U.S. Department of Commerce
National Oceanic and Atmospheric Administration
National Weather Service
National Centers for Environmental Prediction
5830 University Research Court
College Park, MD 20740-3818

Office Note 519
https://doi.org/10.25923/dvv2-3g03

June 2024

Coupling Infrastructure Capability in UFS Weather Model

Denise Worthen[1], Jun Wang[2], Raffaele Montuoro[2], Dom Heinzeller[3], Bin Li[1], Gerhard Theurich[4], Ufuk Turuncoglu[4], Dan Rosen[4], Dusan Jovic[1], Brian Curtis[1], Rahul Mahajan[2], Hang Lei[1], Alexander Richert[1], Arun Chawla[5], Jiande Wang[1],Jessica Meixner[2], Ali Abdolali[6], Matthew Masarik[1], Li Pan[1], Michael Barlage[2], Bin Liu[1], Mariana Vertenstein[7], Tony Craig[8], Rusty Benson[9], Thomas Robinson[9], Thomas Clune[10], Weiyuan Jiang[10], Neil Barton[2], George Vandenberghe[1], Mark Potts[8], Jong Kim[8], Natalie Perlin[8], Cameron Book[8], Ligia Bernardet[11], Fanglin Yang[2], Shan Sun[11], Hyun-Sook Kim[12], Barry Baker[11], Jianping Huang[1], Chan-Hoo Jeon[8], Ivanka Stajner[2]

[1] Lynker@NOAA/NWS/NCEP/EMC
[2] NOAA/NWS/NCEP/EMC
[3] UCAR/UCP/JCSDA
[4] NCAR/CGD/ESMF
[5] TOMORROW IO
[6] USACE/ERDC/CHL
[7] Norwegian Meteorological Institute
[8] NOAA/EPIC
[9] NOAA/OAR/GFDL
[10] NASA/GSFC
[11:] NOAA/OAR/GSL
[12:] NOAA/OAR/AOML
[13] NOAA/OAR/ARL

Table of Contents

Abstract

The Unified Forecast System (UFS) is an end-to-end forecast system for the next generation of the National Centers for Environmental Prediction (NCEP) production suite. The UFS weather-model (UWM) comprises the model component of the UFS. It consists of atmosphere, which currently includes the Finite Volume Cubed Sphere (FV3) dynamical core and the Common Community Physics Package, CCPP; ocean, sea ice, wave, land, aerosol, chemistry, and data model components with a central mediator component to couple the components together. The coupling strategy among the components uses the common Earth System Prediction Suite (ESPS) architecture, an Earth System Modeling Framework (ESMF) and National Unified Operational Prediction Capability (NUOPC) based coupling infrastructure framework. The NUOPC interfaces, also called caps, allow each model component to be an independent ESMF grid component. The coupling communication uses either generic NUOPC connectors or the NUOPC compliant Community Mediator for Earth Prediction Systems (CMEPS). This technical note provides details on the coupling capability available in the latest UWM. The model component NUOPC caps and component coupling strategies are documented. The coupled configurations, computational performance, and features that improve the computation performance are also illustrated.

1. Introduction

NCEP has been running the coupled model Climate Forecast System (CFS) version 2 (Saha, et al, 2014) in operations since 2011. CFS is a coupled system using Global Spectral Model (GSM) as the atmosphere component, the Modular Ocean Model, version 4 (MOM4) as the ocean component, the GFDL sea ice simulator (SIS) as the sea ice component, and the Noah land surface scheme as the land component. The GFDL Flexible Modeling System (FMS) is used as the infrastructure framework. The coupling capability is similar in CFS v1 and v2 (Saha, et al. 2010). A simple coupler developed at EMC allows the GSM to communicate with the dummy atmosphere model in the MOM4/SIS system and the dummy atmosphere then exchanges the coupling fields with MOM4/SIS through FMS. The data mapping of the coupled fields was performed through the exchange grid interpolation that is available in the FMS infrastructure framework. The Noah Land Surface Model (LSM) runs on the same grid as the atmosphere and is called within the atmosphere model as subroutines.

No major upgrades have been made in NCEP's coupled model since the CFS v2 implementation in 2011. Instead, NOAA NWS started the Next Generation Global Prediction System (NGGPS) project to upgrade the model atmosphere dynamical core. In 2016, NOAA announced GFDL's Finite-Volume Cubed-Sphere Dynamical Core (FV3) as its future operational atmospheric dynamical core, with the intention to use it as the foundation to build a community-based, coupled, comprehensive earth modeling system. NOAA will use this Unified Forecast System (UFS) to unify its operational modeling suite and the UFS will become the core operational source system. Since the announcement, significant efforts have been put into integrating the FV3 atmosphere dynamical core and developing the fully-coupled system based on the new dynamical core.

Meanwhile, the Earth System Modeling Framework (ESMF) group published an article summarizing the common earth system model architecture ESPS in 2016 (Theurich, et al, 2016). The architecture enables multiple agencies and programs to develop coupled models using sharable and interoperable model components using a common coupling strategy, ESPS. The ESMF group and EMC developers had been working on implementing the ESPS coupling approach in the NOAA Environmental Modeling System (NEMS) framework with the GSM coupled with Modular Ocean Model, version 5 (MOM5) and the Community Ice Code, version 5 (CICE5); this work was redirected to transitioning the GFDL FV3 dynamical core based coupled system.

NCEP's first operational implementation using FV3 dynamical core was the Global Forecast System version 15 (GFSv15), where the atmosphere model was built as an ESMF grid component and brought into the NEMS system (NOAA note, 2019). After that, the Global Ensemble Forecast System (GEFS) v12 (NOAA note, 2020) and the GFSv16 (NOAA note, 2021) were implemented with the atmosphere one way coupled with WW3 using NUOPC connectors. The aerosol component was embedded in the atmosphere model as FORTRAN subroutines and activated in GEFSv12. Other flagship model components such as MOM6 and CICE5 were integrated into the NEMS system gradually. Later CICE5 was upgraded to CICE6, and the mediator component was upgraded from the NEMS mediator to the CMEPS in 2020. A description and results of the global model coupled prototypes created during this development period was published recently (Stefanova, et. al, 2022). In this note we document the latest status of coupling capability in the UFS weather-model (UWM). It provides technical details on the coupling strategy among the model components used in the latest global prototypes and regional applications such as GFS, GEFS, the Hurricane Analysis and Forecast System (HAFS), the Rapid Refresh Forecast System (RRFS), marine data assimilation system (MARINE-DA) and regional air quality model (UFSAQM).

In Section 2, the implementation of the ESMF/NUOPC-based coupling approach in the UWM is described including the NUOPC caps for all model components, mediator and run sequence. In Section 3, an overview is provided on the coupled applications using these components described in section 2. The computational performance results of the global coupled application and its model components are presented in Section 4. The approach to build applications, library dependency and the test system for the UWM is explained in section 5, and future work is summarized in Section 6.

2. Model coupling description

The UWM is the model source system for NOAA's operational numerical weather prediction applications. The coupling capability has been developed to support UFS coupled applications in the past years (Figure 1). The model tag described here can be found at: https://github.com/ufs-community/ufs-weather-model/tree/UWM_coupling_technote.

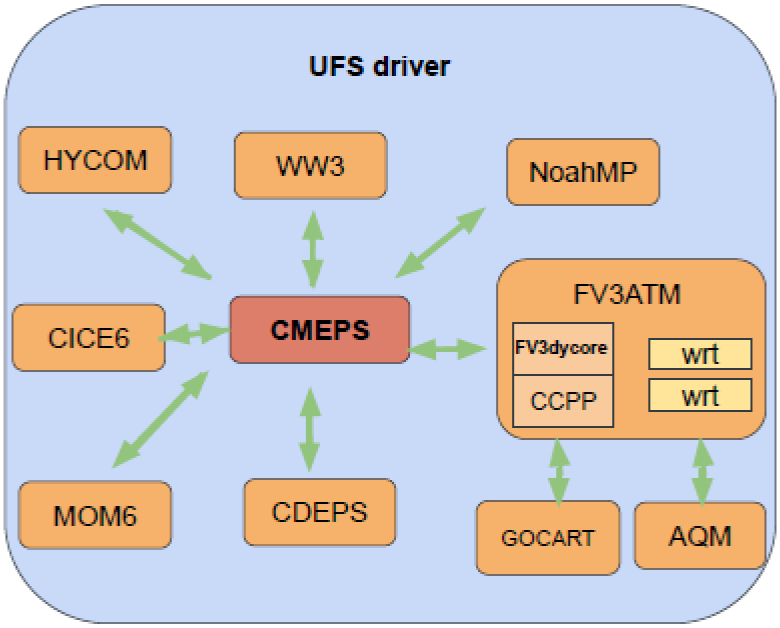


Figure 1 The UWM component structure.

In the system, FV3ATM is the atmosphere component with its repository owned and maintained by EMC and its collaborators. FV3ATM consists of the NUOPC compliant coupling interface (also called FV3 cap), dynamical core, CCPP, moving nest, stochastic physics driver, and IO along with Unified Post Processing (UPP). The dynamical core points to a UWM specific develop branch in a repository owned and maintained by GFDL. The FV3 dynamical core is scalable and flexible supporting both hydrostatic and non-hydrostatic atmospheric simulations. CCPP points to a UFS fork whose authoritative repository is owned by NCAR, it has a collection of atmospheric physical parameterizations and a framework that couples the physics to a host model's dynamical core. The package supports both research and operational development of physical parameterizations and experimentation with physics-dynamics coupling methods. In the UWM, the FV3 dynamical core and CCPP Physics are coupled through the CCPP Framework and the FV3ATM is coupled with ocean, sea ice and wave though CMEPS and coupled with aerosols and chemistry through ESMF connectors.

The Community Data Models for Earth Prediction System (CDEPS) and Community Mediator for Earth Prediction Systems (CMEPS) in UWM are forks of the authoritative ESCOMP CDEPS and CMEPS repositories, respectively. CDEPS is a set of NUOPC-compliant data components along with ESMF-based share code. The data models perform the basic functions of reading external data files, modifying those data, and then sending the data back to the CMEPS mediator. CMEPS is a NUOPC-compliant Mediator component used for coupling Earth system model components. It is developed through the collaboration between NCAR, NOAA/EMC,and NOAA/GFDL. CMEPS is currently used in UWM and NCAR's Community Earth System Model (CESM).

There are two options for the ocean model. The Modular Ocean Model, version 6 (MOM6) is a geophysical model with numerical representation of the global or regional seas. The MOM6 repository is owned and maintained by the Mom-Ocean Consortium with major contributors such as GFDL, NCAR, NASA, EMC and several university-based ocean modeling research groups. It contains driver interfaces for coupled, standalone and unit test runs. In UWM, MOM6 is coupled with the atmosphere, sea ice, and wave models through CMEPS. The other ocean model option is the HYbrid Coordinate Ocean Model (HYCOM), which is a fork from the HYCOM consortium repository. HYCOM is an ocean general circulation model with the hybrid vertical levels of isopycnic, z-level and terrain-following levels. In UWM, HYCOM is coupled with the atmosphere model through CMEPS.

The sea ice model in UWM is the Community Ice Code, version 6 (CICE6), which is a fork from the CICE-Consortium repository. It simulates the growth, melting, and movement of polar sea ice. CICE6 consists of a dynamical core and a column physics core contained within the Icepack submodule. In the UWM, sea-ice dynamics are on the Arakawa-B grid while thermodynamic calculations occur on the Arakawa A-grid by default. The ability to allow CICE sea ice dynamics on the Arakawa C-grid has recently been implemented and can be enabled via namelist control in UWM[1]. When enabled, this allows the coupling of sea ice velocities and ocean velocities directly on their native Arakawa C-grid locations. CICE is coupled to the atmosphere, ocean and wave model through the CMEPS mediator.

The wave model WAVEWATCH III (WW3) points to a UWM branch of the WW3 authoritative repository located under the EMC github community. WW3 is a community wave modeling framework that includes the latest scientific advancements in the field of wind-wave modeling and dynamics. In the UWM, WW3 is coupled to the atmosphere, ocean and sea ice through the CMEPS mediator. A legacy configuration which uses the older, grid-based NUOPC cap is also currently maintained.

The aerosol model points to the NASA GOES ESM 2nd-generation GOddard Chemistry Aerosol Radiation & Transport (GOCART) official repository. GOCART simulates major tropospheric

---

[1] Additional development within the MOM6 and CICE6 NUOPC caps is required to implement this feature fully in the coupled configurations.

aerosol components, including sulfate, dust, black carbon (BC), organic carbon (OC), and sea-salt aerosols. In UWM, GOCART is directly coupled with the atmosphere model through an ESMF connector.

The air quality component is using the CMAQ 5.2.1 tag in the US EPA CMAQ repository. CMAQ is a three-dimensional gridded atmospheric chemistry and transport model that simulates ozone, particulate matter (PM), toxic airborne pollutants, visibility, and acidic and nutrient pollutant species throughout the troposphere. In UWM, CMAQ is coupled with the atmosphere model through an ESMF connector.

The land component is currently pointing to a fork containing the CCPP version of the NoahMP LSM. NoahMP uses multiple physics options for key land-atmosphere interaction processes (Niu et al., 2011). It is developed and maintained through the collaboration of NCAR, NCEP, NASA, and university groups. In UWM, Noah-MP can be called directly with the atmosphere model FV3ATM or can be coupled with FV3ATM through CMEPS.

2.1 Coupled fields definition

In order to explain the coupling strategy in UWM, we first describe how the coupled fields are defined. The coupling fields are defined in a yaml file called fd_ufs.yaml. This file contains a community-based field dictionary for the shared coupling fields among the earth components models in a NUOPC coupling system. Entries in the field dictionary are organized as YAML lists of maps. The NUOPC Field Dictionary data structure in the model code is set up by the NUOPC function called NUOPC_FieldDictionarySetup, which loads the fd_ufs.yaml file. The field metadata described in each entry is used by the NUOPC layer to match fields provided and requested by the various component models. The field dictionary can be shared with other earth modeling systems that use the same ESPS coupling strategy, such as the Community Earth System Model (CESM).

The standard field metadata for each coupling field has following keys and corresponding values:

- Standard_name: <field_name>
  Canonical units: <unit>
  Description: <brief description about this field>
  Alias: <other_field_name>

The standard_name is an abbreviation name with the canonical units that serve to fully define a field. The description keyword gives brief information on this field and the alias keyword lists alternative names for the field. Either the standard_name or alias name can be used in a component model, and the NUOPC layer will recognize these fields as the same field using the definitions provided in the yaml file. Only the standard_name and canonical units are mandatory to define each coupling field while description and alias are optional keys.

Alias names can also be defined as:

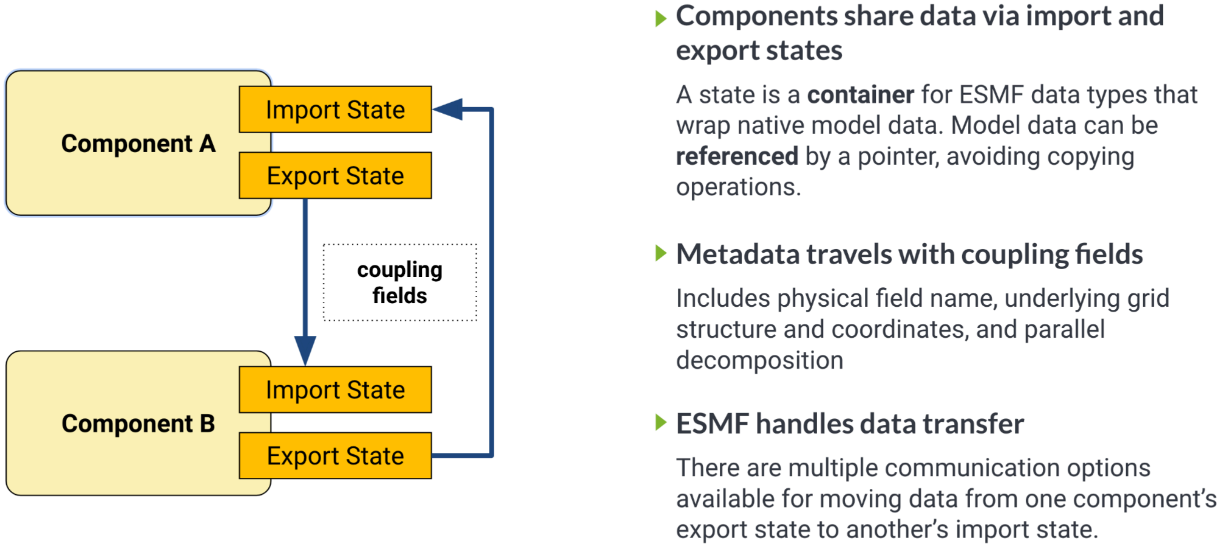
- Alias: <name> or [<name>, <name2>]
  Standard_name: <field_name>

An alias can be one character string or a list of strings. This allows a field to have different names used in the coupling field exchange.

It is required that all fields are fully defined with standard name and canonical units in the dictionary file before they can be used anywhere in the component models or mediator. While adding one alias keyword to a Field definition dictionary entry is allowed and will be parsed by the NUOPC Layer, it is recommended that all synonyms be included as separate entries. More information can be found in the NUOPC documentation (NUOPC Layer Reference. 2.2.1 Field Dictionary file). [CSC members, 2021]

2.2 NUOPC caps

As specified in Theurich et al (2016), a model NUOPC cap is a wrapper upon the component model to build the ESMF grid component within the common ESMF/NUOPC framework. Through these interfaces, the coupling fields can be communicated among the component models and used internally within the models. The coupling fields are implemented as ESMF Fields and are added to the ESMF model grid component import and export states. An ESMF field contains a reference to an ESMF Array as well as additional information (metadata) which describes the field name, geophysical location and decomposition of the constituent field. The following figure shows the caps of two components A and B, their import and export states and the transfer of the coupling fields.



- **Components share data via import and export states**
  A state is a **container** for ESMF data types that wrap native model data. Model data can be **referenced** by a pointer, avoiding copying operations.
- **Metadata travels with coupling fields**
  Includes physical field name, underlying grid structure and coordinates, and parallel decomposition
- **ESMF handles data transfer**
  There are multiple communication options available for moving data from one component's export state to another's import state.

(source: *ESMF Web Tutorial, 2020*)

Each NUOPC cap contains a SetServices subroutine. It registers the model as a general NUOPC model component and specifies three basic steps (execution methods) to run a model: an initialization step, a run step and a finalize step. Any of these can contain multiple phases. Initialization generally consists of reading or setting required parameters, advertising the fields required by or provided by the component and realizing the needed fields after a coupling field is confirmed as required. Realizing a field here means creating and initializing the fields on the component's grid or mesh. The run step advances the model by a single coupling time step; it includes obtaining the imported field values, advancing the model and then filling the exported fields with updated values. The model is cleanly shut down and exited through the finalize step where all the model states and grid components will be properly destroyed. All the model components have an ESMF clock associated with it to manage integration time. The clock is inherited from the top UWM driver component, updated at model run step and synchronized among model components if necessary.

These steps are implemented in all of the UWM component caps. In the sections below, specific details of component caps will be described while generic model instance implementation will be skipped.

2.2.1 FV3ATM

The FV3ATM cap (*fv3_cap.F90*) is the NUOPC driver to run the atmosphere model in UWM. It contains a single public entry point, SetServices, which registers all of the user-provided subroutines accessed by the NUOPC layer. Different from other model components, it contains two types of sub-grid components, a forecast grid component (*module_fcst_grid_comp.F90* and the atmosphere driver code *atmos_model.F90*) as well as a write grid component (*io/module_wrt_grid_comp.F90*). It runs these sub components simultaneously. The purpose of running the asynchronized write grid components is to handle the model outputs on separate sets of MPI tasks from the forecast tasks so that the forecast integration can continue without being blocked by the slow In/Out (I/O) process. This asynchronized I/O approach significantly improved the FV3ATM run speed, especially in UWM configurations where I/O requires a significant amount of run time. The details on write grid components are skipped here because the focus of this note is on the coupling capability. However the write grid components do have an impact on the model computational performance as seen in section 4. The FV3ATM grid component structure is shown in Figure 2.

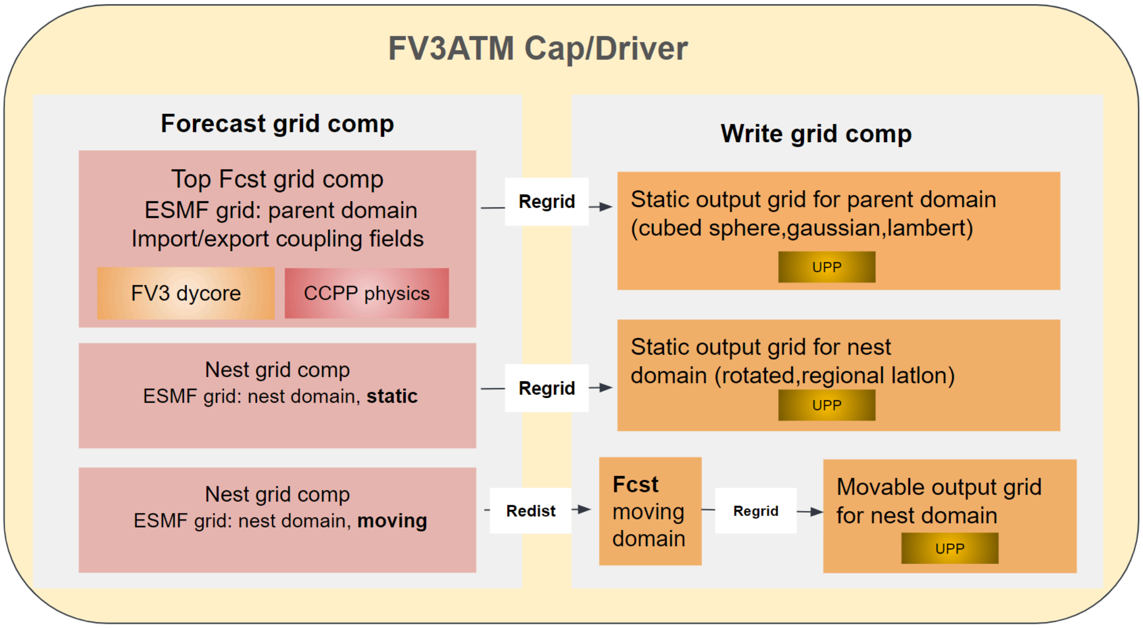


Figure 2. FV3ATM grid components structure

The FV3ATM has two initialization phases: *InitializeAdvertise* and *InitializeRealize*. The *InitializeAdvertise* phase reads in the configuration parameters (Table B.1), creates the forecast grid component and creates coupled fields in its import and export state to communicate with other earth model components. The forecast ESMF grid is also created on the native cubed sphere grid aligned with that from the FV3 dynamical core for coupling fields. The land sea mask is added to the grid for coupling purposes. The FV3ATM has an option to use the fractional grid (frac_grid=.true.) which allows water (ocean or lake) and land to co-exist in an atmosphere grid cell. To have consistent surface types with the ocean model in the coupled mode, the land sea mask is marked as ocean if there is non-zero ocean fraction in the atmosphere grid cell. At the NUOPC level when the handshaking of model components is done, the coupling fields required in the coupled system will be confirmed. The specific coupling fields in FV3ATM import and export states will be created on the ESMF forecast grid in the *InitializeRealize* phase. While the 2D coupled fields are data arrays copied from atmosphere internal fields, the aerosol and chemistry tracers instead use data pointers to save memory. It is worth noting that while the ESMF forecast grid on each MPI task mentioned above is defined from the native grid decomposition of the atmosphere dynamic with the 2 dimensional layout, the data structure inside physics is column based, and the physics state is saved in block and column arrays. The model state is mapped between the dynamic state data structure and the physics state data structure in the physics-dynamics coupling during the forecast integration. Details on dynamics and physics coupling are provided in Section 2.5.

Within the FV3ATM InitializeAdvertise phase, other processes such as forecast nest grid components and the asynchronous write grid components are also created but they do not participate in the NUOPC coupling communication at this time.

The FV3ATM cap contains three run phase subroutines: *modelAdvance*, *ModelAdvance_phase1* and *ModelAdvance_phase2*. The *modelAdvance* run phase allows the atmosphere to complete a full integration step which contains the two sub steps of *ModelAdvance_phase1* and *ModelAdvance_phase2*. This run phase is used by coupling configurations that do not require the atmosphere to pause during an integration step in order to complete the aerosol tendencies. The FV3ATM *ModelAdvance_phase1* calls the first run phase of the forecast grid component to run atmosphere dynamics, radiation and physics and to update some of the coupling fields in the export state. The atmosphere *ModelAdvance_phase2* calls the second run phase of the forecast grid component to update the model state, the data buffer for I/O and write restart files if requested. In some configurations such as atmosphere coupling with an aerosol or chemistry model, the *ModelAdvance_phase1* will first run, and then atmosphere component will pause and the compute tasks will run aerosol or chemistry model, and then go back to atmosphere *ModelAdvance_phase2* call to finish a full atmosphere step.

In the first run phase of the forecast grid component, time consistency is checked between the forecast component and the atmosphere import and export field states before coupling fields can be transferred to and from other components. The coupling fields coming from marine components (ocean, ice and wave) are only assigned on atmosphere grid points with positive ocean fraction. Some fields such as surface roughness, sea ice temperature and sea surface temperature (SST) have a restricted range of values to ensure model stability. If the import merge option *cpl_imp_mrg* is on, the coupled fields (e.g. SST) in the import state with missing values will be set to the internal surface temperature that is read in from initial conditions and updated with climatology. The atmosphere coupling fields can be found in Appendix C.1. For the coupled moving nest configurations, the top parent domain receives the coupling fields from other earth model components and passes them down to the nested domain grid components. The coupled fields in the atmosphere export state requested by other earth model components except the chemistry components are updated. If the atmosphere grid component is coupled with a chemistry component, the *atmos_model_exchange_phase_1* will be called to update the aerosols fields required by aerosol model GOCART or the air quality fields required by CMAQ in the forecast grid component export state. Those fields will then be transferred through the ESMF connector to the GOCART/CMAQ model component import state and used in the GOCART/CMAQ run phase.

In the forecast grid component's second run phase, if fv3atm is coupled with chemistry components, the atmosphere internal chemical tracers will be updated in the *atmos_model_exchange_phase_2* subroutine through ESMF connector after GOCART/CMAQ run phases are finished. Then the dynamics model state is updated with complete physics tendencies, a full atmosphere integration step is completed. Detailed information on the coupling fields imported from or exported to other components can be found in Appendix C.10.1 and C.10.2.

