> |
| Accessed in code | as ‘tend_uzonal’ from the ‘tend_physics’ pool |

tend_w (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|------------------------------------|
| Units | $m s^{-2}$ |
| Description | <i>Tendency of w from dynamics</i> |
| Accessed in code | as ‘w’ from the ‘tend’ pool |

tend_w_buoy (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-2}$ |
| Description | <i>Tendency of w due to buoyancy force</i> |
| Accessed in code | as ‘w_buoy’ from the ‘tend’ pool |

tend_w_pgf (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-2}$ |
| Description | <i>Tendency of w due to pressure gradient force</i> |
| Accessed in code | as ‘w_pgf’ from the ‘tend’ pool |

ter (real) (nCells)

| | |
|------------------|------------------------------------|
| Units | m |
| Description | <i>terrain height</i> |
| Accessed in code | as ‘ter’ from the ‘sfc_input’ pool |

tgbxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>bare ground temperature</i> |
| Accessed in code | as ‘tgbxy’ from the ‘output_noahmp’ pool |

tgvxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>ground temperature under canopy</i> |
| Accessed in code | as ‘tgvxy’ from the ‘output_noahmp’ pool |

tgxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>bulk ground temperature</i> |
| Accessed in code | as ‘tgxy’ from the ‘diag_physics_noahmp’ pool |

th2m (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>2-meter potential temperature</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘th2m’ from the ‘diag_physics’ pool |

thc (real) (nCells, Time)

| | |
|------------------|--|
| Units | $\text{Cal cm}^{-2} \text{K}^{-1} \text{s}^{-0.5}$ |
| Description | <i>thermal inertia</i> |
| Accessed in code | as ‘thc’ from the ‘diag_physics’ pool |

theta (real) (nVertLevels, nCells, Time)

| | |
|------------------|---------------------------------|
| Units | K |
| Description | <i>Potential temperature</i> |
| Accessed in code | as ‘theta’ from the ‘diag’ pool |

theta_amb (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Potential temperature increment</i> |
| Accessed in code | as ‘theta’ from the ‘tend_iau’ pool |

theta_base (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Base state potential temperature</i> |
| Accessed in code | as ‘theta_base’ from the ‘diag’ pool |

theta_m (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>Moist potential temperature: theta*(1+q_v*R_v/R_d)</i> |
| Accessed in code | as ‘theta_m’ from the ‘state’ pool |

theta_pv (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>Potential temperature on dynamic tropopause</i> |
| Accessed in code | as ‘theta_pv’ from the ‘diag’ pool |

Time (real) (Time)

| | |
|------------------|--|
| Units | <i>seconds since YYYY-MM-DD hh:mm:ss</i> |
| Description | <i>CF-compliant valid time</i> |
| Accessed in code | as ‘Time’ from the ‘state’ pool |

tke_pbl (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m^2 s^{-2}$ |
| Description | <i>turbulent kinetic energy from PBL</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘tke_pbl’ from the ‘diag_physics’ pool |

tlag (real) (nLags, nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>daily mean surface temperature of prior days</i> |
| Accessed in code | as ‘tlag’ from the ‘diag_physics’ pool |

tmn (real) (nCells, Time)

| | |
|------------------|------------------------------------|
| Units | K |
| Description | <i>deep soil temperature</i> |
| Accessed in code | as ‘tmn’ from the ‘sfc_input’ pool |

tradxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>surface radiative temperature</i> |
| Accessed in code | as ‘tradxy’ from the ‘output_noahmp’ pool |

trxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $mm s^{-1}$ |
| Description | <i>canopy rain interception ratio</i> |
| Accessed in code | as ‘trxy’ from the ‘output_noahmp’ pool |

tslb (real) (nSoilLevels, nCells, Time)

| | |
|------------------|-------------------------------------|
| Units | K |
| Description | <i>soil layer temperature</i> |
| Accessed in code | as ‘tslb’ from the ‘sfc_input’ pool |

tsnoxy (real) (nSnowLevels, nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>snow temperature</i> |
| Accessed in code | as ‘tsnoxy’ from the ‘diag_physics_noahmp’ pool |

tsq (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | K^2 |
| Description | <i>liquid water potential temperature variance</i> |
| Allocated by | bl_mynn_in |
| Accessed in code | as ‘tsq’ from the ‘diag_physics’ pool |

tvxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | K |
| Description | <i>vegetation leaf temperature</i> |
| Accessed in code | as ‘tvxy’ from the ‘diag_physics_noahmp’ pool |

tyear_accum (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>accumulated yearly surface temperature for current year</i> |
| Accessed in code | as ‘tyear_accum’ from the ‘diag_physics’ pool |

tyear_mean (real) (nCells, Time)

| | |
|------------------|--|
| Units | K |
| Description | <i>annual mean surface temperature</i> |
| Accessed in code | as ‘tyear_mean’ from the ‘diag_physics’ pool |

u (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Horizontal normal velocity at edges</i> |
| Accessed in code | as ‘u’ from the ‘state’ pool |

u10 (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $m s^{-1}$ |
| Description | <i>10-meter zonal wind</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘u10’ from the ‘diag_physics’ pool |

u_amb (real) (nVertLevels, nEdges, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Horizontal normal velocity increment</i> |
| Accessed in code | as ‘u’ from the ‘tend_iau’ pool |

u_init (real) (nVertLevels)

| | |
|------------------|----------------------------------|
| Units | $m s^{-1}$ |
| Description | <i>u reference profile</i> |
| Accessed in code | as ‘u_init’ from the ‘mesh’ pool |

u_pv (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Zonal wind on dynamic tropopause</i> |
| Accessed in code | as ‘u_pv’ from the ‘diag’ pool |

udrunoff (real) (nCells, Time)

| | |
|------------------|--|
| Units | mm |
| Description | <i>underground runoff</i> |
| Accessed in code | as ‘udrunoff’ from the ‘diag_physics’ pool |

umeridional_100hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 100 hPa</i> |
| Accessed in code | as ‘umeridional_100hPa’ from the ‘diag’ pool |

umeridional_1km (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Meridional wind component at 1 km AGL</i> |
| Accessed in code | as ‘umeridional_1km’ from the ‘diag’ pool |

umeridional_200hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 200 hPa</i> |
| Accessed in code | as ‘umeridional_200hPa’ from the ‘diag’ pool |

umeridional_250hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 250 hPa</i> |
| Accessed in code | as ‘umeridional_250hPa’ from the ‘diag’ pool |

umeridional_500hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 500 hPa</i> |
| Accessed in code | as ‘umeridional_500hPa’ from the ‘diag’ pool |

umeridional_50hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 50 hPa</i> |
| Accessed in code | as ‘umeridional_50hPa’ from the ‘diag’ pool |

umeridional_6km (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Meridional wind component at 6 km AGL</i> |
| Accessed in code | as ‘umeridional_6km’ from the ‘diag’ pool |

umeridional_700hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 700 hPa</i> |
| Accessed in code | as ‘umeridional_700hPa’ from the ‘diag’ pool |

umeridional_850hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 850 hPa</i> |
| Accessed in code | as ‘umeridional_850hPa’ from the ‘diag’ pool |

umeridional_925hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed meridional wind at cell centers, vertically interpolated to 925 hPa</i> |
| Accessed in code | as ‘umeridional_925hPa’ from the ‘diag’ pool |

umeridional_surface (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Meridional wind component at midpoint of lowest model layer</i> |
| Accessed in code | as ‘umeridional_surface’ from the ‘diag’ pool |

updraft_helicity_max (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m^2 s^{-2}$ |
| Description | <i>Maximum updraft helicity since last output</i> |
| Accessed in code | as ‘updraft_helicity_max’ from the ‘diag’ pool |

uReconstructMeridional (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Meridional component of reconstructed horizontal velocity at cell centers</i> |
| Accessed in code | as ‘uReconstructMeridional’ from the ‘diag’ pool |

uReconstructX (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Cartesian x-component of reconstructed horizontal velocity at cell centers</i> |
| Accessed in code | as ‘uReconstructX’ from the ‘diag’ pool |

uReconstructY (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Cartesian y-component of reconstructed horizontal velocity at cell centers</i> |
| Accessed in code | as ‘uReconstructY’ from the ‘diag’ pool |

uReconstructZ (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Cartesian z-component of reconstructed horizontal velocity at cell centers</i> |
| Accessed in code | as ‘uReconstructZ’ from the ‘diag’ pool |