The FV3ATM's model clock is inherited from the UWM driver clock in the *ModelSetRunClock*. Unlike other earth models that call ESMF_ClockAdvance to advance the model, the FV3ATM cap calls TimestampExport_phase1 to update the current time from the driver clock after its run phases are finished. This is because FV3ATM has two run phases and some coupled fields are updated in phase1 and the model clock is not updated yet at that time, updating the model clock current time after the two run phases allows all the coupling fields in the export state to have consistent time stamp.

2.2.2 MOM6

The MOM6 NUOPC cap package consists of the cap code itself (*mom_cap*, *mom_cap_methods*), a module for managing ESMF Alarms (*mom_cap_time*) and a time management module for interaction with FMS (*time_utilities*). Two additional modules provide the access to the internal MOM-specific data type (*surface_forcing_CS*) for the ESMF import field data (*mom_surface_forcing_nuopc*) as well as a wrapper module (*mom_ocean_model_nuopc*) for the initialization, update and finalization of the ocean model state.

The MOM6 cap (*mom_cap)* contains a single public entry point, *SetServices*, which registers all of the user-provided subroutines accessed by the NUOPC layer. There are three initialization phases within the cap (*InitializeP0*, *IntializeAdvertise* and *InitializeRealize*).

The function of *InitializeP0* phase is to read in configuration parameters (Table B.2) from ufs.configure. In the *InitializeAdvertise* phase, an internal state derived type (*ocean_internalstate_type)* is defined containing pointers to three MOM6 types (*ocean_public_type*, *ocean_state_type*, and *ice_ocean_boundary_type*). Within the *ocean_internal_state* derived type, the imported coupling fields are populated in the *ice_ocean_boundary* while the exported fields are obtained from *ocean_public_type.* Additional initialization is made to access FMS functionality (*field_manager* and *diag_manager*). The model starting time is obtained from the ESMF clock and the model *runtype* is set. The *ocean_state* and *ocean_public* types are then initialized and the computation domain is defined by calling *mpp_get_compute_domain*. The computation domain is used to allocate the fields within the *ice_ocean_boundary*. Finally, the import and export fields are advertised.

In the *InitializeRealize* phase, the ocean grid and computational domain size are obtained from routine *get_ocean_grid* and an ESMF Distgrid is defined based on the global index of the computational domain. This DistGrid is transferred to the ESMF mesh that is created from the ESMF unstructured grid file[2]. A consistency check is made between the ESMF mesh and the

---

[2] The file name of the ESMF unstructured mesh is specified as an ocean configuration attribute.

MOM6 domain coordinators to make sure the differences are within the predefined tolerance value. Finally, the import and export fields are realized on the mesh[3] and the initial values of the MOM6 export fields are set in a *DataInitialize* phase by calling the *mom_export* routine.

Upon exiting the *InitializeRealize* phase, a call to *ModelSetRunClock* is initiated by NUOPC. This defines the model clock and creates ESMF Alarms for restart writing and stopping using the configuration parameters specified in ufs.configure (Table B.9.1).

The ocean model is advanced in the *ModelAdvance* phase. For an initial (startup) run, a lagged start-up is implemented, such that MOM does not advance during the first coupling time step and just returns. On the second coupling time step, the model is advanced two coupling time steps and subsequent coupling time steps proceed as usual. This startup procedure allows the ocean model, which is in a slow coupling loop, to be advanced once the fields it requires from other components in the fast coupling loop are available. A call to *mom_import* then fills the fields in the *ice_ocean_boundary* from the import state.

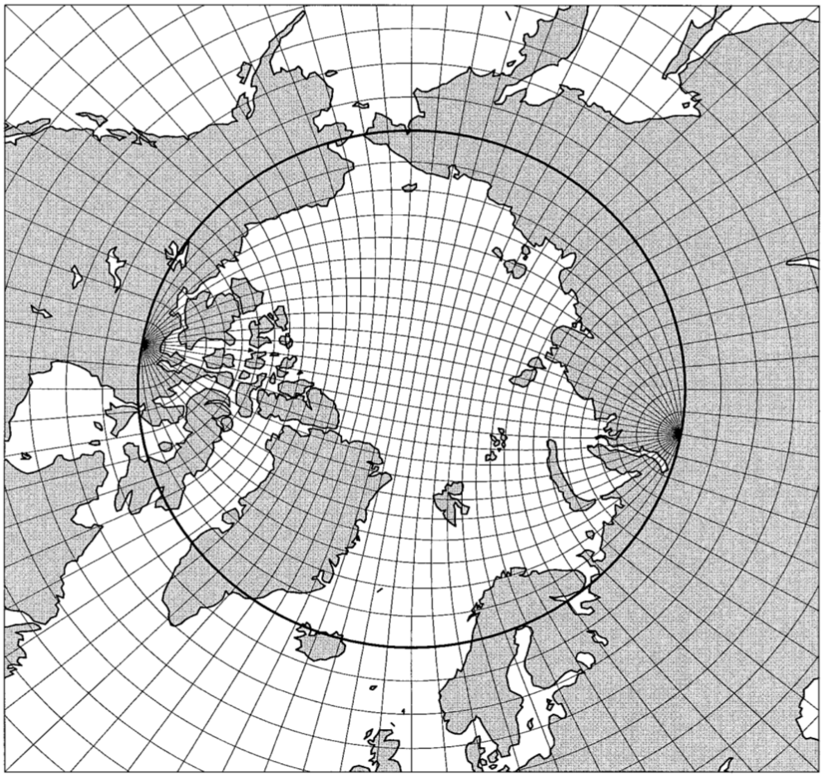


Figure 3. The tripole grid, northern hemisphere. Reproduced from Murray (1996)

In UWM global coupled applications, both the ocean model MOM6 and the sea ice model CICE6 always utilize the same displaced-pole domain configuration known as the tripole grid (Murray, 1996). This means that in the north polar region (north of ~ 60N), the model index

---

[3] The realization of the import and export fields on a grid rather than mesh is not used in UWM.

space does not align with lines of latitude and longitude (Figure 3). This requires that vector or other directional quantities imported from other components are rotated from eastward and northward directions to the model's defined directions along increasing i and j indices in the *model_import* call. The ocean model state is then updated one time step of model integration (*update_model_model*).

Fields in the export state are filled in a call to *mom_export*. Within *mom_export*, the sea surface slope in the I-J direction is calculated and rotated to E-N, vector quantities are rotated from model I-J to E-N and the freeze-melt potential field is provided as either the amount of frazil formation, or the negative of the melt-potential calculated by MOM6. See C.2 MOM6, Table C.10.5 States to OCN and Table C.10.6 Fluxes to OCN for details.

In *ocean_model_finalize* phase, ending restart files are written and MOM is finalized through a call to internal MOM6 routines *ocean_model_end*, *io_infra_end* and *MOM_infra_end*.

2.2.3 CICE

The CICE NUOPC cap package consists of the cap code itself (*ice_comp_nuopc*, *ice_import_export*, *ice_mesh_mod*), a utility routine (*ice_shr_methods*), and three wrapper routines (*CICE_InitMod*, *CICE_RunMod* and *CICE_FinalMod*) for the CICE model itself. Additional files (*ice_prescribed_mod*, *ice_scam*) are used for either prescribed ice or single column mode, respectively. Neither mode is utilized within the UWM.

The CICE cap (*ice_comp_nuopc)* contains a single public entry point, SetServices, which registers all of the user-provided subroutines accessed by the NUOPC layer. It includes two initialization phases (*InitializeAdvertise* and *InitializeRealize*). In the *InitializeAdvertise* phase, configuration parameters (Table B.3) are read from ufs.configure, the CICE input namelist is read, the basic state, grid and a subset of the necessary parameters for running the CICE model are initialized (*cice_init1*) and the requested fields in the import and export state are advertised (*ice_advertise_fields)*. The global index space for the CICE domain is found (*ice_mesh_set_distgrid*) and is used to define an ESMF Distgrid. This DistGrid is transferred to the ESMF mesh[4] and the CICE grid initialization is completed (*init_grid2*).

In the *InitializeRealize* phase, the remaining parameters and variables for CICE and Icepack are initialized and the import and export fields are realized on the ESMF mesh. Upon exiting the *InitializeRealize* phase, a call to *ModelSetRunClock* is initiated by NUOPC. This defines the model clock and creates ESMF Alarms for restart writing and stopping using configuration parameters (Table B.9.1).

---

[4] The file name of the ESMF unstructured mesh is specified as an ice configuration attribute. It is always the same ESMF mesh provided to the ocean component.

At each *ModelAdvance*, the clock is checked to determine whether the restart alarm is ringing and if so, the internal CICE variable *write_restart* is set on[5]. Import field values located on the center of each CICE grid cell are retrieved into a packed temporary array and the halos of the packed array are exchanged before being passed into the appropriate CICE array. Vector fields are imported on the center of grid cells[6] and oriented in the eastward and northward directions[7]. A vector rotation is applied to re-orient them to the model I-J direction. Internally in CICE, the atmospheric wind fields are used on the center grid points for the thermodynamic flux calculations before being regridded to the appropriate dynamics grid location.

The CICE model is advanced via a wrapper routine, *CICE_Run* which advances the model by calling the main driver routine of CICE, *ice_step*. After all calculations are complete, fields are divided by the ice area (*scale_fluxes)*[8] and history fields are accumulated. History and restart files are written if required. At the completion of the *ice_step* routine, control is returned to the NUOPC cap where export state fields are then filled (*ice_export*) with the air-ice and ocean-ice stress vectors at cell centers rotated to E-N. Export field values are filled only at ocean grid points where ice concentrations are in the range [0.0,1.0]. See C.3 CICE, Table C.10.7 States to ICE and Table C.10.8 Fluxes to ICE for details.

The model advance continues until the driver *stop_time* is reached, whereupon NUOPC initiates the component finalization phase.

2.2.4 WW3

The WW3 NUOPC cap package consists of the cap code itself (*wav_comp_nuopc*, *wav_import_export*), a utility routine (*wav_shr_mod*), and a routine to read the WW3 configuration namelist (*wav_shel_inp*). Also included is a routine to define netCDF variables and metadata when netCDF gridded output of mean wave parameters is requested (*wav_grdout*).

The WW3 cap (*wav_comp_nuopc)* contains a single public entry point, SetServices, which registers all of the user-provided subroutines accessed by the NUOPC layer. It includes two initialization phases (*InitializeAdvertise* and *InitializeRealize*). In the *InitializeAdvertise* phase, configuration parameters (Table B.4) are read from ufs.configure and the requested fields in the import and export state are advertised (*advertise_fields)*. In the *InitializeRealize* phase, a UWM specific initialization sequence is called (*waveinit_ufs*). This sets several model component level

---

[5] Restart writing triggered by the ESMF alarm, set by configuration variables *restart_n*, *restart_option*, does not override restart frequencies set by the CICE namelist. To avoid multiple restart frequencies, the appropriate settings in the namelist are set to a value greater than the total forecast length.

[6] Future development will allow C-grid ocean velocities to be transferred from MOM6 on native C-grid locations.

[7] Vector quantities are rotated to E-W and N-S prior to export from the MOM6 NUOPC cap.

[8] The scaling is required because the CMEPS mediator will multiply by the ice area when preparing fields for export to other components.

configuration parameters (Table B.4), reads the wave model input namelist (*read_shel_config*) and initializes the wave model (*w3init*). Using the initialized wave domain, a global index array is created and used to define an ESMF DistGrid. This DistGrid is transferred to the ESMF Mesh[9] and the import and export fields are realized on the ESMF Mesh (*realize_fields*).

Upon exiting the *InitializeRealize* phase, calls to *DataInitialize* and *ModelSetRunClock* are initiated by NUOPC. This sets the initial values for all valid import or export fields and defines the model clock. If the required configuration variables are present (Table B.4), ESMF alarms are created for WW3 restart and history alarm settings.

At each ModelAdvance step, the values of the import State variables are used to fill the native WW3 model arrays, the wave model is advanced, and the export State field values are filled. Because WW3 expects the imported field variables to be global, an all reduce communication (*ESMF_VMAllReduce*) is initiated to effectively scatter the import field values to the global WW3 arrays. When the configuration variable *merge_import* is set true, stored field data that can not be provided by the import state is read from a file and merged with the import state field values. These values are combined using a mask which defines the region of the WW3 domain where values are not obtained from the import State. Export fields are filled by utilizing native WW3 routines to calculate the required values at the coupling timestep. See C.4 WW3 and Table C.10.9 States to WAV for details.

2.2.5 GOCART

The GOCART UWM NUOPC-compliant aerosol cap package consists of a cap code (*Aerosol_Cap.F90*), aerosol component module to update model state from import state and update coupled fields in export state (*Aerosol_Comp_Mod.F90*), a module to update diagnostic field (*AerosolDiagUpdate.F90*) and modules to get coupling field values and statistics of the fields (*Aerosol_Shared_Mod.F90*, and *Aerosol_Tracer_Mod.F90*). Different from other model components, GOCART is using the NASA GEOS' MAPL infrastructure layer, which is built upon ESMF and is the foundation of NASA earth model system GEOS-5.

The aerosol cap contains a public entry point, SetServices, which registers all of the user-provided subroutines accessed by the NUOPC layer. It has three initialization phases: *ModelInitializeP0, ModelInitializeP1* and *ModelDataInitialize*. The *ModelInitializeP0* phase gets component information and prepares the initialization phase for the Advertise and Realize steps. The *ModelInitializeP1* phase advertises the import and export fields. The *ModelDataInitialize* phase gets the import fields, sets up MAPL cap, sets up the aerosol grid component through MAPL, initializes the aerosol component and creates maps linking imported aerosol tracers to MAPL fields. The aerosol coupling fields are listed in Appendix C.5. During this phase, two other grid components, ExtData and History components, are also created in the MAPL cap to process external data and history output fields and files. Under MAPL cap, the aerosol grid

---

[9] The ESMF Mesh is specified as a configuration attribute. In general, no requirement exists that the WW3 mesh be the same as that used for the ocean and ice models.

component also creates the GOCART2G as its child grid component, and the GOCART2G then creates the aerosol species grid components including dust, sea salt, black carbon, organic carbon, sulfur, and Nitrate aerosols. The active species components can be specified in the GOCART2G configuration file (Table 2.7.1). Each aerosol species component has their own cap, import and export states specified by MAPL interface, and it has a configuration file to specify number of bins, attributes, emissions, settling and optical properties (Table 2.7.1). The initialization of these components reads in the variables in their own configuration file (Appendix B.5). The UFS aerosol grid component architecture is shown in the Figure 4 below.

```
UFS-Aerosols (NUOPC)
 |
MAPL (Cap)
 |--AERO------GOCART2G
 |--ExtData       |--DU
 |--History       |--SS
                  |--CA.bc
                  |--CA.oc
                  |--SU
                  |--NI
```

Figure 4: AEROSOL architecture

The *ModelAdvance* step updates the aerosol internal state with coupling fields in the import state of the UWM aerosol chemistry grid component, runs the aerosol component one step by calling the MAPL hierarchical subcomponents and updates the export state of the chemistry grid component. Some diagnostic fields including the coupling fields can be printed out if needed. In the finalization step, the MAPL aerosol component and associated I/O framework are finalized.

It is worth noting that the NUOPC-based aerosol component can run with multiple coupled configurations in the UWM. The GOCARTmodel is shared in NUOPC-compliant applications across U.S. modeling and operational centers. Currently in UWM, the aerosol component is configured to run sequentially on the atmosphere grid and between the atmosphere splitted run phases. It can be configured at runtime to run all or part of the included aerosol processes allowing developers to further refine the model configuration and to control the computational cost, if necessary.

2.2.6 CMAQ

The CMAQ NUOPC cap package consists of an air quality model (AQM) NUOPC cap (*aqm_cap.F90*) and air quality model component (*aqm_comp_mod.F90*). The AQM cap has a single public entry point, SetServices, which registers all of the user-provided subroutines accessed by the NUOPC layer.

The AQM cap has three initialization phases: *InitializeP0, InitializeP1* and *DataInitialize.* The *InitializeP0* phase loads the configuration file and sets the component ready for the next initialization phase. In the initialization phase *InitializationP1*, the import and export coupling fields (Appendix C.6) are advertised through *NUOPC_Advertise* calls. The handshaking to connect FV3ATM is then conducted at the UWM NUOPC level to confirm the coupling fields that are required for FV3ATM and CMAQ coupling. Then in the *DataInitialize* phase, the model grid is defined using an import coupling field since CMAQ runs on the same grid with the same decomposition data structure as the atmosphere using atmosphere forecast tasks. The component internal state is allocated, then CMAQ model initialization is called to read in the configuration file (Appendix B.6) and the import/export fields are connected with the internal state variables. Sharing the same data structure between the FV3atm and CMAQ enables the two components to access the same memory space and share the coupling data by reference. This minimizes the memory footprint and improves runtime performance by avoiding the physical transfer of data.

In the run phase *ModelAdvance*, the chemical and aerosol tracers along with the meteorological and land surface inputs are imported to CMAQ through the NUOPC connector. The CMAQ driver then calls the CMAQ run phase as a one-dimensional column model. A unit conversion implemented in CMAQ is used to get original CMAQ tracer units from the imported tracer fields because the CMAQ aerosols and some tracers are expressed as concentration, while FV3ATM treats them as mixing ratio. The CMAQ model is then called to compute chemical, physical, and dynamical processes affecting air quality and atmospheric chemistry. The unit conversion is called again after to convert required tracers to mixing ratio units before exporting back to FV3ATM at the end of run phase. The chemical and aerosol fields are transferred back to FV3ATM through the NUOPC connector and all the model state fields are updated in FV3ATM run phase 2 *ModelAdvance_phase2* to finish one full integration step. The coupling between FV3ATM and CMAQ occurs two-way (Appendix B.6). Meteorological and surface fields are sent from FV3ATM to CMAQ, while the mixing ratios of 183 actively transported CMAQ species are updated by CMAQ and passed back to FV3ATM. FV3ATM then updates and returns such ratios after performing horizontal advection, PBL diffusion, and convective transport and wet deposition via Scale-Aware Simplified Arakawa Schubert mass flux (SASAS) deep and shallow convective schemes.

It is worth to note that integration of CMAQ in UWM required developing a specialized plug-in I/O interface connecting CMAQ I/O with FV3ATM and external data sources. This internal I/O layer was created in CMAQ to seamlessly replace the EDSS/Model-3 I/O APIs[10] used in CMAQ and to virtualize file I/O, which can now be performed via CMAQ's coupling infrastructure for meteorological fields and tracers and multi-tile disk I/O capability for input emissions.

---

[10] https://www.cmascenter.org/ioapi/documentation/3.1/html/index.html

2.2.7 HYCOM

The HYbrid Coordinate Ocean Model (HYCOM) provides active two-way coupling between the atmosphere and ocean model components. The forecast model interfaces have been wrapped in a NUOPC cap, as defined in section 2.2 NUOPC caps. The HYCOM cap is enabled with the ESPC_COUPLE preprocessor flag when building HYCOM. The HYCOM cap configures the model for coupling through NUOPC and defines the import and export fields as well as the grids used for each field. The cap is split into three initialization phases, a model advance phase, and a finalize phase.

During the first initialization step, phase 0, the HYCOM cap reads configuration settings through an ESMF_config class object that is attached to the component, or through the component attributes. The HYCOM component attributes are listed in the ESMF/NUOPC configuration file ufs.configure file (Table B.7).

In the initialization phase 1, the HYCOM cap reads the configuration file containing the list of HYCOM import and export fields then advertises these fields to connected components. The import and export field configuration lists are read in using ESMF_config class tables: ocn_import_fields and ocn_export_fields, respectively. The table columns define HYCOM field name, standard field name, and field units. The ocn_import_fields and ocn_export_fields lists provide users a mechanism for selecting coupled fields and defining standard names at run time. Table C.7 provides lists of all fields available for coupling. See also Table C.10.5 States to OCN and Table C.10.6 Fluxes to OCN.

In the initialization phase 2, the HYCOM cap initializes the HYCOM model, providing a start and end date-time to the model. Once initialized, HYCOM provides the grid decomposition information to the cap so that cap can create an ESMF DistGrid. Then the cap creates an ESMF Grid and fills in cell center coordinates and cell corner coordinates as spherical degrees. The cap also fills in a land sea mask, with value 0 masking out unused land cells. Once the grid is set up the cap realizes connected fields and removes non-connected fields. If import_setting is set to REQUIRED and an import field is not connected then the HYCOM cap will throw a fatal error, which is written to the ESMF PET logs.

In UWM, HYCOM is used only for the regional configuration of the HAFS application. In this case, the HYCOM domain is larger than the coverage by the atmospheric model, including the non-stationary atmosphere model (e.g. the HAFS parent domain). This requires that off-line sets of atmospheric forcing should be ready from the same source of GFS used for the atmosphere component before numerical integration. The sole purpose of this step is to one-way force a part of the ocean domain whereby HYCOM receives data from CMEPS on the grid provided by HYCOM and then replaces missing data cells with the off-line atmospheric forcing data, which is referred to as data merging.

The model advance step registers a subroutine that is to be called when the model integration step executes. The cap calculates and logs the end time of the advance call, copies imported field data into model data structures, calculates radiation flux and non-directional wind speed if needed, then calls HYCOM_Run, which integrates the model. If field merge is turned on then imported data and GFS data will be merged to fill in unmapped grid cells. After the internal model has completed model data is copied to export fields, thus completing the model advance step.

The cap provides a finalization step, OCEAN_Final. When this subroutine is called, memory allocated for storing coupling field information is freed.

2.2.8 Noah-MP

The Noah-MP model consists of a set of FORTRAN subroutines that do not provide any capability to perform file I/O (i.e. reading/writing input/output files), manage model configuration or advance the model itself to perform simulations. To enable Noah-MP to run as an external land component in UWM, an ESMF/NUOPC cap has been developed. It interacts with the Noah-MP CCPP/Physics driver (*noahmpdrv*) to integrate the model while also providing model I/O and configuration management.

The Noah-MP land grid component has two initialization phases: advertise and realize. In the advertise phase, the model specifies the internal metadata structures to set up import and export fields and states (Appendix C.8), and advertises fields in ESMF/NUOPC layer. In the realize phase, the model reads in configuration variables from top level ESMF/NUOPC configuration file (*ufs.configure*) (Appendix B.8), constructs a multi-tile ESMF grid object by reading the mosaic grid file, sets up the grid mask from the land-sea fraction, initializes the Noah-MP model (*drv_init*), and realizes fields by using model data structures used internally. The initialization includes allocating internal data structures, reading static information such as soil and vegetation types, deep soil temperature and vegetation greenness. In addition to reading static information, the model also reads initial conditions and calls the CCPP/Physics driver routine (*noahmpdrv_init*) to set up soil and vegetation properties. As a part of the realization phase, the model converts the ESMF multi-tile grid (two-dimensional in space) to its ESMF mesh representation (one-dimensional in space), which is used to perform model integration. Since, ESMF is responsible for transferring data back and forth from the multi-tile grid representation to the mesh representation, the component model can be extended to run with different grid/mesh structures other than cubed spheres.

The Noah-MP grid component is designed to have a single run phase subroutine: *ModelAdvance*. During the run phase, the cap updates the import state and calls Noah-MP driver routine (*drv_run*) to run the model through the calling of CCPP/Physics driver run subroutine (*noahmpdrv_run*) and updates the export state with the information calculated by the model. The run phase can be specialized given configuration options to support different applications such as one-way or two-way coupling with active atmosphere or forcing Noah-MP with a data atmosphere that is provided by the CDEPS component. The run phase also transfers forcing provided by external atmospheric component via its import state to the internal Noah-MP model data structures, temporally interpolates vegetation fraction and mean surface diffuse shortwave albedo to the model integration time, calculates solar zenith angle, performs unit conversions, calculates initial values of some model variables by calling stability function and writes initial condition and output to the files.

The Noah-MP land grid component run time is set through the *ModelSetRunClock*. The driver configuration decides the model start and stop times (through use of *model_configure* namelist file) and coupling interval to call the external land component through the use of ESMF/NUOPC run sequence (*ufs.configure*). The *ModelSetRunClock* call also sets up ESMF alarms for restart and stop times that are used internally in the model.

The current version of the Noah-MP component model supports two different modes: (1) standalone mode, which is forced by the data atmosphere and, (2) coupled mode which is used to interact with other prognostic model components such as FV3ATM. The standalone mode was initially tested with the data provided by Global Soil Wetness Project (GSWP; Drimeyer et al., 1999) and ECMWF's ERA5 but it can also be run with the other datasets supported by a CDEPS atmosphere. Coupling through CMEPS supports both one-way (no feedback to atmosphere) and two-way (provides land surface fluxes to atmosphere) interactions with the FV3ATM atmosphere model. Two-way coupling was validated by comparing atmosphere-land fluxes calculated under FV3ATM CCPP/Physics with the component version of the Noah-MP model using one-way coupling configuration.

The current version of the Noah-MP component model uses a cubed-sphere grid as the FV3ATM atmospheric model component and is tested only with global application. However, the NUOPC cap is designed to use an ESMF-provided generic unstructured mesh representation internally to support different varieties of regional and global grids. Support for regional applications such as the HAFS and RRFS will be available in future releases. This will require minor modification in the NUOPC cap, especially for I/O calls and the domain creation part of the coupling interface of the land component. The Noah-MP model used as a component land model will be updated to the latest version 5 in the future.

The coupling fields for import and export states are defined based on the used configuration. Currently the Noah-MP component model can be used with either data atmosphere or active atmosphere component and the model internally adapts those configurations by checking namelist options and also import fields. The side-by-side configuration was designed to perform initial validation of Noah-MP cap by comparing the results calculated on Noah-MP run inside FV3ATM/CCPP with the component version. See also Table C.10.3 States To LND and Table C.10.4 Fluxes To LND.

2.3 CDEPS

The Community Data Models for Earth Predictive Systems (CDEPS) contains a set of NUOPC compliant data components along with ESMF based “stream” code that enables new capabilities in selectively removing feedback in coupled model systems. In this design, a “stream” is defined as a set of data files containing a set of fields, where all the fields are on the same stream mesh and have the same time coordinates. The CDEPS data models perform the basic function of reading external data files, modifying those data, and then sending the data back to the CMEPS mediator or any other active model component directly with NUOPC connectors. It can also perform regridding between the stream resolution and the data model resolution at run time for any regridding option that ESMF supports. This is a key feature for the data model configurations that have multiple streams on different grids/meshes and allow interaction with the external component (active model component or mediator) by interfacing with a common grid/mesh. The I/O infrastructure uses the PIO library so that all the stream data is read in parallel by the CDEPS stream code. In this design, each data component (*datm*, *docn*, etc.) has its own ESMF/NUOPC cap but they share the common infrastructure and routines to support different data modes.

The CDEPS data component has two initialization phases for each data model: advertise and, realize phases. In the advertise phase, the data component queries namelist files and specifies a data model mesh and mask files along with other stream-independent data model specific configuration variables. Then, the advertise phase initializes PIO for reading and writing netCDF files under CDEPS. After that, the top level advertise phase calls the stream specific routines, since advertised fields are changed based on the data mode used. This allows CDEPS to be specialized with the selected data mode and the list of exported fields (Appendix C.10). In the realize phase, the data model reads the stream definition file and runs the data component to prepare initial data for other components.

The CDEPS data component is designed to have a different run phase for each data mode, which is controlled by a top-level data component specific NUOPC cap. In the first advance step, the data model specific run phase initializes the export fields that have a corresponding stream field, then initializes the data mode specific stream and export field pointers. If it is required, the data model also reads the restart files in this initial step. Spatial and temporal interpolation is performed internally using ESMF provided spatial interpolation types and custom temporal interpolation routines if the data model and stream meshes are not identical. After interpolating (or transferring) the stream to the data model mesh, the top-level advance routine calls data mode specific routines, which are responsible for calculating value-added fields (i.e., wind speed from wind components) and converting units of the data stream based on the convention used in CDEPS.

In the finalization phase, the data model just returns a message that indicates the end of the main integration loop.

The CDEPS data component run time is set through the shared *dshr_set_runclock* routine. In this case, the driver configuration dictates the model start and stop times (through use of *ufs.configure* namelist file) and coupling interval to call the data component through the use of ESMF/NUOPC run sequence (*ufs.configure*). The *dshr_set_runclock* call also sets up ESMF alarms for restart and stop times that are used internally in the model and create internal clock representations.

As mentioned previously, CDEPS includes two programming layers to support flexible data components: (1) data models and (2) streams. The data model stays on top and interacts with the other active model components or the CMEPS mediator. Unlike the data models, streams do not directly interact with other components but are used by the data models to create export states. In this case, the streams could have different meshes but they are spatially mapped to data model mesh before passing to the other components. This step also includes temporal interpolation to calculate the data for a specific time, and has ability to perform different temporal interpolation types for each variable. For solar radiation, the data can be interpolated using solar zenith angle weighting in order to represent the diurnal cycle. The decomposition of the data is handled by ESMF.

2.4 CMEPS

In the UWM, CMEPS is the coupler used to exchange fields between the atmosphere (FV3atm or a data atm), ocean (MOM6 or HYCOM), sea ice (CICE), wave (WW3) and land (NoahMP) components.

For each component connected through the mediator, a field connection is made between the provider and acceptor component. This connection is made through a process of matching of standard field names between components using a Field Dictionary (section 2.1). Fields which are connected between components are placed within corresponding import and export states of the mediator. The mediator is responsible for transferring field information contained in these ESMF states between components. This process can require spatial interpolation[11] (mapping from one grid or mesh to another), merging (combining fields from different components) and time-averaging (synchronizing fields between slow and fast coupling components).

The CMEPS mediator is composed of a set of modules which manage generic (i.e. non-component specific) functionality, I/O (for both restart and mediator history files), time functions and additional diagnostic metrics. A second set of modules are component specific, in the general form of *med_phases_prep_[component]* and *med_phases_post_[component]*. The prep and post phases contain the mapping of fields from component to component. Whether mapping occurs each time a component completes their run phase ("post") or prior to executing

---

[11] Fields between components on the same grid, such as ocean and sea ice, are simply redistributed

a run phase ("prep") is a design decision based on the relative frequency of the mapping required and whether an updated coupling field is needed in the mapping (in this case, the mapping is always done in the prep phase). Finally, to accommodate multiple coupled systems, a set of application specific modules (*esmFldsExchange_[cesm,ufs,hafs]_mod*) are present for defining the field exchanges required for CESM and UWM.

The CMEPS mediator is initialized within the *med.F90* module, adding all the entry points for the mediator. There are five initialization phases. During the first phase (*InitializeIPDv03p1*), the mediator's internal state is created and memory for history files is allocated (*med_phases_history_init*) for all possible components. Configuration parameters which determine characteristics of the atmosphere-ocean flux scheme and the application specific coupling mode are also set.

There are multiple coupling modes supported for UWM within CMEPS, but in this document the discussion is restricted to those intended for UWM global coupled application, global NG-GODAS application and the regional hurricane application HAFS. These coupling modes are *ufs.frac*, *ufs.nfrac.aoflux* and *hafs*, respectively. The corresponding definitions for the import and export fields for the components are defined in either *esmFldsExchange_ufs_mod* or *esmFldsExchange_hafs_mod*.

When the applicable *esmFieldExchange* is called as part of the advertise phase of the mediator, fields are added to a list of fields to and from pairs of components. The default source and destination masking values used in the creation of the route handles are then set for each coupling mode and the names and number of any scalar fields are read from the configuration file. In UWM (Appendix B.9), the only scalar fields used are associated with the two-dimensional spatial dimensions corresponding to any component mesh. These values are needed when writing the mediator history and restart files. The mediator list of fields to and from component pairs are then advertised.

CMEPS realizes the exchanged fields through a multi-phase process to transfer and redistribute the component's grid or mesh onto the mediator's mesh. The required import and export component field bundles and, if applicable, field bundles for the ocean albedo calculation and the mediator flux computations are then created.

A second call is made into the application specific *FldsExchange* module to define the required mapping type and merging for all field pairs which exist in the import side field bundle and the export side field bundle. Mapping is defined for each field pair by specifying the source and destination component including the mapping method (e.g. conservative, bilinear), mapping normalization and optional mapping file[12]. Similarly, merges are defined by specifying the component pairs, the merge type and the optional merge weight. Multiple merge types are

---

[12] Mapping files are currently not used in UFS.

supported, including copy, copy-with-weights and merge (i.e, sum). Merges that can be defined using fractions carried in the fractional field bundle utilize the auto-merge functionality.

Once all field exchanges have been realized, the required packed field bundles and any required accumulation field bundles are then created. The field bundles required for the time-varying fractions are initialized and set in *med_fraction_mod* before the final *DataInitialize* cycle is entered, ensuring that all components are at the same time and all data dependencies have been met. The final step within the mediator *DataInitialize* phase is to initialize the mediator's I/O and diagnostic functions, read any mediator restart file and call each component's post-run phase to initialize all values. At this point, the mediator is fully initialized and ready for the Run phases.

Within the *esmFieldExchange* module, the fields and their mapping and merging definitions depend on the target application within the UWM. The global application can currently be coupled with either an active ATM or a data ATM (DATM) component. When coupled with a DATM for the NG-GODAS application, the coupling mode is *ufs.nfrac.aoflux* and the atmosphere-ocean fluxes are calculated in the mediator itself on the ocean grid using the parameterizations defined in *flux_atmocn_mod.* ATM fields are mapped to the OCN using first-order conservative mapping and filling of unmapped destination points with the nearest source point. This mapping type is denoted as *mapconsf_nstod* within CMEPS. Since the source fields are global and unmasked they are normalized using unity normalization[13].

For the global application with an active ATM, the coupling mode is *ufs.frac* and the atmosphere-ocean fluxes are computed in the ATM. These fluxes are mapped to OCN using first-order conservative mapping. This mapping type is denoted as *mapconsf* within CMEPS. The mapping normalization fraction to OCN for fluxes calculated by the ATM is the ocean fraction on the ATM grid (*aofrac*). The mapping normalization fraction to the ATM from OCN or ICE is either the time-varying sea ice fraction (*ifrac*) or its complement *ofrac*. The mapping from WAV to the ATM is normalized using unity normalization since the WAV field is global and unmasked[14].

In the global applications, OCN and ICE are on the same grid. In this case, fields are simply redistributed from source to destination. This mapping type is denoted in CMEPS as *mapcopy*. No mapping normalization is applied.

Since the HAFS application utilizes non-overlapping domains between component models (i.e., wave, ocean and atmosphere) a third mapping type is used for that coupling mode. This mapping type is denoted within CMEPS as *mapfillv_bilnr* and results in bilinear mapping onto the destination domain where it overlaps with the source domain. Non-overlapping regions of the domain are set with a fill value, which indicates to the destination component model where it

---

[13] A static unity normalization is found by mapping a value of 1.0 from source to destination.

[14] Land areas on the WAV grid are ignored and a minimal value 1.e-7 is used for points with positive ocean fraction in the ATM.

must fill values with an outside source. No mapping normalization is used in the HAFS application.

The exchanged fields defined in the *esmFldsExchange* module provides all the required information to map and merge from source to destination component (C.10 CMEPS Field Exchange Table). Given the required mapping types, multiple route handlers are created, depending on the mapping types specified between fields in the source and destination field bundles[15]. Field data within the field bundles which have the same mapping type and normalization type are then loaded into a structure where each source field corresponds to the ungridded dimension of a packed field[16]. This allows for efficient run-time mapping of all fields with a given mapping type. After mapping, the packed field is unwrapped back into the corresponding field.

2.5 Main Driver

UWM is based on the NUOPC layer that sits on top of ESMF. Both layers are designed to make the main program part of the model system itself. The UWM main FORTRAN program is only a few lines of code and is located under *driver/UFS.F90*.

The purpose of the UWM main program is to initialize ESMF, set up the top level driver object, execute the simulation, and cleanly shut down the main program. The details of the top level driver implementation are discussed below. On the main program level, an ESMF_GridComp object is created for the UWM driver component, simulation time information is read from the configuration file *model_configur*e and an ESMF_Clock object is created, taking into account the restart option. The driver object is then initialized, run, and finalized providing the ESMF_Clock object. After successful completion, the main program destroys the objects, and finalizes ESMF.

The UWM driver is implemented under *driver/UFSDriver.F90*, specializing the generic NUOPC_Driver. The public entry point in the UWM driver is its SetServices method via symbol UFSDriver_SS. The driver is responsible for handling build time component options, using preprocessor logic around each of the components' USE statements. Currently 12 component options are implemented. The preprocessor logic in the driver allows components present at build time to be built and run.

The UFSDriver_SS calls NUOPC_CompDerive() to inherit the generic NUOPC_Driver implementation parts. It then specializes the standard specialization points SetModelServices, SetRunSequence, and ModifyCplLists. An ESMF_Config object is created from the

---

[15] The assumption is made that all fields within the same field bundle reside on the same grid or mesh.

[16] Source fields which contain an ungridded dimension are treated in the same way, such that the size of the ungridded dimension in the packed field is increased by the number of source fields with an ungridded dimension times the size of their ungridded dimension.

ufs.configure file and set as the driver component config object. Finally a custom NUOPC field dictionary is read from the file *fd_ufs.yaml*.

The SetModelServices specialization routine is called during driver initialization. It ingests the “EARTH_attributes::”, “DRIVER_attributes::”, and “ALLCOMP_attributes::” configuration labels from the driver config, setting all of the specified attributes on the driver component itself. Next the start type is determined. The supported options are “startup”, “continue”, or “branch”. Attributes are set accordingly on the driver component itself.

The “EARTH_component_list:” configuration label is parsed to obtain the list of run-time components. The 14 options are:

- FV3
- DATM
- HYCOM
- MOM6
- DOCN
- CICE
- WW3
- NOAH
- NOAHMP
- LIS
- IPE
- CMAQ
- GOCART
- CMEPS

Requesting a component at run-time that was not present at build-time triggers an error and application abort.

The components listed in “EARTH_component_list:” are then added to the driver with the appropriate petList and omp_num_threads resource request. Finally the respective component attributes are set, including restart information.

The SetRunSequence specialization routine is also called during driver initialization. This is the place where the run sequence (see below) is ingested from the driver configuration file. Connectors are automatically added as they appear in the run sequence. Only the default generic NUOPC_Connector implementation is used.

Finally the ModifyCplLists specialization routine is called during driver initialization. The CplList attribute of each connector is accessed. For each CplList element the ":DumpWeights=true" and ":SrcTermProcessing=1:TermOrder=SrcSeq" connection options are added. This is in addition to the “ConnectionOptions” attribute string that is read in from the run sequence definition.

2.5.1 Configuration

The *ufs.configure* file is the main configuration file specifying details of the coupled run. The same file is also passed into ESMF_Initialize() as the ESMF default configuration. The standard ESMF configuration label are supported:

- defaultLogFilename:
- logAppendFlag:
- logKindFlag:
- globalResourceControl:

See the ESMF_Initialize() documentation for a detailed description of each option.

Next the UWM driver accesses *ufs.configure* to read the driver configuration. The relevant configuration labels are:

- EARTH_component_list: White space delimited list of component labels. The naming of these labels is not restricted, however, the standard labels are MED, ATM, OCN, ICE, WAV, CHM, etc, for the mediator, atmosphere, ocean, etc.. Multiple instances of the same component are supported using different labels, e.g. ATM-1, ATM-2, etc.
- EARTH_attributes:: List of driver attributes in the format key = value, one per line. The list is terminated by a line containing the string "::" .
- DRIVER_attributes:: List of driver attributes in the format key = value, one per line. The list is terminated by a line containing the string "::" . This is redundant with EARTH_attributes::, and present due to support compatibility with CESM.
- ALLCOMP_attributes:: List of attributes in the format key = value, one per line. The list is terminated by a line containing the string "::" . This list of attributes is set on all components, including the UWM driver component itself.

The *ufs.configure* file contains specific configuration information for each component listed in the "EARTH_component_list:". Assuming a component label ABC, the associated configuration labels are:

- ABC_model: Model or mediator component by name, e.g. "fv3" or "cmeps".
- ABC_petlist_bounds: Lower and upper bound of petList, separated by white space. By default define component ABC on all driver PETs. Each PET provides the associated PE for component execution. The actual number of executing PETs is determined by dividing provided petCount by omp_num_threads.
- ABC_omp_num_threads: Number of threads used per executing PET. By default run single threaded.
- ABC_attributes:: List of attributes in the format key = value, one per line. The list is terminated by a line containing the string "::" .

Only the first label in the list, “ABC_model:”, is required. It provides the association between the component label “ABC” and the actual model component, e.g. “fv3”. All other configuration labels are optional.

The UWM driver sets attributes found under “ABC_attributes::” and “ALLCOMP_attributes::” on each component.

2.5.2 Run sequence

The NUOPC run sequence specifies details about time stepping and integration loops. The run sequence is read from the *ufs.configure* file and loaded into the UWM driver component.

The run sequence definition starts with the “runSeq::” configuration label, and ends with a line that contains the “::” termination string. The lines between those two markers are interpreted as per the NUOPC run sequence definition specified in the NUOPC reference manual.

The basic elements of the NUOPC run sequence syntax are:

- Time stepping loops, starting with a line of the format @T, where T is the loop's time step, and ending with a line containing the @ symbol by itself.
- A line containing a component label by itself executes this component’s default run phase. An alternative run phase can be specified optionally following the component label.
- A line of the format ABC -> XYZ introduces a Connector from component ABC to component XYZ. Connection options can be provided optionally following the Connector syntax. For details about connection options see the NUOPC reference manual.

2.6 Atmosphere dynamics and physics coupling: CCPP Framework

The Common Community Physics Package (CCPP) (Heinzeller et al. 2023) is a collection of atmospheric physical parameterizations for use in Earth system models and a framework that couples the physics to a host model’s dynamical core. A primary goal for this effort is to facilitate research and development of physical parameterizations and experimentation with physics-dynamics coupling methods, while simultaneously offering capabilities for use in numerical weather prediction (NWP) operations. The CCPP Framework supports configurations ranging from process studies to operational NWP as it enables host models to assemble the parameterizations in flexible suites. Framework capabilities include variability of scheme call order, ability to group parameterizations for calls in different parts of the host model allowing intervening computation or coupling to additional components, options to call some parameterizations more often than others, and automatic variable transformations.

The UFS WM calls CCPP from the atmospheric model through the host model physics cap and optionally from within the FV3 dynamical core. Since in the UWM the dynamics time step is shorter than the general physics time step, we refer to the physics called from the dynamical core as tightly coupled or fast physics, as opposed to the traditional or slow physics that are called through the atmospheric host model. The fast physics currently consist of a single scheme, the saturation adjustment step for the GFDL cloud microphysics (Zhou et al., 2019). The UFS WM uses the capability to call individual groups of physics from the Suite Definition File (SDF) to implement the slow and fast physics and perform other operations between the calls to radiation, stochastic physics and the remaining slow physics.

The CCPP Framework requires the host model to manage the data that are requested by the physics. For the slow physics, the FV3ATM atmospheric model allocates the necessary variables, grouped into several derived data types and split into separate blocks for better cache reuse. Blocked data structures in the UWM are used to enable parallel execution of the time integration phase for the physics with multiple OpenMP threads. The members of these data types, i.e. the actual variables required by the physics, are passed from FV3ATM to the CCPP Physics schemes through auto-generated caps as described in detail in Heinzeller et al. (2023). As described in section 2.2.1 in this document, the FV3 dynamical core uses its own set of variables for the dynamics, with a different horizontal decomposition and a different vertical coordinate, etc. Accordingly, the FV3 dynamical core allocates the variables required for the fast physics and passes them to the CCPP Physics schemes.

Recently, the capability to call aerosol-aware parameterizations was added to the UFS WM (Barnes et al., 2022). The CCPP Framework has also been connected to CMEPS (see sections 2.2.8 and 2.4) to enable the computation of thermodynamic fluxes between the atmosphere and the underlying ocean and sea ice using a predefined grid to exchange information among the various component models. The CCPP has been included as a subcomponent of all UWM public releases thus far, with the most recent one being the UFS SRW App v2.2.0. For more information on CCPP, the reader is referred to Heinzeller et al. (2022).

2.7. Configuration of component models

Each coupled application within the UWM is configured at the main Driver level (*model_configure)* and at the coupling level (*ufs.configure*). As explained above (2.5), the *model_configure* file provides Driver-level control of the coupled system while the *ufs.configure* provides application specific control of component models as well as the application's run sequence. Note that for historical reasons, in the UWM, the *model_configure* file contains both ATM-specific control settings (timestep, output type and frequency) as well as Driver level configuration such as the start and end time and integration length. Component level configuration for the individual models within the coupled system can be found within the respective description for each NUOPC cap (2.2) and in Appendix B. A small subset of configuration options applies to multiple components and are set in the ALLCOMP attributes (B.9.1)

For each component model, additional configuration is provided at the internal model level, using one or more namelists (Table 2.7.1). A component's namelist file(s) typically controls internal model physics options as well as configuration settings such as domain size, decomposition and output variables and frequencies. The namelist file(s) for each model component is given below. Please refer to each component's official documentation for details.

Table 2.7.1 Component Namelists

Component Namelist Control		
Component model	**File**	**Details**
FV3atm	input.nml	All physics and coupling options
FV3atm	model_configure	ATM model output frequency and type
FV3atm	diag_table	Output variables
CICE6	ice_in	All model options, output frequency and output variables.
MOM6	MOM_input	All model options
MOM6	diag_table	Output frequency and variables
WW3	ww3_shel.nml	Output and restart frequency. Output variables and expected coupling variables must be consistent with switch file and model initialization file.
WW3	specific switch file	No run-time control of WW3 physics options is possible. All WW3 internal parameterizations are set at compile time via CPP pre-processor directives. Run time configuration of WW3 in ww3_shel.nml must be internally consistent with these compile-time options as well as the WW3 input grid file.
HYCOM	blkdat.input	HYCOM model parameter settings
HYCOM	patch.input	HYCOM model tile settings
HYCOM	port.input	HYCOM model open boundary settings
GOCART	cap.rc	AERO component cap configurations.
GOCART	AERO.rc	AERO root component configurations
GOCART	AERO_HISTORY.rc	Fields in history file
GOCART	AERO_ExtData.rc	Resource file that lists the variables to be read in from external data files
GOCART	GOCART2G_GridComp.rc	Resource file for GOCART2G grid component parameters including instance definition
GOCART	CA2G_instance_CA.bc.rc	Resource file for Black Carbon parameters.
GOCART	CA2G_instance_CA.oc.rc	Resource file for Organic Carbon parameters.
GOCART	DU2G_instance_DU.rc	Resource file for Dust parameters
GOCART	SS2G_instance_SS.rc	Resource file for Sea Salt parameters
GOCART	SU2G_instance_SU.rc	Resource file for Sulfur parameters.
AQM	aqm.rc	Resource file for AQM model
CMAQ	AE_cb6r3_ae6_aq.nml	Particle phase (aerosol) matrix namelist
CMAQ	GC_cb6r3_ae6_aq.nml	Gas phase matrix namelist

CMAQ	NR_cb6r3_ae6_aq.nml	Nonreactive species matrix namelist
CMAQ	Species_Table_TR_0.nml	Tracer species matrix namelist
NOAHMP	ufs_configure	Land model related namelist options
CDEPS	docn_in, datm_in, dlnd_in etc	Datamode, model domain size, and filename for model mesh
CMEPS	N/A	

3. Supported coupled applications

The UWM supports several coupled configurations that are targeted for operational applications. Below is a brief overview of these applications.

3.1 Global coupled prototypes

A series of global coupled prototypes were conducted during the UWM coupling capability development. New components were integrated in the UWM, new features and updates were added in each model component besides the coupling capability. At this time, in the global coupled application, the atmosphere runs on the fractional grid and is two-way coupled with ocean, sea ice, wave and aerosol components. The land component is called within the atmosphere component. Several atmosphere physics schemes have been updated including replacing the GFDL microphysics with Thompson microphysics, and improvement in the convection, planetary boundary layer schemes, and gravity wave drag, for example. The land surface scheme was updated from Noah LSM to Noah-MP LSM. The corresponding changes in coupling fields have been made and tested. The ocean, sea ice and wave models are all two-way coupled, except sea ice-wave coupling, which is one-way from the ice to the wave model. The results from global coupled prototypes can be found in the NCEP notes 510 [NOAA, 2022].

3.2 HAFS coupled system

The Hurricane Analysis and Forecast System (HAFS) is an Unified Forecast System (UFS) hurricane application. It provides tropical cyclone (TC) analyses and forecasts for all global oceanic basins, including TC track, intensity, size and rainfall predictions as well as the large-scale environment flow that is known to influence the TC’s motion. It is a community-based coupled earth modeling system specially calibrated for TC prediction with dynamics and physics, sophisticated vortex initialization, advanced inner-core data assimilation techniques, and various air-sea interaction processes.

The HAFS coupled system is the first UWM based coupled modeling system implemented in operation. The HAFSv1 Initial Operational Capability was implemented in June 2023 running

NCEP's legacy operational hurricane forecast systems, HWRF and HMON. It has two HAFS configurations, HFSA and HFSB. The two configurations are using different CCPP Physics suites, have slight variations in model initialization/warm-cycling and different domain size configurations. Both configurations adopt the two-way atmosphere-ocean coupling between the FV3ATM and HYCOM components. The atmospheric component utilizes a regional storm-centric Extended Schmidt Gnomonic (ESG) grid based moving-nest configuration with a 6-km resolution parent domain nested with a 2-km resolution storm-following domain. The regional atmospheric parent domain is directly coupled to the regional HYCOM ocean domain with a grid spacing ~1/12 degree through the CMEPS mediator. The atmospheric moving nest domain is indirectly coupled to the ocean model component, as it gets the ocean coupling variable (SST) downscaled from its parent after the parent domain receives the SST from the HYCOM domain via CMEPS. Meanwhile, the FV3ATM's two-way nesting capability enables the moving nest to feedback atmospheric variables to its parent, which then drives the HYCOM ocean component through atmospheric forcings. The HFSA configuration also includes the one-way atmosphere and wave coupling, in which the atmospheric component drives the wave component through 10-m winds. A schematic HAFSv1 atmosphere, ocean, and wave domains is shown in Figure 5.

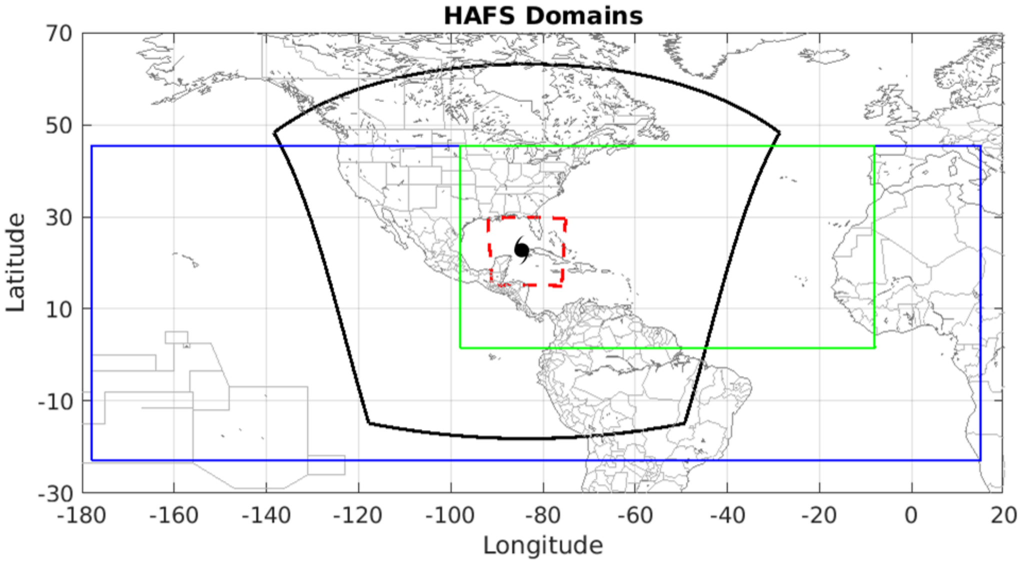


Figure 5: Schematic HAFSv1 domains: FV3ATM parent (black), FV3ATM moving-nest (red), HYCOM ocean domain (blue), and WW3 wave domain (green).

One might notice that, different from a coupled global configuration, the regional HAFS coupled configuration has different domain coverages for different model components. Therefore, in the non-overlapped domain areas, the coupling variables (atmospheric forcings, SST, etc.) have to be pre-processed and prepared from the regional HAFS system's upstream model, GFS.

3.3 Regional AQM

The regional Air Quality Model (AQM) aims to improve the accuracy of air quality forecasts on regional domains. The regional AQM v7 uses the UWM with configuration of online coupling CMAQ v5.2.1 with the FV3 atmosphere. It runs on a single large domain that covers the three operational domains CONUS, Puerto Rico and US Virgin Islands with increased vertical levels to 64. Both atmosphere and chemistry models run on the same horizontal grid with the same vertical coordinates to avoid the uncertainty associated with grid-remapping and vertical interpolation. Higher vertical resolution better resolves the stratospheric intrusion and atmospheric boundary layer processes. Chemical reactions, aerosol processes, and transport are driven by the instantaneous fields at each time step rather than an hourly interval to give consistent interactions between the atmosphere and chemistry. This system uses hourly high resolution FRP-based RAVE wildfire smoke emissions with improved smoke plume rise to improve the representation of wildfire events' impact on air quality prediction. Other emission data updates include using National Emission Inventory (NEI) for CONUS and global emissions for O-CONUS and the MEGAN biogenic emissions. Implementation into operations is planned for early 2024.