uReconstructZonal (real) (nVertLevels, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Zonal component of reconstructed horizontal velocity at cell centers</i> |
| Accessed in code | as ‘uReconstructZonal’ from the ‘diag’ pool |

ust (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>U^* in similarity theory</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘ust’ from the ‘diag_physics’ pool |

ustm (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>U^* in similarity theory without vconv</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘ustm’ from the ‘diag_physics’ pool |

uzonal_100hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 100 hPa</i> |
| Accessed in code | as ‘uzonal_100hPa’ from the ‘diag’ pool |

uzonal_1km (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Zonal wind component at 1 km AGL</i> |
| Accessed in code | as ‘uzonal_1km’ from the ‘diag’ pool |

uzonal_200hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 200 hPa</i> |
| Accessed in code | as ‘uzonal_200hPa’ from the ‘diag’ pool |

uzonal_250hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 250 hPa</i> |
| Accessed in code | as ‘uzonal_250hPa’ from the ‘diag’ pool |

uzonal_500hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 500 hPa</i> |
| Accessed in code | as ‘uzonal_500hPa’ from the ‘diag’ pool |

uzonal_50hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 50 hPa</i> |
| Accessed in code | as ‘uzonal_50hPa’ from the ‘diag’ pool |

uzonal_6km (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Zonal wind component at 6 km AGL</i> |
| Accessed in code | as ‘uzonal_6km’ from the ‘diag’ pool |

uzonal_700hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 700 hPa</i> |
| Accessed in code | as ‘uzonal_700hPa’ from the ‘diag’ pool |

uzonal_850hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 850 hPa</i> |
| Accessed in code | as ‘uzonal_850hPa’ from the ‘diag’ pool |

uzonal_925hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Reconstructed zonal wind at cell centers, vertically interpolated to 925 hPa</i> |
| Accessed in code | as ‘uzonal_925hPa’ from the ‘diag’ pool |

uzonal_surface (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Zonal wind component at midpoint of lowest model layer</i> |
| Accessed in code | as ‘uzonal_surface’ from the ‘diag’ pool |

v (real) (nVertLevels, nEdges, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Horizontal tangential velocity at edges</i> |
| Accessed in code | as ‘v’ from the ‘diag’ pool |

v10 (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | $m s^{-1}$ |
| Description | <i>10-meter meridional wind</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘v10’ from the ‘diag_physics’ pool |

v_init (real) (nVertLevels)

| | |
|------------------|----------------------------------|
| Units | $m s^{-1}$ |
| Description | <i>v reference profile</i> |
| Accessed in code | as ‘v_init’ from the ‘mesh’ pool |

v_pv (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Meridional wind on dynamic tropopause</i> |
| Accessed in code | as ‘v_pv’ from the ‘diag’ pool |

var2d (real) (nCells)

| | |
|------------------|--|
| Units | m |
| Description | <i>standard deviation of subgrid-scale orography</i> |
| Accessed in code | as ‘var2d’ from the ‘sfc_input’ pool |

var2dls (real) (nCells)

| | |
|------------------|---|
| Units | m |
| Description | <i>standard deviation of subgrid-scale orography (meso-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘var2dls’ from the ‘sfc_input’ pool |

var2dss (real) (nCells)

| | |
|------------------|--|
| Units | m |
| Description | <i>standard deviation of subgrid-scale orography (small-scale GWD)</i> |
| Allocated by | ugwp_orog_stream |
| Accessed in code | as ‘var2dss’ from the ‘sfc_input’ pool |

vegfra (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | <i>percent</i> |
| Description | <i>vegetation fraction</i> |
| Accessed in code | as ‘vegfra’ from the ‘sfc_input’ pool |

velocity_potential (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m^2 \ s^{-1}$ |
| Description | <i>Velocity potential</i> |
| Allocated by | jedi_da |
| Accessed in code | as ‘velocity_potential’ from the ‘diag’ pool |

verticesOnCell (integer) (maxEdges, nCells)

| | |
|------------------|--|
| Units | - |
| Description | <i>IDs of vertices (corner points) of a cell</i> |
| Accessed in code | as ‘verticesOnCell’ from the ‘mesh’ pool |

verticesOnEdge (integer) (TWO, nEdges)