3.4 Marine DA forecast model

The marine DA forecast model configuration was originally developed in the UWM for the 40 year analysis NG-GODAS project (Kim et al, 2022) and the marine data assimilation in the global coupled system. The configuration consists of ocean model MOM6, sea ice model CICE6 and a CDEPS data atmosphere model (DATM) coupled through CMEPS mediator. The atmosphere forcing fields are read in the data atmosphere model DATM and passed to CMEPS. The fluxes are computed in CMEPS using bulk formulation along with the SST received from MOM6, then these fields are interpolated and transferred to MOM6 and CICE6 grid components. Because the SST impact has been fed back to the air-sea fluxes, it restricts the forecasted SST drifting away from the analysis. The NG-GODAS prototype reanalysis experiment from 1979 to 2019 was conducted for the assimilation of extensive marine observation data sets, including in-situ temperature and salinity, sea ice concentration, satellite SST and SSS retrievals, and ADT data sets. The results are presented in NOAA note 509 (Kim et al, 2022). This model configuration is currently used to develop the marine data assimilation system for many UFS modeling systems including GFSv17 and GEFSv13.

4. Computational performance and results

4.1 Synchronization

The NUOPC run sequences described in section 2.5.2 are executed asynchronously as possible by the NUOPC_Driver. This carries over to the UWM driver by virtue of being derived from NUOPC_Driver. In practice this leads to minimal synchronization between components, and the concurrent execution of lines in a run sequence where possible.

There are two situations that can lead to synchronization: (1) components run on overlapping persistent execution thread lists (petLists), and (2) a component is the receiver of a Connector that is waiting for data. The sending component of a Connector does not synchronize with the receiver. Instead the delay on the sending side is solely determined by the local copy time of the data into the communication buffer.

Determining the level of concurrency of a specific configuration thus requires understanding of the components' petLists, and the placement of Connectors throughout the run sequence. Here the fully coupled global case is discussed. The run sequence for this case is:

```
@<coupling_interval_slow_sec>
   MED med_phases_prep_ocn_avg
   MED -> OCN :remapMethod=redist
   OCN
   @<coupling_interval_fast_sec>
     MED med_phases_prep_atm
     MED med_phases_prep_ice
     MED med_phases_prep_wav_accum
     MED med_phases_prep_wav_avg
     MED -> ATM :remapMethod=redist
     MED -> ICE :remapMethod=redist
     MED -> WAV :remapMethod=redist
     ATM phase1
     ATM -> CHM
     CHM
     CHM -> ATM
     ATM phase2
     ICE
     WAV
     ATM -> MED :remapMethod=redist
     MED med_phases_post_atm
     ICE -> MED :remapMethod=redist
     MED med_phases_post_ice
```

```
   WAV -> MED :remapMethod=redist
   MED med_phases_post_wav
   MED med_phases_prep_ocn_accum
  @
  OCN -> MED :remapMethod=redist
  MED med_phases_post_ocn
  MED med_phases_restart_write
@
```

Further, MED, ATM, and CHM are defined on overlapping petLists, while OCN, ICE, and WAV each are on exclusive petLists.

On the basis of the associated petLists, the top level, i.e. the <coupling_interval_slow_sec> time loop, supports concurrency between OCN and (MED+@<coupling_interval_fast_sec> block). However, the MED -> OCN and OCN -> MED connectors serialize the execution of OCN and the MED phases located on the top level: med_phases_prep_ocn_avg + med_phases_post_ocn + med_phases_restart_write. In other words, each OCN iteration has to wait for those three MED phases. Luckily the execution time of these three MED phases is very small compared to the cost of each OCN iteration.

The fact that there are no connectors between OCN and the @<coupling_interval_fast_sec> block allows the concurrent execution of both. Thus a well load-balanced system configuration must aim at balancing the cost of each OCN iteration with the cost of executing the @<coupling_interval_fast_sec> iterations.

A similar analysis of the <coupling_interval_fast_sec> time loop indicates that the petList definition and connector placement allow for concurrent execution of ICE, WAV, and (ATM+CHM). Here (ATM+CHM) stands for the

```
ATM phase1
ATM -> CHM
CHM
CHM -> ATM
ATM phase2
```

run sequence section. This section is clearly sequential, both due to the petLists, and the placement of connectors. Typically the feature of reference sharing is used between ATM and CHM, it allows the CHM component to directly operate on the ATM data fields, eliminating the cost of high volume data transfers. In this case the connectors essentially turn into no-ops during execution, but have the important job of negotiating reference sharing during initialization. Still, synchronization between ATM and CHM components is guaranteed when executing the run sequence due to the overlapping petLists, without which reference sharing would not be possible.

Essentially all MED phases inside the <coupling_interval_fast_sec> time loop execute sequentially with ICE, WAV, and (ATM+CHM). There is a small amount of overlap between some of the MED phases with ICE and WAV, due to the placement of the connectors. For practical cases the cost of the MED phases is low compared to the cost of run phase of the model components.

In conclusion, a well load-balanced overall configuration of the fully coupled global model must aim first at balancing the execution times of the ICE, WAV, and (ATM+CHM) run phases. Once balance has been achieved on the <coupling_interval_fast_sec> time loop level, the OCN component needs to be balanced with the @<coupling_interval_fast_sec> iterations on the <coupling_interval_slow_sec> time loop level.

4.2 Impact of component I/O

One challenge often encountered during the load balancing procedure is the impact that component I/O has on the execution time of the component run phase. This is mostly due to output written by model components on certain intervals. Output that is implemented synchronously on the same PETs executing the component's prognostic model delays the return from the component's run phase. Therefore, run phases of the same component vary in time, depending on *whether* output files are written, and also depending on the *volume* of data written at a particular point in time of the simulation.

As a consequence, a configuration that is well balanced for components with I/O disabled, will experience significant imbalance when I/O is enabled. One remedy of this problem is to introduce the asynchronized I/O with dedicated I/O PETs. These PETs handle writing files for a component asynchronously without blocking the forecast of the system. Under UFS, the FV3atm implements a pool of write components responsible for writing FV3 output, without affecting the time a forecast step takes during the run sequence execution.

4.3 Scalability analysis

Scalability is critical for the UWM as various resolutions are utilized in the suite of UFS applications. To ensure that all these tests run efficiently, an analysis has been conducted for the global fully coupled application to investigate the scalability of each model component and coupling overhead in the global coupled configuration. Through this study several computational performance bottlenecks were identified and fixes were implemented to either resolve or mitigate the issue and improve overall coupled application run speed. Section 4.3.1 shows the scalability of five model components. Unless specified, the x-axis in those figures shows the number of cores, which is the number of MPI tasks multiplied by the number of threads. The y-axis is the run time each run takes. The coupling overhead is small in the coupled configurations as shown in the section 4.3.2.

4.3.1 Component Scalability

4.3.1.1 FV3ATM

The standalone atmosphere FV3ATM model was tested on the operational platform WCOSS2 with 25 km resolution C384L127. The configuration is from the GFSv17 High Resolution 1 (GFSv17.HR1) and the test runs 48 hours with 3 hourly history output and 24 hour restart frequency. The time step is 300s. The tests were run with 1, 2 and 4 threads. The scalability is shown in Figure 6. There are two sets of lines. Solid lines are the total time including initialization, run and the finalization time from the 1/2/4 thread tests and the dash lines are the run time only from those runs. From the total time, the FV3ATM at 25 km resolution scales well up to 3600 cores, while from the run time the model scales well to a much larger number of cores (>6000 cores). Comparing the total time and run time, the scalability issue with initialization was identified. It turned out that the longer initialization time associated with the larger number of MPI tasks is due to the slow MPI_ALL2ALL call issue from the current WCOSS2 Slingshot 10 network. There is ongoing work to upgrade the current Cray’s Slingshot 10 network with Slingshot 11 or use UCX network instead to resolve this issue.

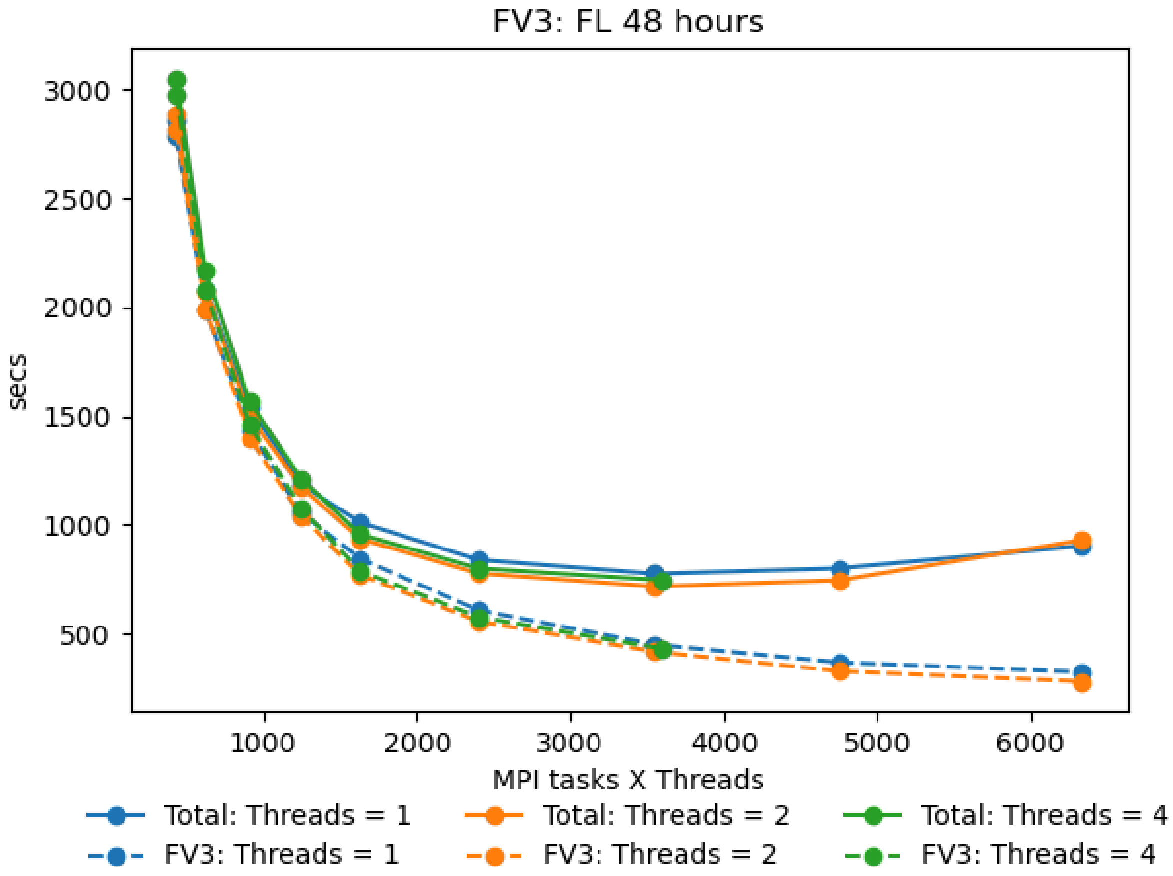


Figure 6: FV3ATM 48 hours run time with the number of MPI cores.

4.3.1.2 MOM6

Standalone MOM6 has been tested on two platforms (HERA and WCOSS2) with different processes settings. The configuration follows the MOM6-examples quarter degree configuration (1440x1080 tripole domain with 75 vertical layers). Ocean time step is 1800 seconds while it uses 3600 seconds time step for its thermal processes. The history file out frequency is 6 hours and restart file frequency is 24 hours, the forecast length is 72 hours. Ocean state variables such as temperature, salinity, currents, sea surface height, along with diagnostic variables such as mixed layer depth and the forcing variables from other components are saved at 6 hour frequency, producing a 1.2 GB file. Figure 7 shows the 72 hours run timing vs process numbers. Overall MOM6 has a good scalability performance for process numbers from 120 to 480. The benefit of using a larger number of processes decreases when it is 960. It is worthy of note that MOM6 threading capability is not mature, all the tests are done with a single thread.

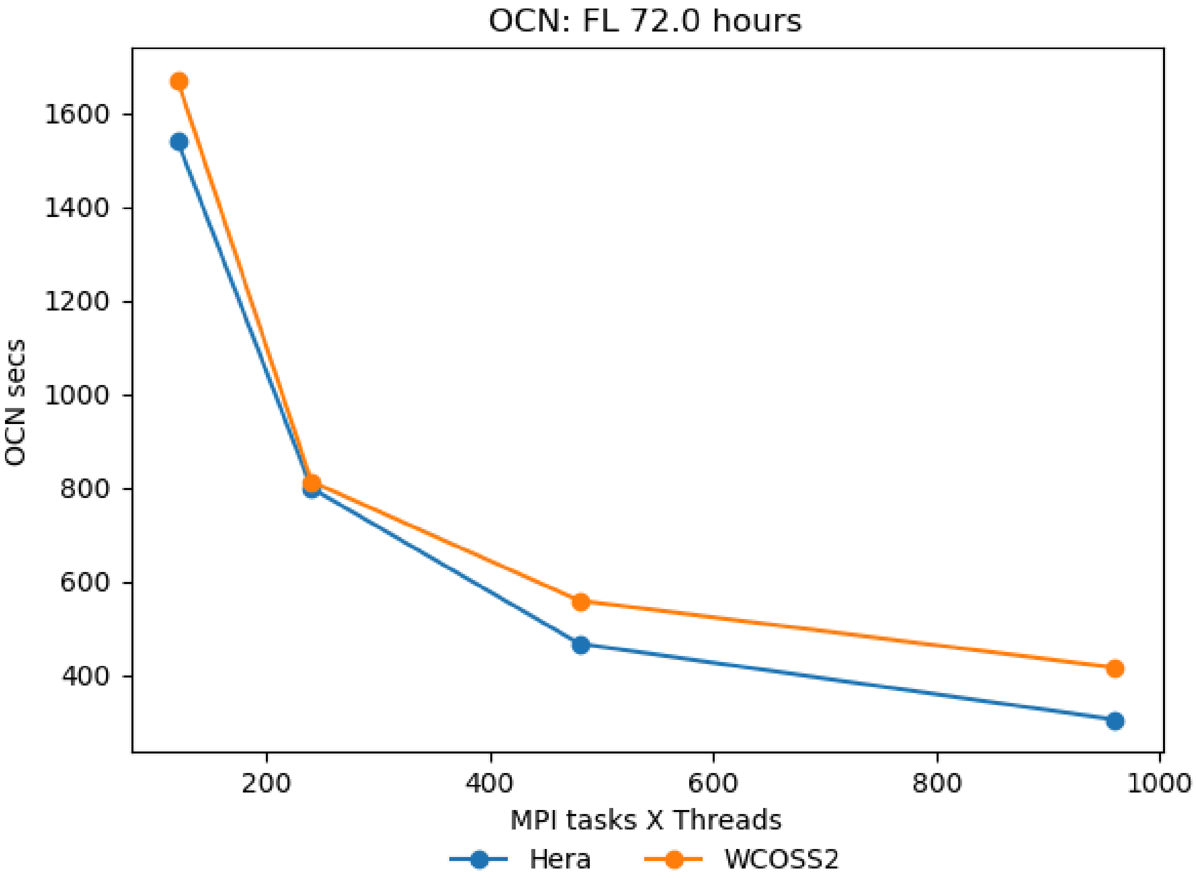


Figure 7: MOM6 24 hours time with the number of MPI cores

4.3.1.3 CICE6

The scalability test was conducted with the CICE tests on WCOSS2. CICE6 is run as a model component in the coupled mode. The resolution is 1/4 degree on a 1440x1080 tripolar grid which overlays on the same ocean grid. The time step is 300s. and history files are updated every 6 hours and restart files output every 24 hours, the forecast length is 24 hours. All tests are using a single thread. Figure 8 shows the scalability of the CICE6 component run time. *SlenderX2* and *SlenderX1* are the two decomposition methods. Overall CICE6 scales well when increasing the number of cores from 48 to 180, the run time is decreased within 4 minutes which is within the required operational time window. There is not much difference between the two decomposition methods.

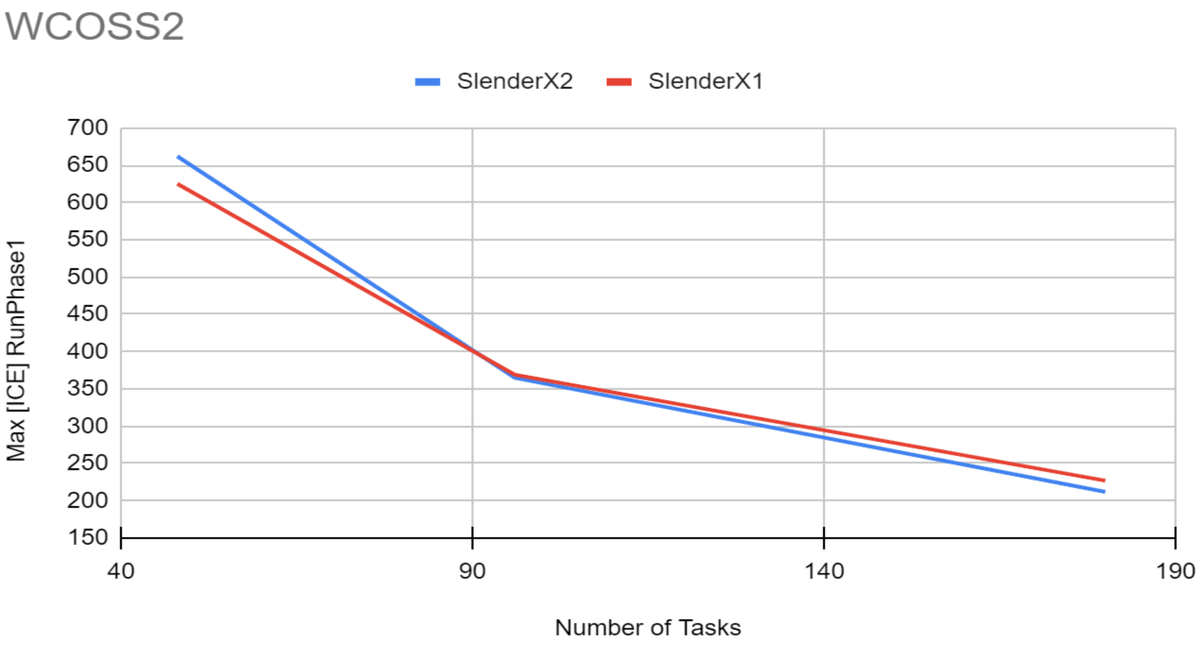


Figure 8: CICE6 24 hours run time with the number of MPI cores.

4.3.1.4 WW3

Scalability tests for WW3 were performed on both WCOSS2 and Hera (shown here), using a tripolar grid with ⅙° resolution, for OMP thread counts=4, 6. Experiments for OMP=1, 2 were also attempted, though the memory available for each task in these configurations is not sufficient for the grid resolution. A global time step of 1800s was used, with forcing inputs of currents, wind, and ice each at a frequency of 1800s. Restart files were written daily with hourly grid and point output during the 5-day run period. Figure 9 shows the scalability of the WAV component run time. The WW3 model scales reasonably with cores less than 4000 with 4 threads and with cores less than 6000 with 6 threads.

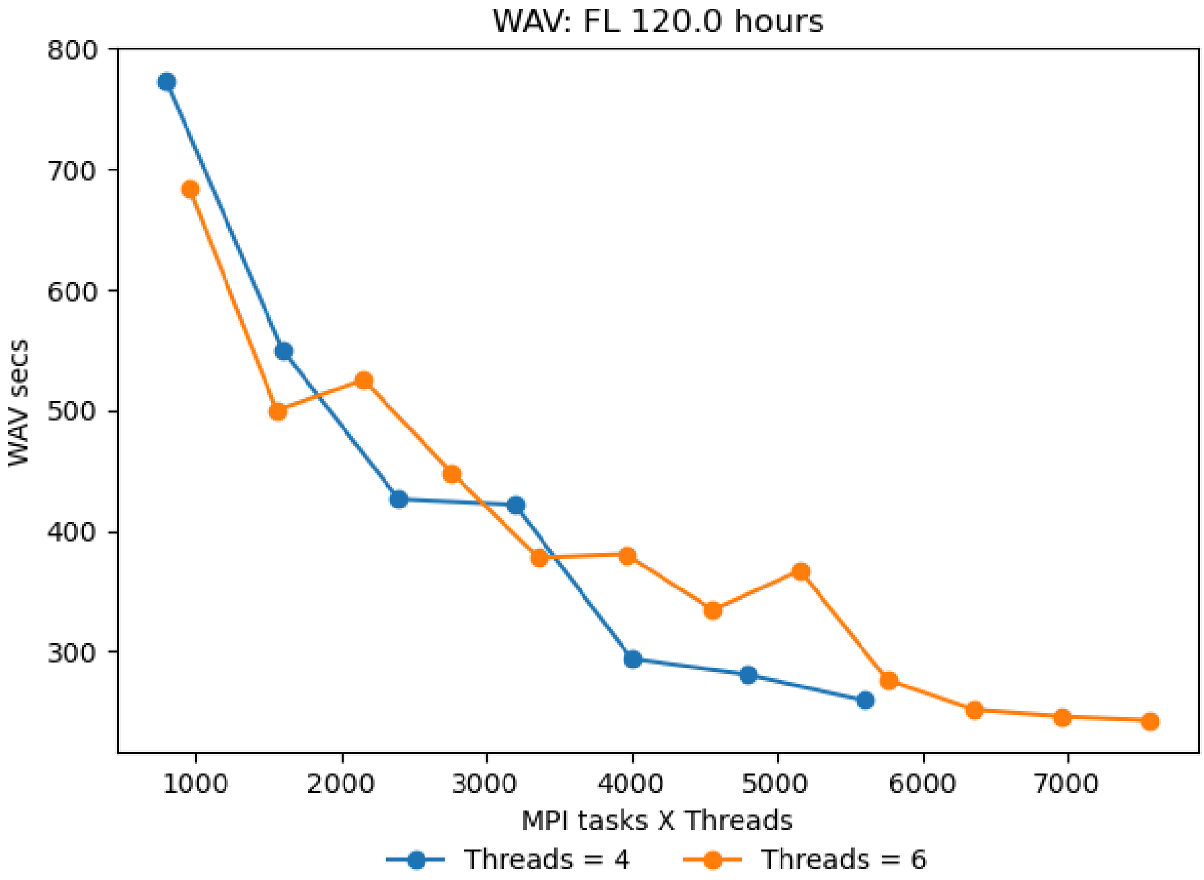


Figure 9: WW3 120 hours run time along with the number of MPI cores.

4.3.1.5 GOCART

The aerosol model GOCART has to run with the atmosphere model FV3ATM and it takes about 30 to 40 percent of total running time. A new version of the GOCART model with threading capability is used to test the model scalability. Figure 10 shows the total time, FV3ATM runtime and the GOCART run time with 1, 2, and 4 threads from the 25km resolution C384L127 FV3ATM and GOCART coupled tests. The forecast length is 24 hours and the time step is 300s. The frequency for history files is 3 hours and the frequency for restart files is 24 hours. The figure shows that the GOCART component scales well up to 4000 cores, after that increasing the number of cores has limited effect. Also the GOCART model does not scale as fast as the FV3ATM, and the FV3ATM still scales well around 7000 cores. When reducing the number of MPI tasks per node and using threads, some of the 4 thread tests need to increase tasks slightly due to the I/O and POST processing.

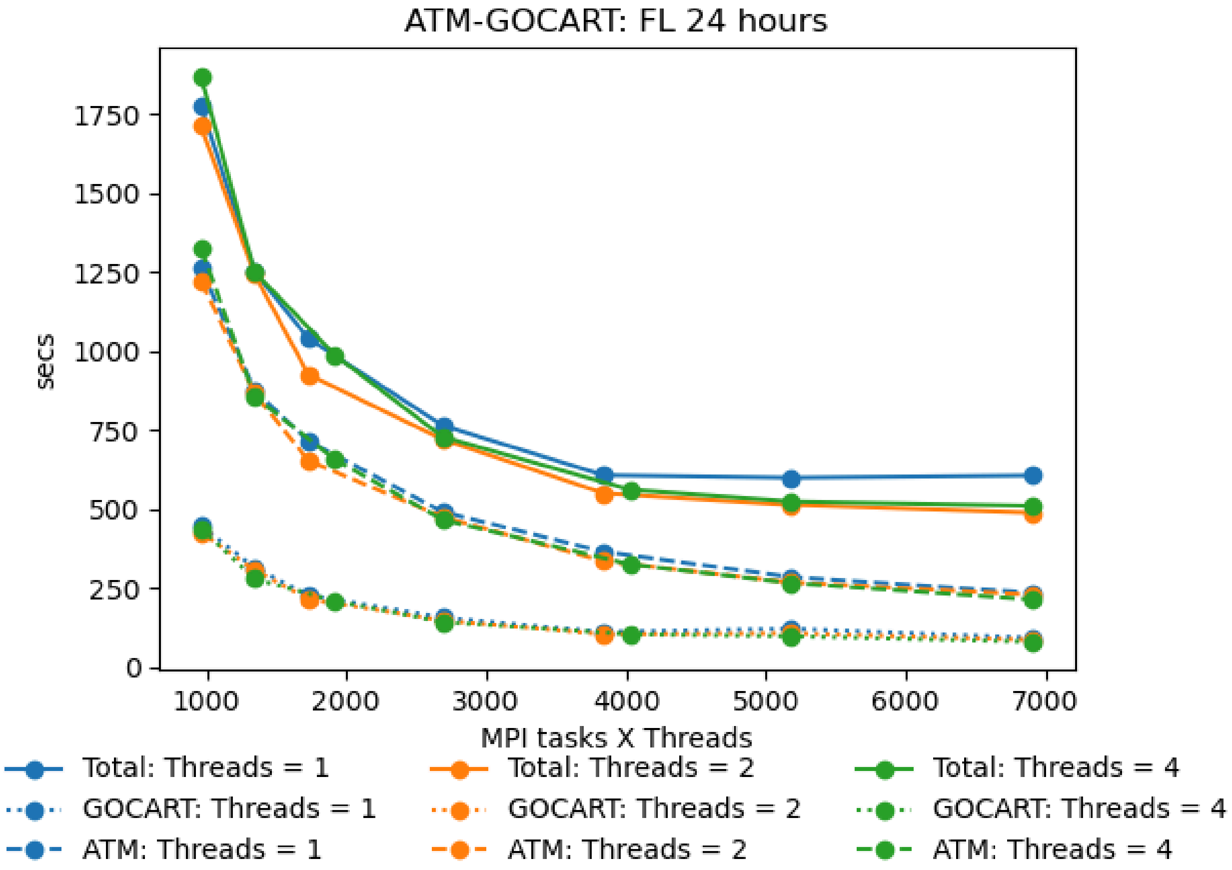


Figure 10. The CHEM scalability results in ATM+CHEM configuration. a) ATM timing b) CHEM timing

4.3.2 Coupling overhead

Many of the UWM applications are using coupled configurations. The individual components interact with each other through the NUOPC coupling infrastructure, in particular through the central CMEPS mediator component. While the overall model performance is significantly affected by the performance of the individual model components, a complete analysis must also consider the cost of coupling.