| | |
|------------------|---|
| Units | - |
| Description | <i>IDs of the two vertex endpoints of an edge</i> |
| Accessed in code | as ‘verticesOnEdge’ from the ‘mesh’ pool |

volc — see ‘aerosols’

vort_pv (real) (nCells, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vertical vorticity on dynamic tropopause</i> |
| Accessed in code | as ‘vort_pv’ from the ‘diag’ pool |

vorticity (real) (nVertLevels, nVertices, Time)

| | |
|------------------|---------------------------------------|
| Units | s^{-1} |
| Description | <i>Relative vorticity at vertices</i> |
| Accessed in code | as ‘vorticity’ from the ‘diag’ pool |

vorticity_100hPa (real) (nVertices, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 100 hPa</i> |
| Accessed in code | as ‘vorticity_100hPa’ from the ‘diag’ pool |

vorticity_200hPa (real) (nVertices, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 200 hPa</i> |
| Accessed in code | as ‘vorticity_200hPa’ from the ‘diag’ pool |

vorticity_250hPa (real) (nVertices, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 250 hPa</i> |
| Accessed in code | as ‘vorticity_250hPa’ from the ‘diag’ pool |

vorticity_500hPa (real) (nVertices, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 500 hPa</i> |
| Accessed in code | as ‘vorticity_500hPa’ from the ‘diag’ pool |

vorticity_50hPa (real) (nVertices, Time)

| | |
|------------------|---|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 50 hPa</i> |
| Accessed in code | as ‘vorticity_50hPa’ from the ‘diag’ pool |

vorticity_700hPa (real) (nVertices, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 700 hPa</i> |
| Accessed in code | as ‘vorticity_700hPa’ from the ‘diag’ pool |

vorticity_850hPa (real) (nVertices, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 850 hPa</i> |
| Accessed in code | as ‘vorticity_850hPa’ from the ‘diag’ pool |

vorticity_925hPa (real) (nVertices, Time)

| | |
|------------------|--|
| Units | s^{-1} |
| Description | <i>Relative vorticity vertically interpolated to 925 hPa</i> |
| Accessed in code | as ‘vorticity_925hPa’ from the ‘diag’ pool |

w (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity at vertical cell faces</i> |
| Accessed in code | as ‘w’ from the ‘state’ pool |

w0avg (real) (nVertLevels, nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>time running-averaged vertical velocity</i> |
| Allocated by | cu_kain_fritsch_in |
| Accessed in code | as ‘w0avg’ from the ‘diag_physics’ pool |

w_100hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 100 hPa</i> |
| Accessed in code | as ‘w_100hPa’ from the ‘diag’ pool |

w_200hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 200 hPa</i> |
| Accessed in code | as ‘w_200hPa’ from the ‘diag’ pool |

w_250hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 250 hPa</i> |
| Accessed in code | as ‘w_250hPa’ from the ‘diag’ pool |

w_500hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 500 hPa</i> |
| Accessed in code | as ‘w_500hPa’ from the ‘diag’ pool |

w_50hPa (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 50 hPa</i> |
| Accessed in code | as ‘w_50hPa’ from the ‘diag’ pool |

w_700hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 700 hPa</i> |
| Accessed in code | as ‘w_700hPa’ from the ‘diag’ pool |

w_850hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 850 hPa</i> |
| Accessed in code | as ‘w_850hPa’ from the ‘diag’ pool |

w_925hPa (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Vertical velocity vertically interpolated to 925 hPa</i> |
| Accessed in code | as ‘w_925hPa’ from the ‘diag’ pool |

w_velocity_max (real) (nCells, Time)

| | |
|------------------|--|
| Units | $m s^{-1}$ |
| Description | <i>Maximum column w velocity since last output</i> |
| Accessed in code | as ‘w_velocity_max’ from the ‘diag’ pool |

waxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>water in the aquifer</i> |
| Accessed in code | as ‘waxy’ from the ‘diag_physics_noahmp’ pool |

weightsOnEdge (real) (maxEdges2, nEdges)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>Weights used in reconstruction of tangential velocity for an edge</i> |
| Accessed in code | as ‘weightsOnEdge’ from the ‘mesh’ pool |

wgapxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>within canopy gap</i> |
| Accessed in code | as ‘wgapxy’ from the ‘output_noahmp’ pool |