The coupling overhead adds to the total weather model cost in three different ways. First there is the cost of transferring data between the components (NUOPC Connectors). Second there is the cost of regridding, merging, and otherwise manipulating the data being exchanged

(CMEPS). Third is the cost of synchronization, where a model sits idle, waiting due to data dependencies.

The importance of the indirect cost of synchronization has already been discussed in 4.1 Synchronization. It can be considered an *indirect* cost. In practice, achieving good load balance can be challenging, and the section on ESMF managed threading below provides additional information on the topic. Here we focus on the first two *direct* contributions to the coupling overhead: the cost of data transfers and the cost of manipulating the data being exchanged inside the mediator component.

The total end-to-end execution time of the UWM comprises the initialization, run, and finalization times. For a given configuration, the initialization and finalization times are fixed costs that do not change with the length of the simulation, while the run portion increases on average linearly with the length of the simulated period. For a quantitative discussion we look at a 6h simulation of the cpld_bmark_p8 UWM configuration with active chemistry, executing on 1320 cores across 33 Hera compute nodes. The total end-to-end execution time for this setup is 660s, with 552s from run, 108s from initialization, and a negligible amount from finalization. The direct coupling overhead costs discussed here contribute to both the initialization and the run time.

The majority of the initialization time stems from the individual model initialization phases. For the specific cpld_bmark_p8 case discussed here, ~70s of the initialization cost is determined by the OCN model's internal initialization. The internal initialization of all the other components is much smaller, and happens concurrently while MOM6 (OCN) executes its initialization. This includes the time needed by the CMEPS mediator to compute all of the regridding weights and route handles (~7s). This time is essentially "hidden" since CMEPS runs on a separate set of PET than the much slower OCN initialization.

The remaining ~38s of initialization time can be considered the *coupling overhead contributing to initialization*. This portion breaks into ~18s from actual inter-component communications needed to set up the coupled system (showing up as the sum of Connector times on the OCN PETs), and the balance of ~20s due to NUOPC infrastructure setup. The infrastructure setup covers reading configuration information from files, initializing the NUOPC field dictionary, creating components, starting logging files, and similar tasks. It is important to note that the coupling overhead to initialization includes costs that increase with the number of PETs and the number of components. However, the initialization overhead is a one time cost that is typically amortized over the longer run time for most practical scenarios.

The run time of the model integration loop for the cpld_bmark_p8 case discussed accounts for a total of 552s. However, this number includes the time spent writing data to disk *after* the last step of the coupled integration loop has finished. Subtracting the I/O overhead leads to 516s for the pure model integration time. The bulk of this time (~483s) is determined by the sequential execution of the FV3ATM (ATM) and GOCART (CHM) components on the same 768 cores. The balance of ~33s can be considered the *coupling overhead during run.* This represents 33s/516s ≅ 6% of the total run time.

Further analysis of the coupling overhead shows that roughly half of it (~16s) comes from the 11 CMEPS run phases. However, a single phase, "med_phases_restart_write", accounts for the bulk of the time at ~13s, which is likely dominated by I/O cost! This leaves a net amount of ~3s that can be attributed to the actual manipulation of the exchanged data by the mediator. This includes all of the regridding operations, accounting for approximately 3s/516s ≅ 0.6% of the total integration time.

The other half (~17s) of the coupling overhead during run originates either directly or indirectly from data transfers between components. It turns out that the direct contribution of actually sending and receiving data through the connector components to and from the mediator component is ~3s for the analyzed case. The remaining ~14s stem from the indirect cost of load imbalance, where mediator PETs have to wait for data from concurrently running components. Almost all of the indirect contribution is due to an imbalance with the MOM6 OCN component, and can be traced to the writing of restart files under the OCN run phase at regular intervals. Only considering the direct coupling overhead due to data transfers, leads to another portion of 3s/516s ≅ 0.6% of the total integration time.

In conclusion, the total coupling overhead of the studied cpld_bmark_p8 case with active chemistry component during the run loop is approximately 6%. However, much of this overhead is due to I/O cost in the mediator and the indirect cost of load imbalance with the MOM6 component when writing its restart files. The estimated pure direct coupling overhead for the analyzed case is approximately 1.2%. Half of it is due to manipulation and regridding of the exchanged data by CMEPS and the other half is due to data the cost of sending and receiving data between the components.

4.4 ESMF managed threading

Many configurations of the UWM consist of half a dozen or more model components. The overall performance is strongly affected by the performance characteristics of the individual components. As discussed in the previous section, an efficient configuration is distinguished by good load balancing across all of the interacting components. The scalability of the individual components is a critical feature used to achieve good load balancing when tuning UWM configurations for performance on any given HPC system.

Many of the components implement hybrid parallelism through MPI+OpenMP to improve individual scalability, and to extend it beyond the MPI-only limit. Some configurations also utilize hybrid parallelism to align the component memory footprint with the specific HPC resource. In all cases the optimal choice of the OpenMP thread to MPI task ratio (*threading level*) is highly dependent on the individual component, the specific configuration, and the underlying HPC resource.

In the traditional approach to hybrid parallelism based on MPI+OpenMP, the entire executable is launched with fewer MPI tasks than the number of available hardware cores provided by the job queuing software. This allows OpenMP threading to be utilized under each MPI task. However, since the ratio of the number of cores available to the number of MPI tasks is fixed at the beginning of the execution, and is the same across the entire executable launch, the threading level is the same for each MPI task, and remains constant throughout the duration of execution. It is this constraint of the traditional threading approach that makes it inefficient for the coupled situation where each individual component has its own optimal threading level. The consequence is inefficient HPC utilization, especially for cases that involve components without OpenMP instrumentation. Some platforms provide the task geometry, but not all the HPCs have the capability available.

ESMF managed threading provides a powerful, flexible and portable solution for coupled components that utilize hybrid parallelism. Under this approach the single executable of the coupled system is launched on the exact number of MPI tasks that corresponds to the number of hardware cores to be utilized. No decision about the threading level is made during the executable launch. This decision is moved into the coupled system itself and delayed until each of the constituent components are created at run time.

Under ESMF managed threading, the user specifies the threading level of each component in the *ufs.configure* file, alongside the number of cores given to each component. This is done by utilizing the optional configuration key *ABC_omp_num_threads*, where *ABC* is the name of the component model. Not specifying this key for a component defaults to running the component unthreaded. The information is read by the UWM top level driver and used when the individual components are created.

The ESMF managed threading implementation supports setting different threading levels for components that run on separate sets of cores (concurrent components), as well as for components that run on overlapping sets of cores (sequential components). The only limitation is between components that are coupled using reference sharing. Under the UWM this mode of connection exists between the FV3ATM and the GOCART components. Reference sharing requires that both components have access to the exact same virtual address spaces, i.e. run under exactly the same MPI tasks. Consequently both components must share the same set of cores, and must specify the same threading level.

For a quantitative comparison between the traditional threading approach, and ESMF managed threading, consider the cpld_bmark_p8 UWM configuration with active chemistry component. In this example, the traditional threading approach is configured to provide 2 hardware cores per MPI task across 1360 cores (on 34 Hera compute nodes with 40 cores each). The total end-to-end execution time of the cpld_bmark_p8 configuration under this condition is 762s, with

662s coming from the actual run loop of the coupled model, the difference (100s) accounting for the initialization and finalization time of the model.

In contrast, the ESMF managed threading configuration for the same cpld_bmark_p8 case runs across 1320 cores (on 33 Hera compute nodes). FV3ATM, GOCART, WW3, and CMEPS are set to run 2-way threaded, while MOM6 and CICE6 run single threaded. The total end-to-end execution time for this setup is 660s, with 552s coming from the run loop, and the difference (108s) accounting for the initialization and finalization time of the model.

Clearly the ESMF managed threading approach leads to improved HPC utilization for the cpld_bmark_p8 UWM configuration with active chemistry on the Hera system. The end-to-end execution time is reduced by roughly 15% over the traditional threading approach, while requiring slightly less compute nodes.

4.5 Mixed mode capability with shared FMS library

The Flexible Modeling System (FMS) is a software framework developed and maintained by NOAA/GFDL. The code is open source and available on GitHub for community access. FMS is used by FV3 dycore and the MOM6 ocean component and is integrated in UWM as a prerequisite library.

In the UWM, the FV3 dycore can be compiled with 32-bit (single precision) real data type for floating-point variables to reduce computational cost. It can also be compiled with 64bit (double precision) for better conservation. However, the MOM6 model requires 64-bit (double precision) real data type for consistent physics representation. Since using 32 bit floating-point data type in the FV3 dycore was found to speed up the total atmosphere model run time by up about 30%, without degrading the forecast quality in the GFSv15 and GFSv16 operational configurations, a requirement for the coupled application to use 32-bit FV3 dynamical core and 64-bit MOM6 with the same shared FMS library was identified. Therefore the capabilty to support simultaneous 32-bit and 64-bit data types in commonly used interfaces of the FMS library was developed.

The required capability was implemented by utilizing the unlimited polymorphic data structures in multiple places in the FMS code that are used by both FV3 dycore and MOM6, as well as code changes to the calls of FMS interfaces in the FV3 dynamical core. The updated FMS is compiled in 64-bit type for floating-point variables and for the subroutines used by FV3 dycore, the single-precision parameters of those subroutines are chosen with the polymorphic class (*) parameters in the select type block. Those parameters are then converted to double-precision variables through casting. With these updates, the wall-clock time needed to run a typical fully coupled application including GOCART can be reduced by about 15%. The reduction is smaller than that in the atmosphere only runs due to the sequential run of ATM+CHM as discussed in section 4.1. Without GOCART, the run time of the coupled run can be reduced by about 23% with a different physics package from that used in GFSv16.

5. Application based coupled configurations and test system

Multiple model components are available under UWM as discussed in section 2. Applications with specific focus can be built based on the coupled configurations. The major applications have been shown in section 3. In this section, the UWM build system is explained in section 5.1, the library dependencies are shown in section 5.2 and the test system in section 5.3.

5.1 Build system

CMake is a cross-platform, open-source build system generator that is used to build the various applications in the UWM. Native build scripts using CMake have been developed for the platforms that the build is being performed on. The process starts by choosing the application from Table 5.1. Based on the choice of the configurations (also called applications in CMake build system), the engaged components of the UWM are selected to build the executable. The target application name is used to specify the dependencies and build steps for each application. With this implementation, the UWM build system can support applications with various levels of complexity to satisfy the needs of the operation and research community. The build system has the capability to build one executable with multiple atmosphere physics suites and to build multiple executables simultaneously.

Table 5.1: Major UWM applications

Configurations	FV3atm	MOM6	HYCOM	CICE6	WW3	AERO	AQM	CDEPS	CMEPS	Stochy Phys
ATM	●									●
S2SWA	●	●		●	●	●			●	●
HAFS	●		●		●			●	●	●
UFSAQM	●						●			●
GODAS		●		●				●	●	●

In the UWM, CMakeLists.txt specifies the source files needed for a particular application, libraries that the application depends on, and the build steps required to produce the component library and the final executable. Once CMake has processed the CMakeLists.txt files, it generates platform-specific build scripts, such as makefiles, which can be used to compile the applications.

By using CMake, the build process for the UWM is made platform-independent, as CMake abstracts away the underlying platform-specific build details. Therefore the same CMakeLists.txt file can be used to build the applications on different platforms, such as Linux, and macOS, without having to write platform-specific build scripts.

While CMake is generic, platform specific details e.g. SIMD architecture flags, optimization options are still included in the build framework of the UWM.

5.2 Library support

The UWM prerequisite libraries include the NCEP software libraries (NCEPLIBS) and third party software libraries. These libraries have their own public accessible repositories. Unified software building stacks have been developed through collaborations with the community to allow those libraries to be used by various systems including UWM. The supported software libraries are included in Table 5.2

Table 5.2 UWM Libraries

Third Party libs	udunits; jpeg; zlib; libpng; szip; jasper; gsl; sqlite; libtiff; proj; geos; hdf5; netcdf; nccmp; nco; cdo; pnetcdf; pio; atlas; boost; cgal; cmakemodules; ecbuild; eckit; eigen; esma_cmake; esmf; fckit; fftw; fms; gftl_shared; gptl; gsl_lite; json; json_schema_validator; madis; mapl; met; metplus; miniconda3; pybind11; r2d2; tau2; yafyaml, storch
NCEPLIBS	bacio; sigio; sfcio; gfsio; w3nco; w3emc; sp; ip; ip2; landsfcutil; nemsio; nemsiogfs; g2; g2c; g2tmpl; crtm; upp; wrf_io; bufr; wgrib2; prod_util; grib_util; ncio; ncdiag

The NCEPLIBS libraries were originally developed by NCEP teams following the NCEP EE2 standard to meet the NOAA security and operational technique standards. The installation instructions were developed to support operation and community users who are working on community platforms. The code of these libraries is public on github (https://github.com/orgs/NOAA-EMC/teams/nceplibs/repositories). Each library has its own repository under the umbrella repository and code managers are assigned for each repository. The third party libraries are stored in authoritative github repositories or specific ftp sites. All third party libraries needed to be reviewed and approved through NOAA's software security review process before installation on operational platforms. Some third party libraries that require changes or modifications due to the security requirement will be forked into a repository under the NOAA/EMC github site. The NCEPLIBS code managers maintain and process the changes and updates. Both NCEPLIBS and third party libraries are organized in the unified building stacks.

An umbrella build system is used to install the libraries. Currently the spack-stack system is used in NOAA R&D platform, while hpc-stack system is still used on the operational platforms before the transition to spack stack is finished. The spack-stack framework is a set of configurations and extensions for Spack, which is a community-supported, multi-platform, Python-based package manager originally developed by the Lawrence Livermore National Lab (LLNL). Spack offers a simple spec syntax so that users can specify versions and configuration options to support multiple versions for a package. More details can be found at https://github.com/JCSDA/spack-stack. The HPC-Stack provides a unified, shell script-based build system to build the software stack required for numerical weather prediction models (https://github.com/NOAA-EMC/hpc-stack).

5.3 Test system

The UWM consists of many coupled earth model components developed in the UFS community. It is used for many applications both in production for the operational weather prediction and in research. The UWM is supported and routinely used on several NOAA R&D HPCS and production HPC systems. Maintaining correctness and integrity of such a system is challenging. Thorough testing is a crucial aspect of code management. One of the main steps of the merge process is to run regression tests on all currently supported (Tier-1) platforms. The main purpose of the regression testing is to ensure that changes in one component or one part of the system do not regress other components or features. This is achieved by holding a set of baselines, a known 'good' output from the current version of the code, and making sure that the code that is about to be merged into the main branch either fully reproduces all tests or changes the output only for tests that are expected to be changed. To be able to compare the outputs between two versions of the code, the program must always produce identical results between two executions on the same machine.

The list of tests to be executed along with the compile options are specified in a main configuration file, which is a simple text file that describes which tests are executed and what compiler options are used for a given group of test runs. The default test configuration file is rt.conf and it can be overridden using a command line option.

Each test has a corresponding configuration file in the *tests/* subdirectory. This file sets environment variables that customize runtime options (typically namelist parameters or other run-time configuration options) of each test allowing per test customisation. Regression test supports execution of multiple tests in parallel using ecFlow or Rocoto workflow managers. It is also possible to specify dependencies between different tests, for example tests that run in restart mode usually depend on the corresponding control test and have to wait until the control test is finished. There is also an option to execute just a single test which is useful during feature development.

In the future, several development teams are going to develop the hierarchical testing framework including unit tests and component tests.

6. Summary and Future work

The coupling capability has been developed in the UWM in the past several years. Nine earth model components with their NUOPC caps were developed and integrated into the system. The mediator CMEPS has been developed with enhanced features in data mapping, processing and transferring methods. The ESPS approach was implemented to build coupled applications with selected NUOPC components and CMEPS if necessary. Currently five major coupled configurations have been developed to support global weather forecast and subseasonal to seasonal prediction, regional air quality, hurricane forecast and marine data assimilation systems. The hierarchical build capability using the cross-platform CMake build tool has been developed to support these applications with flexible selection of components.

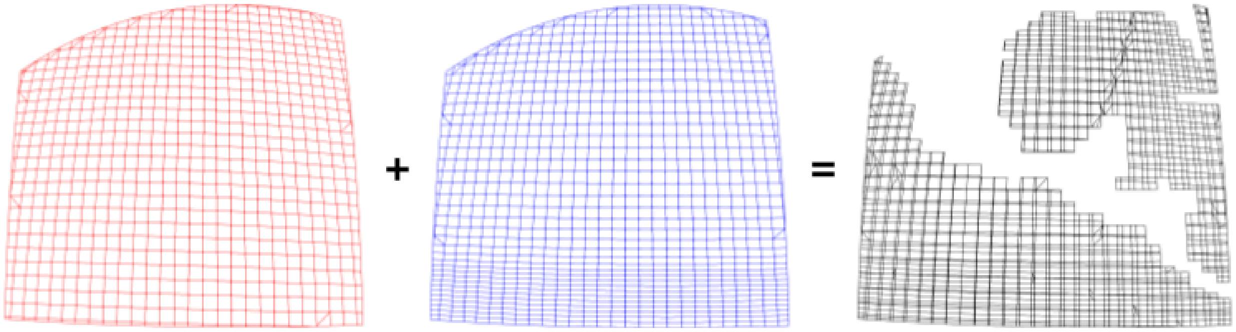
Computational performance has been analyzed using the global coupled applications as an example. Scalability in the FV3ATM, MOM6, CICE6, WW3 and GOCART components was analyzed in the resolution of the latest coupled prototype configuration. All the components are able to run within the operational time window, no major issues are in the component scalability even though the scalability is different in the model components. The threading capability in MOM6 and CICE6 is not mature so that single threads are used in those components. The recent update in GOCART allows it to run with the same threading as used in the atmosphere tasks and it improves the total run time. The overhead of coupling among the components is small. There is ongoing work to analyze the coupled model scalability for high resolution configurations. Future work might be needed to improve the computational performance in those configurations.

The coupled infrastructure is still under active development. Several major development work is listed below:

6.1 Exchange grid

The exchange grid is a two-dimensional composite grid structure formed by overlaying two or more model grids (or meshes) from the participating model components and explicitly exposing cells of overlapping area (Balaji et al., 2006). It is mainly used to resolve the global land/sea mask consistency and supports implicit coupling schemes in cases where the coupled model components do not share the same horizontal grid. The exchange grid approach has been used by several modeling systems including GFDL's climate model and NASA's GEOS modeling system.

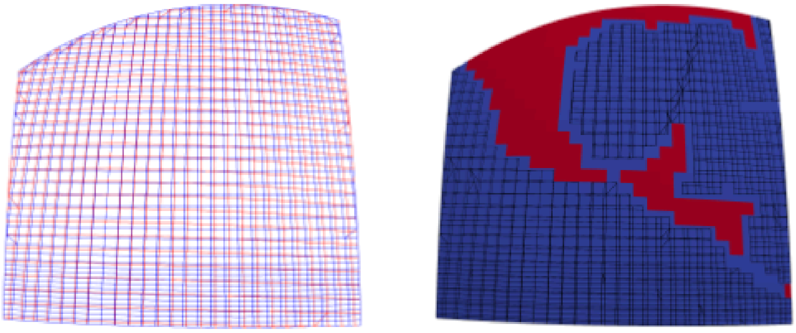
In the UWM, development has started to create the exchange grid by the CMEPS mediator since it has grid (or mesh) information of all the model components. The CMEPS mediator provides functions to evaluate the impacts of different model coupling strategies and can be configured to compute atmosphere-ocean fluxes on different model grids. In this case, the user can select whether to compute these fluxes on the atmosphere grid (*agrid*), the ocean grid (*ogrid*), or an exchange grid (*xgrid*) that is generated automatically during the initialization. The choice of where to compute surface fluxes allows the users to conduct sensitivity studies on the impacts of computing fluxes on different grids. Figure 11, shows a snapshot of an exchange grid constructed by using the global C96 resolution atmospheric model and 1.0 deg. ocean model grids. The initial implementation constructs an exchange grid by using just atmosphere and ocean model grids since the component version of the land model is still under development. This can be easily seen from Figure 11 since the exchange grid is not available over land.



(a) ATM, C96
agrid

(b) OCN, 1.0 deg.
ogrid

(c) Exchange Grid
xgrid



(d) ATM + OCN

(e) ATM Mask + Exchange Grid

Figure 11: Exchange grid subsetted over Central America. (a) atmosphere model grid (cubed-sphere, C96), (b) ocean model grid (1.0 deg.), (c) exchange grid created by ESMF, (d) overlapped atmosphere and ocean model grids and, (e) atmospheric model land-sea mask and exchange grid.

To calculate atmosphere-ocean fluxes in the same way done under the FV3ATM atmospheric model component, CMEPS is extended to include a CCPP host model. In this design, CMEPS is able to drive a very simple CCPP suite file, which includes *sfc_ocean* and other necessary steps that *sfc_ocean* depends on, to calculate atmosphere-ocean fluxes on specified grid (*agrid*, *ogrid* or *xgrid*). Then, it sends back the calculated atmosphere-ocean fluxes such as latent and sensible heat fluxes to the other physical model components. This configuration is exercised using the *ufs.frac.aoflux* coupling mode.

The new atmosphere-ocean flux calculation implementation also enables users to compare and validate the CMEPS calculated atmosphere-ocean fluxes with the ones that are calculated under FV3ATM/CCPP Physics in a very special way. In this configuration, the CMEPS mediator calculates atmosphere-ocean fluxes on the atmospheric model grid by using special *ufs.frac.aoflux* coupling mode and the coupled model still uses the atmosphere-ocean fluxes calculated by FV3ATM. With this implementation, users will be able to perform different experiments with different hıorizontal resolutions to better understand the added value of using an exchange grid to calculate atmosphere-ocean fluxes. Once validated, the exchange will be used as default in the atmosphere-ocean coupling in UWM.

6.2 Hierarchical System Development

The ESMF team is introducing a portable, component agnostic, NUOPC application layer called the Earth System Modeling Executable (ESMX). The ESMX infrastructure includes infrastructure for building and testing standalone NUOPC components and coupled NUOPC components. The package includes the ESMX Driver, a flexible NUOPC driver that incorporates NUOPC components through a build-time YAML file. It also includes an ESMX Data component, a lightweight NUOPC data component that provides prescribed data in export fields and verifies data in import fields. Finally, the package includes a CMake build and test system. The ESMX build system leverages CMake to find libraries and FORTRAN modules then link them into a single coupling capable ESMX application. Tests can be added within the build configuration file for individual components and multiple components. Once complete, the ESMX infrastructure will be able to provide a hierarchical system development framework capable of building and testing individual components using prescribed data and coupled components without modifying the current UWM build and test system.

Additionally, the Earth Prediction Innovation Center (EPIC) is tasked with developing new capabilities to allow the execution of UWM components and subcomponents in isolation and using idealized forcings and configurations. This effort is aimed at testing the UWM building blocks to address biases in the fully coupled system by first identifying and addressing biases and improving individual components.. This effort complements the testing and evaluation of the entire nonlinear system.

6.3 Develop the write grid component in coupled system

The atmosphere model component FV3ATM writes out restart files on the compute tasks that are conducting the forecast integration. Because of that, the forecast will be on hold when writing out restart files which slows down the total forecast time. An asynchronous I/O implementation is being developed to allow the write grid component that runs on different MPI tasks to write out restart files. In this way the forecast tasks can proceed with integration without taking time to write restart files when running the standalone atmosphere. In the coupled mode, since the coupling field exchange only happens on the forecast tasks, this implementation will reduce the running time on the atmosphere grid component and allow the atmosphere to proceed to the coupling stage directly without spending additional time on writing restart files. However, speeding up the atmosphere only does not guarantee a reduced running time in the coupled experiment, as other components also spend time on writing history or restart files. The atmosphere is actually waiting for other components to finish writing before they are ready for the next coupling step as described in the coupling overhead section 4.3.2. Asynchronized I/O capability on other component models in the coupled system is under development in order to remove the I/O dependency in the coupled configurations.

6.4 Develop flexible coupled configurations

Currently the coupled configuration such as the coupled domain, coupling starting and ending time are pre-determined. While retrospective experiments from those configurations are planned for operational implementation, some back up plans to mitigate the risk of unsatisfied forecast skills have been made to develop flexible coupled configurations. Those configurations include changing the coupling domain and coupling starting time. One example is that in the beginning of the forecast the atmosphere is only coupled with the ocean on the tropical domain, then after a certain forecast time during the integration, the atmosphere will be coupled with ocean and ice on the global domain. This feature allows developers to closely look into coupling domain impact on the weather and climate forecast.

Besides the four major developments listed above, several other on-going developments include: 1) Developing the inline coupled data assimilation system within the JEDI framework. It will build the coupling interface (JEDI-FV3) between the generic components of the JEDI system and the UWM which uses FV3 as its atmospheric dynamical core. 2) Developing the seasonal forecast system, including extending the CDEPS data model in UWM and enhancing CMEPS diagnostic and coupling capability to ensure global conservations. 3) Developing regional coupled systems using CMEPS to properly handle incomplete overlapped domains among the component models. 4) Utilizing advanced data compression for I/O as it remains a constant challenge across all the model components while higher and higher resolutions are used. 5) Optimizing the UWM code. One requirement in the operational implementation procedure is to run the system efficiently on operational platforms. There are ongoing efforts to optimize the model component code and the coupling system to meet the requirement.

In summary, the coupling infrastructure is still under active development. Several coupled applications have been developed and are moving towards operational implementations. New features are under development to enhance and extend the coupling capability to support the research community and to improve the operational model forecast skills.

Acknowledgements:

The development of the coupled capability in the UWM cannot succeed without leveraging the expertise in the UFS community. The UFS community developer groups such as the Atmospheric Physics and Dynamics, Marine components, Atmospheric Chemistry, Data Assimilation, Application Support and The Earth Prediction Innovation Center made significant contributions to this work. The co-authors also thank the model infrastructure community including the FMS develop group, MAPL develop group, NetCDF and HDF5 develop teams. We also gave special thanks to developers who left the team or retired including: Minsuk JI, Rocky Dunlap, Kyle Gerheiser, and Shrinivas Moorthi.

This work is supported by NOAA/EMC, the NWS Office of Science and Technology Integration (OSTI) and the Weather Program Office (WPO) of NOAA Oceanic and Atmospheric Research (OAR) via multiple programs (UFS-R2O, SENA, and HPC).