wind_speed_level1_max (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m s^{-1}$ |
| Description | <i>Maximum wind speed in lowest model level since last output</i> |
| Accessed in code | as ‘wind_speed_level1_max’ from the ‘diag’ pool |

woodxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | $g m^{-2}$ |
| Description | <i>mass of wood</i> |
| Accessed in code | as ‘woodxy’ from the ‘diag_physics_noahmp’ pool |

wslakexy (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>lake water storage</i> |
| Accessed in code | as ‘wslakexy’ from the ‘diag_physics_noahmp’ pool |

wspd (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>wind speed at lowest model level</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘wspd’ from the ‘diag_physics’ pool |

wstar (real) (nCells, Time)

| | |
|------------------|---|
| Units | $m\ s^{-1}$ |
| Description | <i>mixed velocity scale from PBL scheme</i> |
| Allocated by | bl_ysu_in |
| Accessed in code | as ‘wstar’ from the ‘diag_physics’ pool |

wtxy (real) (nCells, Time)

| | |
|------------------|---|
| Units | mm |
| Description | <i>groundwater storage</i> |
| Accessed in code | as ‘wtxy’ from the ‘diag_physics_noahmp’ pool |

wwAvg (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>time-averaged rho*omega/zz used in scalar transport</i> |
| Accessed in code | as ‘wwAvg’ from the ‘diag’ pool |

wwAvg_split (real) (nVertLevelsP1, nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>time-averaged rho*omega/zz used in scalar transport</i> |
| Accessed in code | as ‘wwAvg_split’ from the ‘diag’ pool |

xCell (real) (nCells)

| | |
|------------------|--|
| Units | m |
| Description | <i>Cartesian x-coordinate of cells</i> |
| Accessed in code | as ‘xCell’ from the ‘mesh’ pool |

xEdge (real) (nEdges)

| | |
|------------------|--|
| Units | m |
| Description | <i>Cartesian x-coordinate of edges</i> |
| Accessed in code | as ‘xEdge’ from the ‘mesh’ pool |

xice (real) (nCells, Time)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>fractional area coverage of sea-ice</i> |
| Accessed in code | as ‘xice’ from the ‘sfc_input’ pool |

xicem (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>sea-ice flag from previous time-step</i> |
| Accessed in code | as ‘xicem’ from the ‘diag_physics’ pool |

xland (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>land-ocean mask (1=land including sea-ice ; 2=ocean)</i> |
| Accessed in code | as ‘xland’ from the ‘sfc_input’ pool |

xmb_shallow (real) (nCells, Time)

| | |
|------------------|--|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>cloud base mass flux for shallow convection</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘xmb_shallow’ from the ‘diag_physics’ pool |

xmb_total (real) (nCells, Time)

| | |
|------------------|---|
| Units | $kg\ m^{-2}\ s^{-1}$ |
| Description | <i>cloud-base mass flux</i> |
| Allocated by | cu_grell_freitas_in |
| Accessed in code | as ‘xmb_total’ from the ‘diag_physics’ pool |

xsaixy (real) (nCells, Time)

| | |
|------------------|---|
| Units | - |
| Description | <i>stem area index</i> |
| Accessed in code | as ‘xsaixy’ from the ‘diag_physics_noahmp’ pool |

xtime (text) (Time)

| | |
|------------------|----------------------------------|
| Units | <i>YYYY-MM-DD hh:mm:ss</i> |
| Description | <i>Model valid time</i> |
| Accessed in code | as ‘xtime’ from the ‘state’ pool |

xVertex (real) (nVertices)

| | |
|------------------|---|
| Units | <i>m</i> |
| Description | <i>Cartesian x-coordinate of vertices</i> |
| Accessed in code | as ‘xVertex’ from the ‘mesh’ pool |

yCell (real) (nCells)

| | |
|------------------|--|
| Units | <i>m</i> |
| Description | <i>Cartesian y-coordinate of cells</i> |
| Accessed in code | as ‘yCell’ from the ‘mesh’ pool |

yEdge (real) (nEdges)

| | |
|------------------|--|
| Units | <i>m</i> |
| Description | <i>Cartesian y-coordinate of edges</i> |
| Accessed in code | as ‘yEdge’ from the ‘mesh’ pool |

yVertex (real) (nVertices)