Appendix A: Acronyms

A:
    ATM: The atmosphere
    AQM: Air quality model
C:
    CCPP: The Hybrid Coordinate Ocean Model
    CDEPS: The Community Data Models for Earth Predictive Systems
    CESM: The Community Earth System
    CFS: Climate Forecast System
    CHM: The chemistry model
    CICE5: Consortium sea-ice model version 5
    CICE6: Consortium sea-ice model version 6
    CMAQ: The Community Multiscale Air Quality Modeling System
    CMEPS: The Community Mediator for Earth Prediction Systems
D:
    DA: Data Assimilation
    DATM: Data atmosphere model
    DOCN: Data ocean model
E:
    EE2: Environment Equivalence standards
    EMC: Environmental Modeling Center
    EPA: U.S. Environmental Protection Agency
    ESCOMP: Earth System Community Modeling Portal
    ESMF: Earth System Modeling Framework
    ESMX: The Earth System Model eXecutable
    ESPS: The Earth System Prediction Suite
F:
    FMS: Flexible Modeling System
    FV3: Finite-Volume Cubed-Sphere Dynamical Core
    FV3ATM: atmospheric model with FV3 dynamical core and GFS physics
G:
    GEFS: Global Ensemble Forecast System
    GFDL: Geophysical Fluid Dynamics Laboratory

GFS: Global Forecast System
GOES: The Geostationary Operational Environmental Satellite
GOCART: The Goddard Chemistry Aerosol Radiation and Transport
GSM: Global Spectral Model

H:

HAFS: Hurricane Analysis and Forecast System
HWRF: Hurricane Weather and Research Forecast
HMON: Hurricanes in a Multi-scale Ocean-coupled Non-hydrostatic model
HPC: High Performance Computing
HRLDAS: High Resolution Land Data Assimilation System
HYCOM: The Hybrid Coordinate Ocean Model

I:

IPE: Ionosphere Plasmasphere Electrodynamics

J:

JEDI: Joint Effort for Data assimilation Integration

L:

LIS: Land Information System
LLNL: The Lawrence Livermore National Laboratory
LSM: Land Surface Model

M:

MED: Mediator
MOM4: Modular Ocean Model version 4
MOM5: Modular Ocean Model version 5
MOM6: Modular Ocean Model version 6
MPI: The message passing interface

N:

NASA: National Aeronautics and Space Administration
NCAR: National Center for Atmospheric Research
NCEP: National Centers for Environmental Prediction
NCEPLIBS: The NCEP software libraries
NCO: NCEP Central Operations
NEMS: The NOAA Environmental Modeling System
NetCDF: The network Common Data Form
NG-GODAS: Next Generation Global Ocean Data Assimilation System
NGGPS: Next Generation Global Prediction System
Noah LSM (NOAH): Noah Land Surface Model
Noah-MP (NOAHMP): Noah with multi-parameterization land surface model
NUOPC: The National Unified Operational Prediction Capability

O:

OCN: the model model

P:

PET: Persistent Execution Thread
PIO: Parallel I/O library

R:

R2O: Research to Operation
RDHPCS: Research and Development High Performance Computing Systems

S:

SIS: sea ice simulator
SST: Sea surface temperature

U:

UFS: Unified Forecast System
UWM: UFS weather model
UFSAQM: UFS regional air quality model
UPP: Unified Post processing

W:

WAV: the wave model
WW3: WAVEWATCH III wave model

Y:

YAML: Yet Another Markup Language, a human-readable data-serialization language

Appendix B: Configuration options

B.1 Driver

The following options with the *model_configure* file are for the overall coupled system

Driver Configuration Options

Setting	Description	Valid or Default Values
start_year	model start year	integer
start_month	model start month	integer
start_day	model start day	integer
start_hour	model start hour	integer
start_minute	model start minute	integer
start_second	model start second	integer
nhours_fcst	model forecast time in hours	real
fh_rot	forecast restart of time in hours	real(8)

B.1 FV3ATM

B.1.1 Options set *via* ATM attributes

The following options are passed to the Fv3ATM cap through ATM component attributes in the *ufs.configure* file

ATM Configuration Options

Setting	Description	Valid or Default Values
Profilememory	ESMF Memory Profiling	false
Verbosity	ESMF Verbosity	0
DumpFields	Write import and export states to file	false
OverwriteSlice	Only active when DumpFields is true	true

B.1. 2 Options set *via* input.nml

The following sub-set of options within the ATM *input.nml* are specific to coupled applications within the UWM.

ATM Coupled Configuration options

Setting	Description	Valid or Default Values
frac_grid	fractional grid	FALSE
cplchm	coupled with GOCART	FALSE
cplaqm	coupled with AQM	FALSE
cplflx	coupled with ocean and ice	FALSE
cplice	coupled with ice	FALSE
cplocn2atm	ocean coupled to atm	FALSE
cplwav	one way coupled to wave	FALSE
cplwav2atm	wave coupled to atmosphere	FALSE
cpllnd	coupled with land	FALSE
use_cice_alb	use albedos from CICE	FALSE
cpl_imp_mrg	merge import with internal forcings	FALSE
cpl_imp_dbg	write import data to file post merge	FALSE
use_med_flux	use atmosphere-ocean fluxes imported from mediator	FALSE

B.2 MOM6

The following options are passed to the MOM6 cap through OCN component attributes in the *ufs.configure* file

MOM6 Configuration Options

Setting	Description	Valid or Default Values
Profilememory	ESMF Memory Profiling	false
Verbosity	ESMF Verbosity	0
DumpFields	Write import and export states to file	false
OverwriteSlice	Only active when DumpFields is true	true
mesh_ocn	File name of ESMF Unstructured mesh file	specific to application and resolution
use_coldstart	When false, a lagged startup procedure is used for MOM6	false
use_mommesh	When true, mom6 cap utilizes mesh and not grid	true

B.3 CICE

The following options are passed to the CICE cap through ICE component attributes in the *ufs.configure* file

CICE Configuration Options

Setting	Description	Valid or Default Values
Profilememory	ESMF Memory Profiling	false

Verbosity	ESMF Verbosity	0
DumpFields	Write import and export states to file	false
OverwriteSlice	Only active when DumpFields is true	true
mesh_ice	File name of ESMF Unstructured mesh file	specific to application and resolution
eps_imesh	The allowable error in latitude or longitude between the internal model grid and cap mesh representation	specific to application and resolution
stop_n	Number until stop interval	variable
stop_option	Stop option units	hours
stop_ymd	Stop date (YYYYMMDD)	-999

B.4 WW3

The following options are passed to the WW3 cap through WAV component attributes in the *ufs.configure* file

WAV Configuration Options

Setting	Description	Valid or Default Values
Profilememory	ESMF Memory Profiling	false
Verbosity	ESMF Verbosity	0
mesh_wav	File name of ESMF Unstructured mesh file	specific to application and resolution
user_sets_histname	Override default WW3 file name convention for wave history file output	false
user_sets_restname	Override default WW3 file name convention for wave restart files	true
gridded_netcdfout	Write mean gridded wave parameter output to netCDF	false
history_option	Time unit for history (mean gridded wave field) output alarm	secs,mins, hours,days, monthly, yearly
history_n	Time increment for history alarm	any
history_ymd	History output at date	-999, YMD

B.5 GOCART

The following GOCART configuration options are passed to the GOCART cap through component attributes in the *.rc and ExtData file.

GOCART Configuration Options

Setting	Description	Default Value
NX	Number of grid cells in the two MPI subdomain dimensions	4
NY	Number of grid cells in the two MPI subdomain dimensions	24
IOSERVER_NODES	Atmospheric model configuration parameters	0
DYCORE	This value does nothing but MAPL will crash if it is not declared	NONE
NUM_BANDS	Atmospheric model configuration parameters	30
USE_SHMEM	This setting is deprecated but still has an entry in the file	0
MAPL_ENABLE_TIMERS	Toggles printed output of runtime MAPL timing profilers	NO
MAPL_ENABLE_MEMUTILS	Enable runtime output of the programs' memory usage	NO
PRINTSPEC	(0: OFF, 1: IMPORT & EXPORT, 2: IMPORT, 3: EXPORT)	0
CAP_IMPORTS	Meteorological fields imported from atmospheric model	See Table C.5.2
CAP_EXPORTS	Prognostic Tracers Table	seas*,dust*,dms, msa,so2,so4,bc*,oc*,nh3,nh4a,no3an*
CAP_DIAGNOSTICS	Diagnostic Tracers Table	pm10, pm25

nbins	Aerosol bins	due to aerosols
aerosol_radBands_optics_file	Aerosol radiation band optical file	due to aerosols
aerosol_monochromatic_optics_file	Aerosol monochromatic optical file	due to aerosols
particle_radius_microns	Dry particle radius [um], used for settling	due to aerosols
radius_lower	Dry particle radius lower bound	due to aerosols
radius_upper	Dry particle radius upper bound	due to aerosols
emission_scheme(SST)	SST Emissions method	3
emission_scale	a global scaling factor	due to aerosols
sstEmisFlag	Apply a correction to emissions based on SST	2
hoppelFlag	Apply Hoppel correction (set non-zero, see Fan and Toon 2011)	.false.
weibullFlag	Apply Weibull distribution (set non-zero, see Fan and Toon 2011)	.false.
aircraft_fuel_emission_factor	Aircraft emission factor	due to aerosols
aviation_vertical_layers	Heights [m] of LTO, CDS and CRS aviation emissions layers	0.0 100.0 9.0e3 10.0e3
monoterpenes_emission_fraction	Fraction of biogenic VOCs emissions for SOA production	0.0
isoprene_emission_fraction	Fraction of biogenic VOCs emissions for SOA production	0.0
pom_ca_ratio	Ratio of POM/BRC -> convert source masses from carbon to POM	1.8
hydrophobic_fraction	Initially hydrophobic portion	due to aerosols
fscav	Scavenging efficiency per bin [km-1]	due to aerosols
particle_density	Dry particle density [kg m-3]	due to aerosols
molecular_weight	Molecular weight of species [kg mole-1]	due to aerosols
fnum	Number of particles per kg mass	due to aerosols
particle_radius_number	Number median radius [um]	due to aerosols
rhFlag	Method of apply relative humidity to particle radius	due to aerosols
sigma	Sigma of lognormal number distribution	Due to aerosols
pressure_lid_in_hPa	Pressure lid	0.01
Ch_DU	Resolution dependent tuning constant for emissions (a,b,c,d,e,f) (Ginoux, K14)	0.2 0.2 0.07 0.07 0.07 0.056
maringFlag	Maring settling velocity correction	.true.
emission_scheme(DU)	Dust emissions methods	choose among: fengsha, ginoux, k14
alpha	Fengsha settings	0.04
gamma	Fengsha settings	1.0
vertical_to_horizontal_flux_ratio_limit	Fengsha settings	2.e-04
so4_anthropogenic_fraction	Fraction of anthropogenic emissions that are SO4	0.03
export_H2O2	OH H2O2 NO3 from GMI Combined Stratosphere Troposphere	no
using_GMI_OH	OH H2O2 NO3 from GMI Combined Stratosphere Troposphere	.false.
using_GMI_NO3	OH H2O2 NO3 from GMI Combined Stratosphere Troposphere	.false.
using_GMI_H2O2	OH H2O2 NO3 from GMI Combined Stratosphere Troposphere	.false.
aerosol_monchromatic_optics_wavelength_in_nm_from_LUT	Optics parameters	470 550 670 870
wavelengths_for_profile_aop_in_nm	Optics parameters	470 550 670 870
waelengths_for_vertically_integrated_aop_in_nm	Optics parameters	470 550 670 870
GRID_TYPE	The type of grid	Cubeds-sphere or latlon
IM_WORLD	The number of grid boxes in the i-dimensions	720
JM_WORLD	The number of grid boxes in the j-dimensions	361
POLE	Defines the latitude coordinates of the grid	PC
DATLINE	Define the longitude coordinates of the grid	DC
DU_SRC	Dust input for Ginoux	File

DU_CLAY	Clay fraction for Fengsha input	File
DU_SAND	Sand fraction for Fengsha input	File
DU_SLIT	Silt fraction for Fengsha input	File
DU_SSM	Sediment Supply Map for Fengsha input	File
DU_RDRAG	Albedo drag partition for Fengsha input	File
DU_UTHRES	Threshold velocity for Fengsha input	File
SU_ANTHROL1	SO2 anthropogenic source emissions (surface)	File
SU_ANTHROL2	SO2 anthropogenic source emissions (elev)	File
SU_SHIPSO2	SO2 ship emissions	File
SU_SHIPSO4	Sulfate ship emissions	File
SU_AIRCRAFT	Sulfate emissions from aircraft fuel consumption	File
SU_DMSO	DMS concentration	File
SU_AVIATION_LTO	Sulfate aviation emissions during the three phase of flight	File
SU_AVIATION_CDS	Sulfate aviation emissions during the three phase of flight	File
SU_AVIATION_CRS	Sulfate aviation emissions during the three phase of flight	File
SU_H2O2	H2O2 mixing ratio	File
SU_OH	OH mixing ratio	File
SU_NO3	NO3 mixing ratio	File
pSO2_OCS	Production of SO2 from OCS oxidation	File
OC_ISOPRENE	VOCs offline MEGAN biog (isoprene)	File
OC_LIMO	VOCs offline MEGAN biog (limo)	File
OC_MTPA	VOCs offline MEGAN biog (mtpa)	File
OC_MTPO	VOCs offline MEGAN biog (mtpo)	File
OC_BIOFUEL	OC biofuel source	File
OC_ANTEOC1	OC anthropogenic emissions (surface)	File
OC_ANTEOC2	OC anthropogenic emissions (elev)	File
OC_SHIP	OC ship emissions	File
OC_AIRCRAFT	OC aircraft fuel consumption	File
OC_AVIATION_LTO	OC aviation emissions during the three phases of flight	File
OC_AVIATION_CDS	OC aviation emissions during the three phases of flight	File
OC_AVIATION_CRS	OC aviation emissions during the three phases of flight	File
pSOA_ANTHRO_VOC	OC from SOA production	File
BC_BIOFUEL	BC biofuel source	File
BC_ANTEBC1	BC anthropogenic emission (surface)	File
BC_ANTEBC2	BC anthropogenic emission (elev)	File
BC_SHIP	BC ship emissions	File
BC_AIRCRAFT	BC aircraft fuel consumption	File
BC_AVIATION_LTO	BC aviation emission during the three phase of flight	File
BC_AVIATION_CDS	BC aviation emission during the three phase of flight	File
BC_AVIATION_CRS	BC aviation emission during the three phase of flight	File
BRC_BIOMASS	Brown carbon biomass burning	File
BRC_TERPENE	Brown carbon terpene emission	File
BRC_BIOFUEL	Brown carbon biofuel source	File
BRC_ANTEBRC1	Brown carbon anthropogenic emissions (surface)	File
BRC_ANTEBRC2	Brown carbon anthropogenic emissions (elev)	File
BRC_SHIP	Brown carbon ship emissions	File
BRC_AIRCRAFT	Brown carbon aircraft fuel consumption	File
BRC_AVIATION_LTO	Brown carbon aviation emissions during the three phases of flight	File
BRC_AVIATION_CDS	Brown carbon aviation emissions during the three phases of flight	File
BRC_AVIATION_CRS	Brown carbon aviation emissions during the three phases of flight	File
pSOA_BIOB_VOC	Brown carbon from SOA production	File
EMI_NH3_AG	Nitrate sources from agriculture	File
EMI_NH3_EN	Nitrate sources from energy	File
EMI_NH3_IN	Nitrate sources from industry	File
EMI_NH3_RE	Nitrate sources from residential	File
EMI_NH3_TR	Nitrate sources from transportation	File

EMI_NH3_OC	Nitrate sources from oceanic	File
NITRATE_HNO3	HNO3 emissions	File
NI_regionMask	NI regional mask	File
SU_BIOMASS	SO2 from GBBEPx biomass emissions	File
OC_BIOMASS	OC from GBBEPx biomass emissions	File
BC_BIOMASS	BC from GBBEPx biomass emissions	File
EMI_NH3_BB	NH3 from GBBEPx biomass emissions	File

B.6 CMAQ

The following CMAQ configuration options are passed to the CMAQ cap through component attributes in aqm.rc file.

AQM Configuration Options

Setting	Description	Default Value
ae_matrix_nml	Aerosol species definition namelist	
gc_matrix_nml	Gas-chemistry species definition namelist	
nr_matrix_nml	Non-reactive species definition namelist	
tr_matrix_nml	Tracers species definition namelist	
csqy_data	Cross-Section and Quantum Yield data	
optics_data	Optical properties for cloud water and ice plus the refractive indices for aerosol species	
omi_data	Ozone Monitoring Instrument (OMI) profiles	
init_concentrations	Species initialization	true
run_aerosol	enable aerosol module	true
run_rescld	enable wet deposition processes in resolved-scale clouds	true
mie_optics	compute aerosol optical properties using Mie theory	false
mp_tracer_map	Microphysics scheme used by coupled atmospheric model: supported options are: gfdl, thompson, zhao-carr, wsm6	
ctm_aod	Compute and export Aerosol Optical Depth (AOD)	true
ctm_pmdiag	Compute and export PM2.5 mode fractions as diagnostic tracers	true
ctm_stdout	Output verbosity	all
emission_sources	Input emissions: List emission sources using custom names, then enter emission details for each source using keywords starting with the custom name as prefix, followed by underscore ('_')	
myemis_type	Emissions type	
myemis_format	Emissions file format (binary, netcdf)	
myemis_file	Emissions file name	
myemis_frequency	Emissions time dependency (hourly, daily, weekly, monthly, static)	
myemis_species	List of emission species	

B.7 HYCOM

The following options are passed to the HYCOM cap through OCN component attributes in the *ufs.configure* file.

HYCOM Configuration Options

Setting	Description	Default Value
Verbosity	ESMF output level for logs.	0
Diagnostic	ESMF output level for import and output fields.	0
cdf_impexp_freq	(deprecated) Use Diagnostic setting.	9999
cpl_hour	(deprecated) Coupling time step provided by driver.	0
cpl_min	(deprecated) Coupling time step provided by driver.	0
cpl_sec	(deprecated) Coupling time step provided by driver.	0

base_dtg	(deprecated) Base date-time provided by driver.	9999999999
hycom_arche_output	Create a HYCOM "archive-like" file from "ice" Import/Export state.	true
hyc_esmf_exp_output	(deprecated) Use Diagnostic setting.	true
hyc_esmf_imp_output	(deprecated) Use Diagnostic setting.	true
hyc_impexp_file	File containing import and export field configuration.	<none>
espc_show_impexp_minmax	Write import and export field min/max values to standard output	true
import_setting	Mark import fields as required, optional, or unused.	FLEXIBLE
import_diagnostics	Write out diagnostics for field merging.	false
merge_all_import	Turn on/off field merging, which provides values to unmapped cells.	false
skip_first_import	Skip copying import data from cap to model during first timestep.	false
ocean_start_dtg	Used to initialize HYCOM start date-time.	<none>
start_hour	Used to initialize HYCOM start date-time.	0
start_min	Used to initialize HYCOM start date-time.	0
start_sec	Used to initialize HYCOM start date-time.	0
end_hour	Used to initialize HYCOM end date-time.	0
end_min	Used to initialize HYCOM end date-time.	0
end_sec	Used to initialize HYCOM end date-time.	0

B.8 NOAH-MP

The following options are passed to the Noah-MP cap through LND component attributes in the *ufs.configure* file.

Noah-MP Configuration Options

Setting	Description	Valid Values
case_name	Case name used as a prefix in the file naming for I/O	any
mosaic_file	The path and name of the mosaic grid file in Grid Spec format such as INPUT/C96_mosaic.nc	any
input_dir	The directory that stores initial conditions, static information and grid related files that are pointed by mosaic file	Any (default is INPUT/)
restart_dir	The directory that has restart file	any (default is RESTART/)
ic_type	Indicates the source of the initial conditions. Two options are supported ‘custom’ (i.e. C96.initial.tile[1-6].nc) and ‘sfc’ (default, sfc_data.tile[1-6].nc).	custom, sfc (default is sfc)
start_type	Option for defining the run type (cold or warm start)	startup, continue
layout	Defines decompositions in each direction on each tile (i.e. 3:8 for C96). This needs to be consistent with resolution.	any number pair separated with double column
num_soil_levels	Number of soil levels used by Noah-MP Land Model	any number
forcing_height	Height of the atmospheric forcing in meters. Normally used to overwrite the height provided by CDEPS or active atmosphere.	any number
soil_level_thickness	Thickness of the soil levels. Needs to be consistent with num_soil_levels	Any list of number separated with double column
soil_level_nodes	Depths of the node points for each soil level.Needs to be consistent with num_soil_levels	Any list of number separated with double column
dynamic_vegetation_option	Options for dynamic vegetation (idveg)	1 (off), 2 (on)
canopy_stomatal_resistance_option	Canopy stomatal resistance (iopt_crs)	1 (ball-berry), 2 (jarvis)

soil_wetness_option	Options for soil moisture factor for stomatal resistance (iopt_btr)	1 (noah), 2 (clm), 3 (ssib)
runoff_option	Options for runoff and groundwater (iopt_run)	1 (simgm), 2 (simtop), 3 (schaake96), 4 (bats)
surface_exchange_option	Options for surface layer drag coefficient (iopt_sfc, ch and cm)	1 (m-o),2 (chen97)
supercooled_soilwater_option	Options for supercooled liquid water (iopt_frz)	1(ny06), 2 (koren99)
frozen_soil_adjust_option	Options for frozen soil permeability (iopt_inf)	1 (ny06), 2 (koren99)
radiative_transfer_option	Options for radiation transfer (iopt_rad)	1 (gap=f(3d,cosz), 2 (gap=0), 3 (gap=1-fveg)
snow_albedo_option	Options for snow surface albedo (iopt_alb)	1 (bats), 2 (class)
precip_partition_option	Options for rainfall & snowfall (iopt_snf)	1 (jordan91), 2 (bats), 3 (noah)
soil_temp_lower_bdy_option	Options for lower boundary of soil temperature (iopt_tbot)	1 (zero-flux), 2 (noah)
soil_temp_time_scheme_option	Options for snow/soil temperature time scheme (only layer 1, iopt_stc)	1 -> semi-implicit; 2 -> full implicit (original noah)
surface_evap_resistance_option	Option for surface resistance to evaporation/sublimation (iopt_rsf)	1->sakaguchi/zeng; 2->seller; 3->mod sellers; 4->1+snow
surface_thermal_roughness_option	Option for thermal roughness scheme option (iopt_trs)	1->z0h=z0; 2->rb reversed
surface_diagnose_approach_option	Option for surface 2m t/q diagnostic approach (iopt_diag)	
glacier_option	Options for glacier treatment (fixed to 2 in noahmpdrv)	(1->phase change; 2->simple)
output_freq	Options for output frequency in seconds	any number (default is 21600)
output_mode	Option for output mode to control number of output fields that are written to restart and history files	all, mid, low
restart_freq	Options for restart output frequency in seconds	Any number (default is equal to output_freq)
do_mynnedmf	Option for MYNN-EDMF (default is false)	true or false (default is false)
do_mynnsfclay	Option for MYNN surface layer scheme (default is false)	true or false (deafult is false)
soil_type_category	Option for soil type (isot)	0 (Zobler - 9 category), 1 (STATSGO - 19 category, default), 2 (STAS-RUC - 19 category)
veg_type_category	Option for source of vegetation data (ivegsrc)	0 (USGS), 1 (IGBP, default), 2 (UMD), 3 (NLCD40), 4 (USGS-RUC), 5 (MODI-RUC)
initial_emiss	Option for initial surface lw emissivity in fraction (default value is 0.95)	any number between 0-1
initial_albedo	Option for initial mean surface albedo (default value is 0.2)	any number between 0-1
has_export	Option to enable export fields (default value is true)	true or false
calc_snet	Option to calculate shortwave radiation internally (default value is false)	true or false

B.9 CMEPS

B.9.1 Options set via MED attributes

MED Configuration Options

Setting	Description	Valid Values
atm_model	The name of the atmosphere model	fv3,cdeps
ice_model	The name of the sea ice model, if present	cice
ocn_model	The name of the ocean model, if present	mom6, hycom
wav_model	The name of the wave model, if present	ww3
lnd_model	The name of the land model, if present	noahmp
history_option_[component]_avg	Time unit for mediator history files for named component, averaged (optional)	nsecs,nminutes,nhours, ndays,monthly, yearly
history_n_[component]_avg	Time averaging increment for mediator files for named component (optional)	any
history_option_[component]_inst	Time unit for instantaneous mediator history files for named component (optional)	nsteps,nsecs, nminutes,nhours,ndays, monthly,yearly
history_n_[component]_inst	Time increment for mediator history files for named component (optional)	any
coupling_mode	coupling mode	ufs.nfrac, ufs.frac, ufs.nfrac.aoflux hafs
history_tile_atm	size of ATM tile; if set, mediator history files for ATM will be written on 6 tiles of given dimension	any cubed-sphere dimension (96, 192, etc)

B.9.1 Options set via ALLCOMP attributes

ALLCOMP Configuration Options

Setting	Description	Valid Values
start_type	Initialization type	startup, continue
restart_n	Time increment for restart	any
restart_option	Time unit for restart_n	nsteps,nsecs, nminutes,nhours,ndays, monthly,yearly
restart_ymd	Restart date	-999,YMD
restart_dir	Directory for mediator restart file	any
dbug_flag	Integer value specifying debug verbosity for CMEPS, CICE, MOM6 and WW3. When > 0, field debugging information is written to PET logs	any
case_name	Application name	application specific
wave_coupling_to_cice	Enable additional import and export fields for wave-cice coupling	true,false
ScalarFieldName	Name of string field used to pass component scalar values to CMEPS.	cpl_scalars
ScalarFieldCount	Number of scalar fields.	application specific
ScalarFieldIdxGridNX	Index of ScalarFieldCount which contains the gridNX value	1
ScalarFieldIdxGridNY	Index of ScalarFieldCount which contains the gridNY value	2
ScalarFieldIdxNextSwCday	Index of ScalarFieldCount which contains the NextSwCday	3
stop_option	Time unit for model run duration	nsteps,nsecs, nminutes,nhours,ndays, monthly,yearly
stop_n	Time increment for model run duration	any
stop_ymd	Model stop time specified in YMD	-999,YMD

flux_albav	When true, fixed albedos are specified for the ocean	true, false
use_nextswcday	When true, the mediator ScalarFieldIdxNextSwCday is used for calculation of ocean albedo	true,false
use_min_albedo	When true, enforce a minimum value of ocean albedo	true,false
min_abedo	The value associated with use_min_albedo	0 - 1.0
albdif, albdir	The diffuse and direct albedo used when flux_albav is true	0 - 1.0
orb_eccen, orb_iyear, orb_iyear_align, orb_mode, orb_mvelp, orb_obliq	orbital parameters for calculation of ocean albedo	various

Appendix C: Description of the import and export coupled fields in model component caps in UWM

C.1 FV3ATM

Table C.1.1 FV3ATM Vertical Type Definitions

VERTICAL TYPES	MEANING	LEVELS
G	soil levels	3D
I	model interfaces	3D
L	model levels	3D
S	surface	2D
T	tracers	4D

Table C.1.2 FV3ATM Import Fields

ATM Import Field	Units	Field Description	Level Type	Field Source
stc	K	Soil temperature	G	JEDI
smc	fraction	Volume fraction of condensed water in soil	G	JEDI
u	m s-1	D grid zonal wind	L	JEDI
v	m s-1	D grid meridional wind	L	JEDI

ua	m s-1	A grid zonal wind	L	JEDI
va	m s-1	A grid meridional wind	L	JEDI
t	K	Temperature	L	JEDI
delp	Pa	Pressure thickness	L	JEDI
sphum	kg kg-1	Specific humidity	L	JEDI
ice_wat	kg kg-1	Cloud ice mixing ratio	L	JEDI
liq_wat	kg kg-1	Cloud water mixing ratio	L	JEDI
o3mr	kg kg-1	Ozone mixing ratio	L	JEDI
inst_tracer_diag_aod	Numeric	Instantaneous aerosol optical depth	S	CMAQ
phis	m2 s-2	Surface geopotential	S	JEDI
u_srf	m s-1	Surface u-wind	S	JEDI
v_srf	m s-1	Surface v-wind	S	JEDI
slmsk	Numerical	Land sea mask	S	JEDI
weasd	mm	Liquid water equiv of accumulated snow depth over land and sea ice	S	JEDI
tsea	K	Sea surface temperature from JEDI	S	JEDI
vtype	Numerical	Vegetation type classification	S	JEDI
stype	Numerical	Soil type classification	S	JEDI
vfrac	fraction	Areal fractional cover of green vegetation	S	JEDI
snwdph	mm	Liquid water equivalent snow depth	S	JEDI
f10m	ratio	Ratio of wind at surface adjacent layer to wind at 10m	S	JEDI
zorl	cm	Surface_roughness_length	S	JEDI

t2m	K	2 meter air temperature	S	JEDI
inst_tracer_mass_frac	fraction	Instantaneous tracers	T	GOCART/CMAQ

See Table C.10.1 States To ATM and Table C.10.2 Fluxes To ATM for details of fields of import fields when FV3ATM is coupled to external components via CMEPS.

ATM Import Field	Field Alias (used by CMEPS)	Units	Field Description
ice_fraction	Si_ifrac	ND	Sea ice area fraction
sea_surface_temperature	So_t	K	Sea surface temperature
sea_ice_surface_temperature	Si_t	K	Surface skin temperature over ice
sea_ice_volume	Si_vice	m	Sea ice thickness
snow_volume_on_sea_ice	Si_vsno	m	Liquid water equivalent surface snow depth from coupled process
inst_ice_[ir,vs]_[dif,dir]_albedo	Si_a[ni,vs][df,dr]	ND	Instantaneous [NIR, visible] [difffuse,direct] albedo over sea ice
wave_z0_roughness_length	Sw_z0	ND	Surface roughness length from wave model
inst_snow_area_fraction_lnd	Sl_sfrac	ND	Surface snow area fraction over land
inst_temp_height2m_lnd	Sl_tref	K	Temperature at 2m over land
inst_spec_humid_height2m_lnd	Sl_qref	kg-1 kg-1	Specific humidity at 2m over land
inst_spec_humid_lnd	Sl_q	kg-1 kg-1	Surface specific humidity over land
inst_drag_wind_speed_for_moment	Sl_cmm	m s-1	Surface drag wind speed for momentum

um			
inst_drag_mass_flux_for_heat_and_moisture	Sl_chh	kg m-2 s-1	Surface drag mass flux for heat and moisture
inst_func_of_roughness_length_and_vfrac	Sl_zvfun	ND	Function of surface roughness length and green vegetation fraction
lwup_flx_ice	Faii_lwup	W m-2	Surface upwelling longwave flux for coupling
laten_heat_flx_atm_into_ice	Faii_lat	W m-2	Surface upward latent heat flux from coupled process
sensi_heat_flx_atm_into_ice	Faii_sen	W m-2	Surface upward sensible heat flux from coupled process
stress_on_air_ice_[zonal,merid]	Faii_tau[x,y]	N m-2	Surface [x,y] momentum flux from coupled process
laten_heat_flx_atm_into_ocn	Faox_lat	W m-2	Latent heat flux over ocean (from CMEPS)
sensi_heat_flx_atm_into_ocn	Faox_sen	W m-2	Sensible heat flux over ocean (from CMEPS)
lwup_flx_ocn	Faox_lwup	W m-2	Longwave up flux over ocean (from CMEPS)
stress_on_air_ocn_[zonal,merid]	Faox_tau[x,y]	N m-2	Surface [x,y] momemtum flux over ocean (from CMEPS)
inst_laten_heat_flx_lnd	Fall_lat	kg kg-1 m s-1	Surface latent heat flux over land, converted to evaporative flux
inst_sensi_heat_flx_lnd	Fall_sen	W m-2	Surface sensible heat flux over land
inst_potential_laten_heat_flx_lnd	Fall_evap	W m-2	Surface upward potential latent heat flux over land
inst_upward_heat_flux_lnd	Fall_gflx	W m-2	Soil heat flux over land
inst_runoff_rate_lnd	Fall_roff	kg m-2 s-1	Surface runoff over land
inst_subsurface_runoff_rate_lnd	Fall_soff	kg m-2 s-1	Sub-surface runoff over land

Table C.1.3 FV3ATM Export Field Information

ATM Export Fields	Units	Field Description	Level Type	Field Destination
temperature_of_soil_layer	K	Soil temperature	G	CMAQ
inst_soil_moisture_content	fraction	Volume fraction of condensed water in soil	G	GOCART, CMAQ
stc	K	Soil temperature	G	JEDI
smc	fraction	Volume fraction of condensed water in soil	G	JEDI
inst_pres_interface	Pa	Air pressure at model layer interfaces	I	GOCART
inst_geop_interface	m2 s-2	Geopotential at model layer interfaces	I	GOCART, CMAQ
inst_ice_nonconv_tendency_levels	kg m-2 s-1	Instantaneous 3D flux of ice from nonconvective precipitation	L	GOCART
inst_liq_nonconv_tendency_levels	kg m-2 s-1	Instantaneous 3D flux of liquid water from nonconvective precipitation	L	GOCART
inst_pres_levels	Pa	Mean layer pressure	L	GOCART, CMAQ
inst_geop_levels	m2 s-2	Geopotential at model layer centers	L	GOCART, CMAQ
inst_temp_levels	K	Temperature updated by physics	L	GOCART, CMAQ
inst_zonal_wind_levels	m s-1	Zonal wind updated by physics	L	GOCART, CMAQ
inst_merid_wind_levels	m s-1	Meridional wind updated by physics	L	GOCART, CMAQ
inst_cloud_frac_levels	fraction	Instantaneous 3D cloud fraction	L	GOCART, CMAQ
u	m s-1	D grid zonal wind	L	JEDI
v	m s-1	D grid meridional wind	L	JEDI

ua	m s-1	A grid zonal wind	L	JEDI
va	m s-1	A grid meridional wind	L	JEDI
t	K	Temperature	L	JEDI
delp	Pa	Pressure thickness	L	JEDI
sphum	kg kg-1	Specific humidity	L	JEDI
ice_wat	kg kg-1	Cloud ice mixing ratio	L	JEDI
liq_wat	kg kg-1	Cloud water mixing ratio	L	JEDI
o3mr	Kg kg-1	Ozone mixing ratio	L	JEDI
soil_type	Numerical	Soil type for lsm	S	CMAQ
inst_vegetation_area_frac	fraction	Areal fractional cover of green vegetation	S	CMAQ
inst_sensi_heat_flx	W m-2	Instantaneous sfc sensible heat flux	S	CMAQ
inst_laten_heat_flx	W m-2	Instantaneous sfc latent heat flux	S	CMAQ
inst_pres_height_surface	Pa	Instantaneous surface air pressure for coupling	S	CMAQ
inst_net_sw_flx	W m-2	Instantaneous net sfc downward sw flux	S	CMAQ
canopy_moisture_storage	kg m-2	Canopy water amount	S	CMAQ
inst_aerodynamic_conductance	m s-1	Momentum exchange coefficient	S	CMAQ
inst_canopy_resistance	s m-1	Aerodynamic resistance in canopy	S	CMAQ
leaf_area_index	Numeric	Leaf area index	S	CMAQ
height	m	Height above mean sea level	S	CMAQ
inst_surface_soil_wetness	fraction	Normalized soil wetness	S	GOCART

inst_up_sensi_heat_flx	W m-2	Instantaneous upward sensible heat flux for chemistry coupling	S	GOCART
ice_fraction_in_atm	fraction	Ice fraction over open water in atmosphere	S	GOCART
lake_fraction	fraction	Fraction of horizontal grid area occupied by lake	S	GOCART
ocean_fraction	fraction	Fraction of horizontal grid area occupied by ocean	S	GOCART
inst_pbl_height	m	Atmosphere boundary layer thickness	S	GOCART, CMAQ
surface_cell_area	m2	Area of the grid cell	S	GOCART, CMAQ
inst_convective_rainfall_amount	m	Total convective precipitation	S	GOCART, CMAQ
inst_friction_velocity	m s-1	Surface friction velocity	S	GOCART, CMAQ
inst_rainfall_amount	m	Total rain and snow precipitation	S	GOCART, CMAQ
inst_surface_roughness	cm	Surface roughness length	S	GOCART, CMAQ
inst_land_sea_mask	Numeric	Land sea mask	S	GOCART, CMAQ
surface_snow_area_fraction	fraction	Surface snow area fraction	S	GOCART, CMAQ
inst_[zonal,merid]_wind_height10m	m s-1	Instantaneous [x,y] wind at 10m for coupling	S	GOCART, CMAQ
inst_temp_height_surface	K	Instantaneous surface skin temperature for coupling	S	GOCART, CMAQ
phis	m2 s-2	Surface geopotential	S	JEDI
u_srf	m s-1	Surface u-wind	S	JEDI
v_srf	m s-1	Surface v-wind	S	JEDI
slmsk	Numeric	Land sea mask	S	JEDI

weasd	mm	Liquid water equiv of accumulated snow depth over land and sea ice	S	JEDI
tsea	K	Sea surface temperature from JEDI	S	JEDI
vtype	Numeric	Vegetation type classification	S	JEDI
stype	Numeric	Soil type classification	S	JEDI
vfrac	fraction	Areal fractional cover of green vegetation	S	JEDI
snwdph	mm	Liquid water equivalent snow depth	S	JEDI
f10m	Numeric	Ratio of wind at surface adjacent layer to wind at 10m	S	JEDI
zorl	cm	Surface_roughness_length	S	JEDI
t2m	K	2 meter air temperature	S	JEDI
inst_tracer_mass_frac	fraction	Tracer concentration updated by physics	T	GOCART, CMAQ

All fields exported by FV3 to external components via CMEPS are 2D surface fields.

ATM Export Fields	Field Alias (used by CMEPS)	Units	Field Description
inst_temp_height2m	Sa_t2m	K	Temperature at 2m height
inst_temp_height_lowest	Sa_tbot	K	Temperature at lowest model layer
inst_temp_height_lowest_from_phys	Sa_ta	K	Temperature at lowest model layer from physics
inst_temp_height_surface	Sa_tskn	K	Surface skin temperature for coupling

inst_spec_humid_height2m	Sa_q2m	kg kg-1	Specific humidity height at 2m height
inst_spec_humid_height_lowest	Sa_shum	kg kg-1	Specific humidity at lowest model layer
inst_spec_humid_height_lowest_from_phys	Sa_qa	kg kg-1	Specific humidity at lowest model layer from physics
inst_[zonal,merid]_wind_height_lowest	Sa_[u,v]	m s-1	Surface [u,v] wind
inst_[zonal,merid]_wind_height10m	Sa_[u,v]10m	m s-1	[U,V] at 10m for coupling
inst_pres_height_lowest	Sa_pbot	Pa	Pressure at lowest model layer
inst_pres_height_lowest_from_phys	Sa_prsl	Pa	Pressure at lowest model layer from physics
inst_pres_height_surface	Sa_pslv	Pa	Pressure at land and sea surface
inst_height_lowest	Sa_z	m	Height at lowest model layer based on hydrostatic assumptions
inst_exner_function_height_lowest	Sa_exner	ND	Exner function
surface_friction_velocity	Sa_ustar	m s-1	Surface friction velocity
openwater_frac_in_atm	Sa_ofrac	ND	Ocean fraction used by atmosphere for coupling
inst_[zonal,merid]_moment_flx_atm	Faxa_tau[x,y]	N m-2	Surface [x,y] momentum flux
inst_sensi_heat_flx	Faxa_sen	W m-2	Surface sensible heat flux
inst_laten_heat_flx	Faxa_lat	W m-2	Surface latent heat flux
inst_evap_rate	Faxa_evap	kg m-2 s-1	Latent heat flux converted to evaporation rate
inst_down_lw_flx	Faxa_lwdn	W m-2	Surface downward long wave flux
inst_net_lw_flx	Faxa_lwnet	W m-2	Net downward long wave flux
inst_down_sw_flx	Faxa_swdn	W m-2	Downward shortwave flux

inst_net_sw_flx	Faxa_swnet	W m-2	Net downward shortwave flux
inst_down_sw_[ir,vis]_[dir,dif]_flx	Faxa_sw[n,v][dr,df]	W m-2	Surface [nIR,visible] [beam,diffuse] downward shortwave flux
inst_prec_rate	Faxa_rain	kg m-2 s-1	Rain rate
inst_prec_rate_conv	Faxa_rainc	kg m-2 s-1	Convective rain rate
inst_fprec_rate	Faxa_snow	kg m-2 s-1	Snow rate

C.2 MOM6

Table C.2.1 MOM6 Import Field Information.

See Table C.10.5 States to OCN and Table C.10.6 Fluxes to OCN for details. All fields imported by MOM6 are 2-Dimensional surface fields with the exception of the partitioned Stokes drift, which contains an ungridded dimension equal to the number of Stokes drift partitions.

Ocean Import Field	Units	Field Description	Notes
Sa_pslv	Pa	Pressure of overlying sea ice and atmosphere	
Si_ifrac	ND	Sea ice fraction	Not used in UWM (needed by CFC tracers in MOM6)
So_duu10n	m2 s-2	10m-wind squared	
Sw_lamult	ND	Langmuir multiplier from wave model	Not used in UWM
Sw_pstokes_[x,y]	m s-1	Partitioned Stokes Drift	Imported oriented E-W and rotated in cap to model I-J
Fioi_salt	kg m-2 s-1	Salt flux	
Fioi_melth	W m-2	Heat flux associated with sea ice and snow melt	
Fioi_meltw	kg m-2 s-1	Water flux due to sea ice and snow melt	
Faxa_rain	kg m-2 s-1	Mass flux of liquid water	
Faxa_snow	kg m-2 s-1	Mass flux of frozen water	
Foxx_tau[x,y]	N m-2	Wind stress into ocean	Imported oriented E-W and rotated in cap to

			model I-J
Foxx_sen	W m-2	Sensible heat flux into ocean	
Foxx_evap	kg m-2 s-1	Specific humidity flux	
Foxx_lwnet	W m-2	Net longwave radiation	
Foxx_swnet_[v,i][dr,df]	W m-2	Net visible and near-IR direct and diffuse shortwave radiation	
Foxx_rofl	kg m-2 s-1	Liquid runoff	Not used in UWM; river runoff is provided via MOM's internal data file read
Foxx_rofi	kg m-2 s-1	Frozen runoff	Not used in UWM
Foxx_[hrain,snow,hevap ,hcond,rofl,rofi]	W m-2	Enthalpy of rain, snow, evap, condensation, liquid and frozen runoff	

Table C.2.2 MOM6 Export Field Information

All fields exported by MOM6 are 2-Dimensional surface fields.

Ocean Export Field		Field Description	Notes
So_omask	ND	Ocean land-sea mask	
So_t	K	Ocean surface temperature	
So_s	psu	Ocean surface salinity	
So_[u,v]	m s-1	Ocean surface velocities	Rotated in cap from model I-J to E-W

So_dhd[x,y]	ND	Ocean surface slope	Calculated in cap from model sea-surface height field; rotated in cap from model I-J to E-W
So_bldepth	m	Ocean mixed layer depth	Not used in UWM
Fioo_q	W m-2	Ocean freeze-melt potential	Calculated in cap from model frazil rate and model melt_potential fields

C.3 CICE

Table C.3.1 CICE Import Field Information

See Table C.10.4 States to ICE and Table C.10.8 Fluxes to ICE for details. All field imported by CICE are 2-dimensional surface fields with the exception of the wave elevation spectrum, which contains an ungridded dimension equal to the number of frequency bands used.

CICE6 Import Field	Units	Field Description	Notes
So_dh[dx,dy]	ND	Ocean surface slope	Imported oriented E-W and rotated in cap to model I-J
So_t	K	Ocean surface temperature	
So_s	psu	Ocean surface salinity	
So_[u,v]	m s-1	Ocean surface current	Imported oriented E-W and rotated in cap to model I-J
Sa_z	m	Height of atmosphere lowest level	

Sa_[u,v]	m s-1	Zonal and meridional wind at lowest model layer	Imported oriented E-W and rotated in cap to model I-J
Sa_shum		Specific humidity at lowest model layer	
Sa_tbot	K	Temperature at lowest model layer	
Sa_pbot	Pa	Pressure at lowest model layer	
Sa_ptem	K	Potential temperature at lowest model layer	Not provided by ATM, calculated in cap from Sa_tbot and Sa_pbot
Sa_dens	kg m-3	Air density at lowest model layer	Not provided by ATM, calculated in cap from Sa_z, Sa_shum and Sa_tbot
Sw_elevation_spectrum	m2 s-1	Wave spectral density	Only advertised if *wave_coupling_to_cice = .true.*
Faxa_sw[v,n][dr,dff]	W m-2	Downward shortwave visible and near-IR direct and diffuse flux	
Faxa_lwdn	W m-2	Downward longwave flux	
Faxa_rain	kg m-2 s-1	Liquid precipitation rate	
Faxa_snow	kg m-2 s-1	Frozen precipitation rate	
Fioo_q	W m-2	Feat consumed or released to bring ocean surface layer of to the freezing point of seawater	
Faxa_[bcph,dstwet,dstdry]	kg m-2 s-1	Deposition fluxes of black carbon, wet and dry dust	Not used in UWM

Table C.3.2 CICE Export Field Information

All fields exported by CICE are 2-Dimensional surface fields.

CICE6 Export Field		Field Description	Notes
Si_imask	ND	Land-sea mask	
Si_ifrac	ND	Ice fraction	
Si_t	K	Surface temperature of sea ice	
Si_vice	m	Mean ice volume per unit area	the sum of each category ice thickness*each category ice fraction
Si_vsno	m	Mean snow volume per unit area	the sum of each category snow thickness*each category ice fraction
Si_tref	K	Surface 2m air temperature calculated by ICE surface flux scheme	Not used by UWM
Si_qref	kg kg-1	Surface 2m specific humidity calculated by ICE surface flux scheme	
Si_snowh	m	Snow thickness on sea ice	
Si_u10	m s-1	Surface wind speed calculated by ICE surface flux scheme	
Si_a[vs,ni][dr,df]	ND	Visible and near-ir direct and diffuse sea ice albedo	
Si_thick	m	ice thickness	Advertised by ICE only if *wave_coupling_to_cice = .true.*

Si_floediam	m	ice floe diameter	
Si_ifrac_n	ND	ice fraction for each category thickness	Not used by UWM
Fioi_melth	W m-2	heat flux associated with sea ice and snow melt	
Fioi_swpen	W m-2	shortwave penetration through sea ice	
Fioi_swpen_[v,i][dr,df]	W m-2	shortwave penetration through sea ice for visible and near-IR, direct and diffuse	
Fioi_meltw		water flux associated with sea ice and snow melt	
Fioi_salt	kg m-2 s-1	salt flux	
Fioi_tau[x,y]	N m-2	stress on ocean by ice	Rotated in cap from model I-J orientation to E-W
Faii_tau[x,y]	N m-2	stress on air by ice	
Faii_lat	W m-2	latent heat flux at ice surface	
Faii_sen	W m-2	sensible heat flux at ice surface	
Faii_lwup	W m-2	upward longwave radiation at ice surface	
Faii_evap	kg m-2 s-1	evaporation rate at ice surface	
Faii_swnet	W m-2	net shortwave at ice surface	Not used by UWM
Fioi_[bcpho,bcphi,flxdst]	kg m-2 s-1	black carbon and dust deposition to ocean by ice	

C.4 WW3

Table C.4.1 WW3 Import Field Information

See Table C.10.9 States to WAV for details. All fields imported by WW3 are 2-Dimensional surface fields

WW3 Import Field	Units	Field Description	Notes
Si_ifrac	ND	Sea ice fraction	
So_[u,v]	m s-1	Surface ocean currents	
So_t	K	Sea surface temperature	
Sa_tbot	K	Air temperature at lowest model level	
Sa_[u,v]10m	m s-1	Atmospheric wind at 10m	
Si_thick	m	Sea ice thickness	Only advertised if *wave_coupling_to_cice = .true.*
Si_floediam	m	Sea ice floe diameter	

Table C.4.2 WW3 Export Field Information

All fields exported by WW3 are 2-Dimensional surface fields with the exception of the partitioned Stokes drift and the wave elevation spectrum.

WW3 Export Field		Field Description	Notes

Sw_z0	m	Wave roughness length	
Sw_pstokes_[x,y]	m s-1	Partitioned Stokes Drift	
Sw_lamult	ND	Langmuir multiplier	Not used by UWM
Sw_[u,v]stokes	m s-1	Surface stokes drift currents	
Sw_elevation_spectrum	m2 s-1	Wave elevation spectrum	Only advertised if *wave_coupling_to_cice = .true.*

C.5 GOCART

Table C.5.1 GOCART Import Field Information

All fields imported to the GOCART component are obtained directly from the ATM

GOCART Import Fields	Units	Field Description
inst_pres_interface	Pa	Air pressure at model layer interfaces
inst_pres_levels	Pa	Mean layer pressure
inst_geop_interface	m2 s-2	Geopotential at model layer interfaces
inst_geop_levels	m2 s-2	Geopotential at model layer centers
inst_temp_levels	K	Temperature updated by physics
inst_zonal_wind_levels	m s-1	Zonal wind updated by physics
inst_merid_wind_levels	m s-1	Meridional wind updated by physics
inst_cloud_frac_levels	ND	Instantaneous 3D cloud fraction

GOCART Import Fields	Units	Field Description
inst_pres_interface	Pa	Air pressure at model layer interfaces
inst_ice_nonconv_tendency_levels	kg m-2 s-1	Instantaneous 3D flux of ice from nonconvective precipitation
inst_liq_nonconv_tendency_levels	kg m-2 s-1	Instantaneous 3D flux of liquid water from nonconvective precipitation
inst_tracer_mass_frac	ND	Tracer concentration updated by physics
inst_pbl_height	m	Atmosphere boundary layer thickness
surface_cell_area	m2	Area of the grid cell
inst_convective_rainfall_amount	m	Total convective precipitation
inst_rainfall_amount	m	Total rain and snow precipitation
inst_zonal_wind_height10m	m s-1	Instantaneous x wind at 10m for coupling
inst_merid_wind_height10m	m s-1	Instantaneous y wind at 10m for coupling
inst_friction_velocity	m s-1	Surface friction velocity
inst_land_sea_mask	ND	Land sea mask
inst_temp_height_surface	K	Instantaneous surface skin temperature for coupling
inst_up_sensi_heat_flx	W m-2	Instantaneous upward sensible heat flux for chemistry coupling
inst_surface_roughness	cm	Surface roughness length
inst_surface_soil_wetness	ND	Normalized soil wetness
inst_soil_moisture_content	ND	Volume fraction of condensed water in soil

GOCART Import Fields	Units	Field Description
inst_pres_interface	Pa	Air pressure at model layer interfaces
ice_fraction_in_atm	ND	Ice fraction over open water in atmosphere
lake_fraction	ND	Fraction of horizontal grid area occupied by lake
ocean_fraction	ND	Fraction of horizontal grid area occupied by ocean
surface_snow_area_fraction	ND	Surface snow area fraction

Table C.5.2 GOCART Export Field Information

All fields exported to the GOCART component are sent directly to the ATM

GOCART Export Fields	Units	Field Description
inst_tracer_mass_frac	ND	Instantaneous tracers mass concentration
inst_tracer_up_surface_flx	kg m-2 s-1	Instantaneous tracer surface upward flux
inst_tracer_down_surface_flx	kg m-2 s-1	Instantaneous tracer surface downward flux

C.6 CMAQ

Table C.6.1 CMAQ Import Field Information

All fields imported by the CMAQ component are obtained directly from the ATM.

AQM Import Fields	Units	Field Description
canopy_moisture_storage	kg m-2	Canopy water amount
height	m	Height_above_mean_sea_level
inst_aerodynamic_conductance	m s-1	Momentum exchange coefficient
inst_canopy_resistance	s m-1	Aerodynamic_resistance_in_canopy
inst_cloud_frac_levels	ND	Instantaneous 3D cloud fraction
inst_convective_rainfall_amount	m	Total convective precipitation
inst_friction_velocity	m s-1	Surface friction velocity
inst_geop_interface	m2 s-2	Geopotential at model layer interfaces
inst_geop_levels	m2 s-2	Geopotential at model layer centers
inst_land_sea_mask	ND	Land sea mask
inst_laten_heat_flx	W m-2	Instantaneous sfc latent heat flux
inst_merid_wind_levels	m s-1	Meridional wind updated by physics
inst_net_sw_flx	W m-2	Instantaneous net sfc downward sw flux
inst_pbl_height	m	Atmosphere boundary layer thickness
inst_pres_height_surface	Pa	Instantaneous surface air pressure for coupling

inst_pres_interface	Pa	Air pressure at model layer interfaces
inst_pres_levels	Pa	Mean layer pressure
inst_rainfall_amount	m	Total rain and snow precipitation
inst_sensi_heat_flx	W m-2	Instantaneous sfc sensible heat flux
inst_soil_moisture_content	ND	Volume fraction of condensed water in soil
inst_surface_roughness	cm	Surface roughness length
inst_spec_humid_height2m	kg kg-1	Specific humidity at 2m
inst_temp_height2m	K	Temperature at 2m
inst_temp_height_surface	K	Instantaneous surface skin temperature for coupling
inst_temp_levels	K	Temperature updated by physics
inst_tracer_mass_frac	ND	Tracer concentration updated by physics
inst_zonal_wind_height10m	m s-1	Instantaneous x wind at 10m for coupling
inst_vegetation_area_frac	ND	Areal fractional cover of green vegetation
inst_merid_wind_height10m	m s-1	Instantaneous y wind at 10m for coupling
inst_zonal_wind_levels	m s-1	Zonal wind updated by physics
leaf_area_index	ND	Leaf area index
sea_ice_area_fraction	ND	Sea ice fraction
soil_type	ND	Soil type for lsm
surface_cell_area	m2	Surface cell area
surface_snow_area_fraction	ND	Surface_snow_area_fraction
temperature_of_soil_layer	K	Soil temperature

Table B.6.2 CMAQ Export Field Information

All fields exported by the CMAQ component are sent directly to the ATM.