| | |
|------------------|---|
| Units | <i>m</i> |
| Description | <i>Cartesian y-coordinate of vertices</i> |
| Accessed in code | as ‘yVertex’ from the ‘mesh’ pool |

z0 (real) (nCells, Time)

| | |
|------------------|--------------------------------------|
| Units | <i>m</i> |
| Description | <i>roughness height</i> |
| Accessed in code | as ‘z0’ from the ‘diag_physics’ pool |

z_iso_levels (real) (nIsoLevelsZ)

| | |
|------------------|---|
| Units | Pa |
| Description | <i>Levels for vertical interpolation of height to isobaric surfaces</i> |
| Accessed in code | as ‘z_iso_levels’ from the ‘diag’ pool |

z_isobaric (real) (nIsoLevelsZ, nCells, Time)

| | |
|------------------|---|
| Units | m |
| Description | <i>Height interpolated to isobaric surfaces defined in z_iso_levels</i> |
| Accessed in code | as ‘z_isobaric’ from the ‘diag’ pool |

zb (real) (nVertLevelsP1, TWO, nEdges)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Coefficients for contribution from u to omega diagnosis, edge-oriented</i> |
| Accessed in code | as ‘zb’ from the ‘mesh’ pool |

zb3 (real) (nVertLevelsP1, TWO, nEdges)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Coefficients for 3rd-order correction to contribution from u to omega diagnosis, edge-oriented</i> |
| Accessed in code | as ‘zb3’ from the ‘mesh’ pool |

zb3_cell (real) (nVertLevelsP1, maxEdges, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Coefficients for 3rd-order correction to contribution from u to omega diagnosis, cell-oriented</i> |
| Accessed in code | as ‘zb3_cell’ from the ‘mesh’ pool |

zb_cell (real) (nVertLevelsP1, maxEdges, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>Coefficients for contribution from u to omega diagnosis, cell-oriented</i> |
| Accessed in code | as ‘zb_cell’ from the ‘mesh’ pool |

zCell (real) (nCells)

| | |
|------------------|--|
| Units | m |
| Description | <i>Cartesian z-coordinate of cells</i> |
| Accessed in code | as ‘zCell’ from the ‘mesh’ pool |

zEdge (real) (nEdges)

| | |
|------------------|--|
| Units | m |
| Description | <i>Cartesian z-coordinate of edges</i> |
| Accessed in code | as ‘zEdge’ from the ‘mesh’ pool |

zgrid (real) (nVertLevelsP1, nCells)

| | |
|------------------|---|
| Units | m MSL |
| Description | <i>Geometric height of layer interfaces</i> |
| Accessed in code | as ‘zgrid’ from the ‘mesh’ pool |

znt (real) (nCells, Time)

| | |
|------------------|---------------------------------------|
| Units | m |
| Description | <i>roughness length</i> |
| Accessed in code | as ‘znt’ from the ‘diag_physics’ pool |

zol (real) (nCells, Time)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>z/L height over Monin-Obukhov length</i> |
| Allocated by | bl_mynn_in, bl_ysu_in |
| Accessed in code | as ‘zol’ from the ‘diag_physics’ pool |

zs (real) (nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>depth of centers of soil layers</i> |
| Accessed in code | as ‘zs’ from the ‘diag_physics’ pool |

zsnsoxy (real) (nzSnowLevels, nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>layer-bottom depth from snow surf</i> |
| Accessed in code | as ‘zsnsoxy’ from the ‘diag_physics_noahmp’ pool |

zVertex (real) (nVertices)

| | |
|------------------|---|
| Units | m |
| Description | <i>Cartesian z-coordinate of vertices</i> |
| Accessed in code | as ‘zVertex’ from the ‘mesh’ pool |

zwtxy (real) (nCells, Time)

| | |
|------------------|--|
| Units | m |
| Description | <i>water table depth</i> |
| Accessed in code | as ‘zwtxy’ from the ‘diag_physics_noahmp’ pool |

zxu (real) (nVertLevels, nEdges)

| | |
|------------------|--|
| Units | <i>unitless</i> |
| Description | <i>dz/dx on horizontal coordinate surfaces at u levels</i> |
| Accessed in code | as ‘zxu’ from the ‘mesh’ pool |

zz (real) (nVertLevels, nCells)

| | |
|------------------|---|
| Units | <i>unitless</i> |
| Description | <i>d(zeta)/dz, vertical metric term</i> |
| Accessed in code | as ‘zz’ from the ‘mesh’ pool |

Appendix E

MPAS Copyright

The MPAS-specific source code developed under this project has been copyrighted under a BSD license. MPAS is freely available and is open-source. Any external software components have their own copyright statement directly in their source code included with this release. The full copyright statement is:

Copyright (c) 2013-2019, Los Alamos National Security, LLC (LANS) (LA-CC-13-047) and the University Corporation for Atmospheric Research (UCAR).

All rights reserved.

LANS is the operator of the Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 with the U.S. Department of Energy. UCAR manages the National Center for Atmospheric Research under Cooperative Agreement ATM-0753581 with the National Science Foundation. The U.S. Government has rights to use, reproduce, and distribute this software. NO WARRANTY, EXPRESS OR IMPLIED IS OFFERED BY LANS, UCAR OR THE GOVERNMENT AND NONE OF THEM ASSUME ANY LIABILITY FOR THE USE OF THIS SOFTWARE. If software is modified to produce derivative works, such modified software should be clearly marked, so as not to confuse it with the version available from LANS and UCAR.

Additionally, redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- 3) None of the names of LANS, UCAR or the names of its contributors, if any, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix F

Revision History

2 June 2025

- Add new namelist options and fields for MPAS v8.3.0.
- Update physics options in Chapter 6

27 June 2024

- Add new namelist options and fields for MPAS v8.2.0.
- Update physics options in Chapter 6

18 April 2024

- Add new Section 8.3 for invariant stream capability.
- Add new namelist options and fields for MPAS v8.1.0.

6 July 2023

- Update units and description for several fields following the MPAS-A v8.0.1 release.
- Add missing diagnostic fields to Appendix D.

16 June 2023

- Updates for MPAS v8.0.0, including new namelist options.
- Updated physics scheme versions in Table 6.3

8 July 2019

- Removed references to the `double_to_float_grid` utility from Section 3.4.

8 June 2019

- Significant updates to reflect MPAS v7.0, principally, new namelist options, model fields, and a description of the limited-area simulation capability
- Moved description of SST and sea-ice updates to a new Chapter 8
- Updated physics scheme versions in Table 6.3

11 May 2019

- Update the default value for `config_o3climatology` to match the v6.3 release: the default is now ‘true’.

17 April 2018

- Minor updates for MPAS v6.0, including new `config_topo_data` option and new `initial_time` field.

22 March 2018

- Update Appendix D with correct units for `scalars_tend`.

1 August 2017

- Update documentation in Chapter 3 for building with PIO 2.x versions
- Correct other minor typographical errors and unclear wording

12 May 2017

- Add documentation for new `config_extrap_airtemp` initialization namelist option
- Remove documentation for `config_smdiv_p_forward` model option, which was removed in v5.1
- Add note that `config_len_disp` is also used by 3-d divergence damping in v5.1
- Change default value of `config_smdiv` model option to 0.1, reflecting changes in v5.1
- Other minor changes to match the MPAS-Atmosphere v5.1 release

23 December 2016

- Added description of the new ‘convection_permitting’ physics suite, as well as a list of all possible physics parameterizations, to Chapter 6.
- Updated the chapter on building MPAS to mention specific versions of PIO that are known to work with MPAS, information on the method used to build single-precision executables in v5.0, and a mention of the *experimental* OpenMP capability in v5.0.
- Appendices A, B, and D are now automatically generated based on XML Registry files.
- Many other small changes to reflect the state of the v5.0 release.
- Minor wording changes and corrections of typographical errors throughout document.

19 May 2015

- Added chapter describing the MPAS-Atmosphere physics suites introduced in MPAS v4.0.
- Added a section on building the model with single-precision reals.
- Updated the I/O chapter to include the new io_type option introduced in MPAS v4.0.
- Updated a few namelist options to match MPAS v4.0 code.
- Minor wording changes and corrections of typographical errors throughout document.

18 November 2014

- Added chapter describing the MPAS runtime I/O system available in MPAS v3.0.
- Updated the chapter on running MPAS-Atmosphere to match MPAS v3.0 code.
- Updated a few namelist options to match MPAS v3.0 code.
- Minor wording changes throughout document.

13 June 2013

Initial version.