AQM Export Fields	Units	
inst_tracer_mass_frac	ND	Instantaneous tracers
inst_tracer_diag_aod	ND	Instantaneous aerosol optical depth

C.7 HYCOM

Table C.7.1 HYCOM Import Field Information

See Table C.10.5 States to OCN and Table C.10.6 Fluxes to OCN for details. All fields imported by HYCOM are 2-Dimensional surface fields.

HYCOM Import Field	Units	Field Description	
tau[x,y]10	N m-2	Momentum flux [east,north]ward	inst_[zonal,merid]_moment_flx_atm
swflxd	W m-2	Downward shortwave flux (alias for swflx_net2down)	inst_net_sw_flx
lwflxd	W m-2	Downward longwave flux (alias for lwflx_net2down)	inst_net_lw_flx
prcp		Precipitation (m / s)	inst_prec_rate
mslprs	Pa	Sea level pressure anomaly	inst_pres_height_surface
latflx	W m-2	Latent heat flux	inst_laten_heat_flx

sensflx	W m-2	Sensible heat flux	inst_sensi_heat_flx
gt	K	Surface temperature	Not used in UWM
[u,v]10	m s-1	Wind speed [east,north]ward	
airhum		Specific humidity (%)	
airtmp	K	Air temperature at lowest level	
wndspd10	m s-1	Non-directional wind speed	
ustara10	m s-1	Friction speed	
swflx_net	W m-2	Shortwave flux	
swflx_net2down	W m-2	Downward shortwave flux	
lwflx_net	W m-2	Longwave flux	
lwflx_net2down	W m-2	Downward longwave flux	
sic	ND	Ice concentration	
sitx		Ice x-stress	
sity		Ice y-stress	
siqs		Solar thru grid cell average	
sifh		Freeze, melt, H. Flux	
sifs		Salt flux	
sifw		Water flux	
sit_sfc		Sea ice temperature	
sih		Sea ice thickness	
siu		Sea ice x-velocity	
siv		Sea ice y-velocity	

Table C.7.2 HYCOM Export Field Information

HYCOM Export Field	Units	Field Description	
sst	K	Sea surface temperature	sea_surface_temperature
mask	ND	Land sea mask	ocean_mask
ss[u,v]	m s-1	Average currents over top thkcdw meters, [east,north]ward	Not used in UWM
sss	ppt	Sea surface salinity	
ssh	m	Sea surface height	
ssfi	W m-2	Oceanic heat flux available to sea ice	
mlt	m	Ocean mixed layer thickness	
sbhflx	W m-2	Sensible heat flux	
lthflx	W m-2	Latent heat flux	

C.8 Noah-MP

Table C.8.1 Noah-MP Import Field Information

See Table C.10.3 States to LND and Table C.10.4 Fluxes To LND for details. All fields imported by LND are 2-Dimensional surface fields

LND Import Fields	Units	Field Description	Notes
Sa_z	m	Height of atmosphere lowest level	
Sa_topo	m	instantaneous surface height	
Sa_ta	K	Bottom layer temperature from phys	
Sa_tskn	K	Sea surface skin temperature	
Sa_pslv	Pa	Instantaneous pressure land and sea surface	
Sa_prsl	Pa	Pressure at lowest model layer from phys	
Sa_pbot	Pa	Pressure at lowest model layer	
Sa_shum	kg kg-1	Bottom layer specific humidity	
Sa_qa	kg kg-1	Bottom layer specific humidity from phys	
Sa_[u,v]	m s-1	Bottom layer [zonal][merid] wind	
Sa_exner	ND	Dimensionless exner function at surface adjacent layer	
Sa_ustar	m s-1	Surface friction velocity	
Faxa_swdn	W m-2	Downward SW heat flux	
Faxa_lwdn	W m-2	Downward LW heat flux	
Faxa_swnet	W m-2	Net SW heat flux	

Faxa_rainc	kg m-2 s-1	Convective part of precipitation	
Faxa_rainl	kg m-2 s-1	Large-scale part of precipitation	
Faxa_rain	kg m-2 s-1	Total precipitation rate	
Faxa_snow	kg m-2 s-1	Total snow precipitation rate	
Faxa_snowc	kg m-2 s-1	Convective part of snow precipitation	
Faxa_snowl	kg m-2 s-1	Large scale part of snow precipitation	
Sa_vfrac	ND	Areal fractional cover of green vegetation	For JEDI integration
Sa_zorl	cm	Composite surface roughness	For JEDI integration

Table C.8.2 NoahMP Export Field Information

All fields exported by LND are 2-Dimensional surface fields

LND Export Fields	Units	Field Description	Notes
Sl_lfrin	ND	Land fraction	
Sl_sfrac	ND	Surface snow area fraction over land	
Sl_tref	K	Air temperature at 2m over land	
Sl_qref	kg kg-1	Specific humidity at 2m over land	
Sl_q	kg kg-1	Specific humidity over land	
Sl_cmm	m s-1	Drag coefficient of momentum over land	
Sl_chh	kg m-2 s-1	Drag coefficient of heat and moisture over land	
Sl_zvfun	ND	Function of surface roughness length and green vegetation fraction	

Fall_lat	W m-2	Latent heat flux over land	
Fall_sen	W m-2	Sensible heat flux over land	
Fall_evap	W m-2	Surface upward potential latent heat flux over land	
Fall_gflx	W m-2	Upward heat flux from land	
Fall_roff	kg m-2 s-1	Runoff rate from land	
Fall_soff	kg m-2 s-1	Subsurface runoff rate from land	

C.9 CDEPS

Table C.9.1 CDEPS Export Field Information

CDEPS Export Fields	Data Model Provider	Units	Field Description	AO Flux calculation	OCN HYCOM	OCN MOM6	ICE	WAV	Noah MP	ATM
Sa_[u,v]10m	DATM	m s-1	[Zonal,Meridional] wind at 10m	✓				✓		
Sa_z	DATM	m	Height at lowest model layer	✓			✓		✓	
Sa_pslv	DATM	Pa	Surface pressure	✓	✓	✓			✓	
Sa_tbot	DATM	K	Air temperature at lowest model layer	✓			✓	✓	✓	
Sa_shum	DATM	kg kg-1	Specific humidity at lowest model layer	✓			✓		✓	
Sa_[u,v]	DATM	m s-1	[Zonal,Meridional] wind at lowest model layer	✓			✓		✓	
Sa_q2m	DATM	kg kg-1	Specific humidity at 2m	✓						
Sa_t2m	DATM	K	Air temperature at 2m	✓						
Sa_pbot	DATM	Pa	Pressure at lowest model layer	✓			✓		✓	
Sa_topo	DATM	m	Topographic height						✓	
Sa_tskn	DATM	K	Skin temperature						✓	

Faxa_rain	DATM	kg m-2 s-1	Precipitation rate		✓	✓	✓		✓	
Faxa_snow	DATM	kg m-2 s-1	Snow rate			✓	✓		✓	
Faxa_rainc	DATM	kg m-2 s-1	Convective precipitation rate						✓	
Faxa_rainl	DATM	kg m-2 s-1	Large scale precipitation rate						✓	
Faxa_snowc	DATM	kg m-2 s-1	Convective snowfall rate						✓	
Faxa_snowl	DATM	kg m-2 s-1	Large scale snowfall rate						✓	
Faxa_swnet	DATM	W m-2	Net downward shortwave flux		✓				✓	
Faxa_lwnet	DATM	W m-2	Net downward longwave flux		✓					
Faxa_sen	DATM	W m-2	Sensible heat flux		✓	✓				
Faxa_lat	DATM	W m-2	Latent heat flux		✓					
Faxa_evap	DATM	kg m-2 s-1	Evaporative flux			✓				
Faxa_tau[x,y]	DATM	N m-2	[Zonal,Meridional] momentum flux		✓	✓				
Faxa_swdn	DATM	W m-2	Downward shortwave flux						✓	
Faxa_lwdn	DATM	W m-2	Downward longwave flux			✓	✓		✓	
Faxa_sw[n,v][dr,	DATM	W m-2	Downward [near-IR,visible]			✓	✓		✓	

df]			[direct,diffuse] shortwave flux							
So_t	DOCN	K	Sea surface temperature				✓	✓		✓

C.10 CMEPS Field Exchange Table

CMEPS field exchanges are defined from the perspective of the Mediator component. In CMEPS, these fields reside in the *FBImp(compxxx,compxxx)*, where *(compxxx,compxxx)* is understood to mean *from* component *xxx* on the component *xxx* grid.
Fields exported *to* a component *yyy* reside in the *FBexp(compyyy)*. Named components are, for example, *atm* and *ocn* so that a field imported from the atmosphere on the atmosphere grid would be in the *FBimp(compatm,compatm*). Fields exported to the ocean component would be in *FBexp(compocn)*.

Field names within CMEPS and component caps use a CESM-style naming convention where possible. By this convention, each component is represented by a one letter abbreviation: *a* (atmosphere), *o* (ocean), *i* (sea ice), *l* (land), *w* (wave), *x* (mediator). States of a given component model are represented by *S.* Fluxes are represented by *Fxyz* where *xy* are the components on either side of the flux and z is the component grid where the flux is calculated. The name of the appropriate flux or state is appended using *_fieldname*.

For example, *So_t* is the ocean surface temperature state and *Fioi_taux* is the ice-ocean stress in the zonal direction calculated by the ice model. Fields which are merged in the mediator between component models use *x* to represent the component where the fields are calculated. For example, the field *Foxx_taux* is a merged field between the air-ocean stress and the ice-ocean stress. In this case, the *Foxx* can be read as "flux between the ocean (*o*) and multiple components (*x*), merged by the mediator (*x*).

Note that the active ATM component in UFS (fv3ATM) relies on descriptive field names rather than the CESM-style naming convention described above.

Table C.10.1 States To ATM

To ATM	Field Alias (used by ATM)	From OCN	From ICE	From WAV	From LND
Si_ifrac	sea_ice_fraction		Si_ifrac		

Sl_lfrac	land_fraction				Sl_lfrac
Si_vice	inst_ice_volume		Si_vice		
Si_vsno	inst_snow_volume		Si_vsno		
Si_t	sea_ice_surface_temperature		Si_t		
Si_a[vs,ni][dr,df]	inst_ice_[vis,ir]_[dir,dif]_albedo		Si_a[vs,ni][dr,df]		
So_t	sea_surface_temperature	So_t			
Sw_z0	wave_z0_roughness_length			Sw_z0	
Sl_sfrac	inst_snow_area_fraction_lnd				Sl_sfrac
Sl_tref	inst_temp_height2m_lnd				Sl_tref
Sl_qref	inst_spec_humid_height2m_lnd				Sl_qref
Sl_q	inst_spec_humid_lnd				Sl_q
Sl_cmm	inst_drag_wind_speed_for_momentum				Sl_cmm
Sl_chh	inst_drag_mass_flux_for_heat_and_moisture				Sl_chh
Sl_zvfun	inst_func_of_roughness_length_and_vfrac				Sl_zvfun

Table C.10.2 Fluxes To ATM

To ATM	Field Alias (used by ATM)	From ICE	From LND	From MED AOflux
Faii_tau[x,y]	stress_on_air_ice_[zonal,merid]	Faii_tau[x,y]		
Faii_lat	inst_laten_heat_flx_atm_into_ice	Faii_lat		
Faii_sen	inst_sensi_heat_flx_atm_into_ice	Faii_sen		

Faii_lwup	inst_up_lw_flx_ice	Faii_lwup		
Faii_evap	inst_evap_rate_atm_into_ice	Faii_evap		
Fall_lat	inst_laten_heat_flx_lnd		Fall_lat	
Fall_sen	inst_sensi_heat_flx_lnd		Fall_sen	
Fall_evap	inst_potential_laten_heat_flx_lnd		Fall_evap	
Fall_gflx	inst_upward_heat_flux_lnd		Fall_gflx	
Fall_roff	inst_runoff_rate_lnd		Fall_roff	
Fall_soff	inst_subsurface_runoff_rate_lnd		Fall_soff	
When *use_med_flux = .true.*				
Faox_lat	inst_laten_heat_flx_atm_into_ocn			Faox_lat
Faox_sen	inst_sensi_heat_flx_atm_into_ocn			Faox_sen
Faox_lwup	inst_up_lw_flx_ocn			Faox_lwup
Faox_tau[x,y]	stress_on_air_ocn_[zonal,merid]			Faox_tau[x,y]

Table C.10.3 States To LND

To LND	Fr ATM	Field Alias (used by ATM)	From CDEPS
Sa_z	Sa_z	inst_height_lowest	
Sa_ta	Sa_ta	inst_temp_height_lowest_from_phys	
Sa_pslv	Sa_pslv	inst_pres_height_surface	
Sa_qa	Sa_qa	inst_spec_humid_height_lowest_from_phys	
Sa_[u,v]	Sa_[u,v]	inst_[zonal,merid]_wind_height_lowest	

Sa_prsl	Sa_prsl	inst_pres_height_lowest_from_phys	
Sa_tskn	Sa_tskn	inst_temp_height_surface	
Sa_exner	Sa_exner	inst_exner_function_height_lowest	
Sa_ustar	Sa_ustar	surface_friction_velocity	
Sa_vfrac	Sa_vfrac	vfrac	
Sa_zorl	Sa_zorl	zorl	
When coupling_mode = ufs.nfrac.aoflux			
Sa_z			Sa_z
Sa_topo			Sa_topo
Sa_tbot			Sa_tbot
Sa_pbot			Sa_pbot
Sa_shum			Sa_shum
Sa_[u,v]			Sa_[u,v]
Sa_pslv			Sa_pslv

Table C.10.4 Fluxes To LND

To LND	Fr ATM	Field Alias (used by ATM)	From CDEPS
Faxa_lwdn	Faxa_lwdn	inst_down_lw_flx	
Faxa_swdn	Faxa_swdn	inst_down_sw_flx	
Faxa_swnet	Faxa_swnet	inst_net_sw_flx	
Faxa_rain	Faxa_rain	inst_prec_rate	
Faxa_snow	Faxa_snow	inst_fprec_rate	
Faxa_rainc	Faxa_rainc	inst_prec_rate_conv	
When coupling_mode = ufs.nfrac.aoflux			
Faxa_lwdn			Faxa_lwdn
Faxa_swdn			Faxa_swdn
Faxa_snowc			Faxa_snowc
Faxa_snowl			Faxa_snowl
Faxa_rainc			Faxa_rainc
Faxa_rainl			Faxa_rainl
Faxa_rain			Faxa_rain
Faxa_swnet			Faxa_swnet

Table C.10.5 States to OCN

To OCN	From ATM	Field Alias (used by ATM)	From ICE	From WAV
Sa_pslv	Sa_pslv	inst_pres_height_surface		
Sw_pstokes_[x,y]				Sw_pstokes_[x,y]
When *coupling_mode* = *'hafs'*				
Sa_[u,v]10m	Sa_[u,v]10m	inst_[zonal][merid]_wind_height10m		
Sa_t2m	Sa_t2m	inst_temp_height2m		
Sa_q2m	Sa_q2m	inst_spec_humid_height2m		
Sa_pslv	Sa_pslv	inst_pres_height_surface		
Sa_tskn	Sa_tskn	inst_temp_height_surface		

Table C.10.6 Fluxes to OCN

To OCN	From ATM	Field alias (used by ATM)	From ICE	From MED AOflux
When *coupling_mode* /= *'hafs'*				
Faxa_rain	Faxa_rain	inst_prec_rate		
Faxa_snow	Faxa_snow	inst_fprec_rate		
Foxx_swnet_[i,v][dr,df]	Faxa_sw[n,v][dr,df]	inst_down_sw_[ir,vis]_[dir,dif]_flx	Fioi_swpen_[i,v][dr,df]	
Fioi_meltw			Fioi_meltw	
Fioi_melth			Fioi_melth	

Fioi_salt			Fioi_salt	
When *med_aoflux_to_ocn = .true.*				
Foxx_tau[x,y]			Fioi_tau[x,y]	Faox_[x,y]
Foxx_lwnet	Faxa_lwdn	inst_down_lw_flx		Faox_lwup
Foxx_sen				Faox_sen
Foxx_evap				Faox_evap
When *med_aoflux_to_ocn = .false.*				
Foxx_tau[x,y]	Faxa_tau[x,y]	inst_[zonal,merid]_moment_flx_atm	Fioi_tau[x,y]	
Foxx_lwnet	Faxa_lwnet	inst_net_lw_flx		
Foxx_sen	Faxa_sen	inst_sensi_heat_flx		
Foxx_evap	Faxa_evap	inst_evap_rate		
When *coupling_mode = 'hafs'*				
Faxa_tau[x,y]	Faxa_tau[x,y]	inst_[zonal,merid]_moment_flx_atm		
Faxa_rain	Faxa_rain	inst_prec_rate		
Faxa_swnet	Faxa_swnet	inst_net_sw_flx		
Faxa_lwnet	Faxa_lwnet	inst_net_lw_flx		
Faxa_sen	Faxa_sen	inst_sensi_heat_flx		
Faxa_lat	Faxa_lat	inst_laten_heat_flx		

Table C.10.7 States to ICE

To ICE	From ATM	Field Alias (used by ATM)	From OCN	From WAV	Notes
Sa_[u,v]	Sa_[u,v]	inst_[zonal,merid]_wind_height_lowest			
Sa_z	Sa_z	inst_height_lowest			
Sa_tbot	Sa_tbot	inst_temp_height_lowest			
Sa_pbot	Sa_pbot	inst_pres_height_lowest			
Sa_shum	Sa_shum	inst_spec_humid_height_lowest			
So_t			So_t		
So_s			So_s		
So_[u,v]			So_[u,v]		
So_dh[dx,dy]			So_dh[dx,dy]		
Sw_elevation_spectrum				Sw_elevation_spectrum	advertised by ICE only if *wav_coupling_to_ice* = *.true.*

Table C.10.8 Fluxes to ICE

To ICE	From ATM	Field Alias (used by ATM)	From OCN
Faxa_lwdn	Faxa_lwdn	inst_down_lw_flx	
Faxa_sw[n,v][dr,df]	Faxa_sw[n,v][dr,df]	inst_down_sw_[ir,vis]_[dir,dif]_flx	
Faxa_rain	Faxa_rain	inst_prec_rate	
Faxa_snow	Faxa_snow	inst_fprec_rate	

Fioo_q			Fioo_q

Table C.10.9 States to WAV

To WAV	From ATM	Field Alias (used by ATM)	From OCN	From ICE	Notes
Sa_[u,v]10m	Sa_[u,v]10m	inst_[zonal][merid]_wind_height10m			
Sa_tbot	Sa_tbot	inst_temp_height_lowest			
Si_ifrac				Si_ifrac	
So_[u,v]			So_[u,v]		
So_t			So_t		
Si_floediam				Si_floediam	Advertised by WAV only if *wave_coupling_to_cice = .true.*
Si_thick				Si_thick	

Table C.10.10 States to MED

To MED	From ATM	Field Alias (used by ATM)	From OCN	From ICE	From LND	Notes
Si_imask				Si_imask		
So_omask			So_omask			
Sl_lfrin					Sl_lfrin	
Sa_aofrac	Sa_aofrac	openwater_frac_in_atm				
Sa_[u,v]	Sa_[u,v]	inst_[zonal][merid]_wind_height_lowest				
Sa_z	Sa_z	inst_height_lowest				
Sa_tbot	Sa_tbot	inst_temp_height_lowest				
Sa_pbot	Sa_pbot	inst_pres_height_lowest				
Sa_pslv	Sa_pslv	inst_pres_height_surface				
Sa_shum	Sa_shum	inst_spec_humid_height_lowest				
Sa_ptem						Not Provided by ATM, calculated in CMEPS from Sa_pbot and Sa_tbot
Sa_dens						Not provided by ATM, calculated in CMEPS from Sa_pbot, Sa_shum and Sa_tbot
Sa_[u,v]10m	Sa_[u,v] 10m	inst_[zonal][merid]_wind_height10m				

Sa_t2m	Sa_t2m	inst_temp_height2m				
Sa_q2m	Sa_q2m	inst_spec_humid_height2m				
So_a[vs,ni][dr,df]						Ocean albedo calculated by mediator

Appendix D: Description of the library installation locations on the UWM supported platforms.

Hera:
/scratch1/NCEPDEV/nems/role.epic/spack-stack/spack-stack-1.5.0/envs/unified-env-noavx512/install/modulefiles/Core
/scratch1/NCEPDEV/nems/role.epic/spack-stack/spack-stack-1.5.0/envs/unified-env-noavx512/install/modulefiles/Core /scratch1/NCEPDEV/jcsda/jedipara/spack-stack/modulefiles
Jet:
/mnt/lfs4/HFIP/hfv3gfs/role.epic/spack-stack/spack-stack-1.5.0/envs/unified-env/install/modulefiles/Core
Gaea:
/lustre/f2/dev/wpo/role.epic/contrib/spack-stack/c4/spack-stack-1.5.0/envs/unified-env/install/modulefiles/Core
Gaea C5:
/lustre/f2/dev/wpo/role.epic/contrib/spack-stack/c5/spack-stack-1.5.0/envs/unified-env/install/modulefiles/Core /lustre/f2/dev/wpo/role.epic/contrib/spack-stack/c5/modulefiles
Orion:
/work/noaa/epic/role-epic/spack-stack/orion/spack-stack-1.5.0/envs/unified-env/install/modulefiles/Core
Hercules:

/work/noaa/epic/role-epic/spack-stack/hercules/spack-stack-1.5.0/envs/unified-env/install/modulefiles/Core
/work/noaa/epic/role-epic/spack-stack/hercules/spack-stack-1.5.0/envs/unified-env/install/modulefiles/Core
Derecho:
/glade/work/epicufsrt/contrib/spack-stack/derecho/spack-stack-1.5.0/envs/unified-env/install/modulefiles/Core
/glade/work/epicufsrt/contrib/spack-stack/derecho/spack-stack-1.5.1/envs/unified-env/install/modulefiles/Cor
WCOSS-2 (Acorn only)
module use /lfs/h1/emc/nceplibs/noscrub/hpc-stack/libs/hpc-stack/modulefiles/stack

Reference:

Balaji, V., Anderson, J.L., Held, I.M., Winton, M., Durachta, J.W., Malyshev, S., & Stouffer, R.J. (2006). The Exchange Grid: A mechanism for data exchange between Earth System components on independent grids. Parallel Computational Fluid Dynamics 2005, Elseiver, pp. 179-186. doi: 10.1016/B978-044452206-1/50021-5

Barnes, H. C., Grell, G., Freitas, S., Li, H., Henderson, J., and Sun, S.: Aerosol Impacts for Convective Parameterizations: Recent Changes to Grell–Freitas Convective Parameterization, AMS, https://ams.confex.com/ams/102ANNUAL/meetingapp.cgi/Paper/391989, 2022.

Byun, D. & Schere, K.L. (2006). Review of the governing equations, computational algorithms, and other components of the Model-3 Community Multiscale Air Quality (CMAQ) Modeling System. Appl. Mech. Rev. 59(2): 51-77. doi: 10.1115/1.2128636

Dirmeyer, P. A., A. J. Dolman, and N. Sato,1999: The Global Soil Wetness Project: A pilot project for global land surface modeling and validation. Bull. Amer. Meteor. Soc., 80, 851-878.

Heinzeller, D., Bernardet, L., Firl, G., Zhang, M., Sun, X., and Ek, M. (2023). The Common Community Physics Package (CCPP) Framework v6, Geosci. Model Dev., 16, 2235–2259, https://doi.org/10.5194/gmd-16-2235-2023

Kim, J.,Teng, Y.,Paturi, S.,Vernieres, G., Sluka, T., Zhu, J., Hao, Y., Worthen, D., Li, B., Wang, J., Flampouris, S., Liu, X., Lee, H., Kleist, D., Tallapragada, V.: NOAA 2022: The NOAA-NCEP 40 year Reanalysis with the Next Generation Global Ocean Data Assimilation System (NG-GODAS): 1979 to 2019. Office note (National Centers for Environmental Prediction (U.S.)) ; 509, DOI : https://doi.org/10.25923/26ds-q363.

Lapenta, W., 2015: The Next Generation Global Prediction System. *Third Symp. on Building a Weather-Ready Nation: Enhancing Our Nation's Readiness, Responsiveness, and Resilience to High Impact Weather Events*, Phoenix, AZ, Amer. Meteor. Soc., 1.4. [Available online at https://ams.confex.com/ams/95Annual/webprogram/Paper269068.html.]

Lin S.J., Benson R., Chen, J. H., Chen X., Harris L., Liang Z., Morin M., Putman W., Rees S., and Zhou L. 2016: A brief overview of the FV3 dynamical core, https://www.weather.gov/media/sti/nggps/20160127_FV3_Overview_Description.pdf

Mogensen, K., S. Keeley, and P. Towers (2012a), Coupling of the NEMO and IFS models in a single executable, *Tech. Memo. 673*, Eur. Cent. for Medium-Range Weather Forecasts, Reading, U. K.

Michalakes, J., Govett, M., Benson, R., Black, T., Juang, H. Reinecke, A.,Skamarock, B., Duda, M., Henderson, T., Madden, P., Mozdzynski, G., Vasic, R., NOAA 2015: AVEC Report: NGGPS Level-1 Benchmarks and Software Evaluation. DOI : https://doi.org/10.25923/zq35-b106

Michalakes, John, et al, NOAA 2016. AVEC Report: NGGPS Phase-2 Benchmarks and Software Evaluation, DOI : https://doi.org/10.25923/p0xe-mz37, 2016.

Murray, R. (1996). Explicit Generation of Orthogonal Grids for Ocean Models, J. Comp. Physics, 126, 251-273, doi: https://doi.org/10.1006/jcph.1996.0136

Niu, G.-Y., et al. (2011), The community Noah land surface model with multiparameterization options (Noah-MP): 1. Model description and evaluation with local-scale measurements. J. Geophys. Res., 116, D12109, doi: 10.1029/2010JD015139.

NOAA, 2019: Upgrade NCEP Global Forecast Systems (GFS) to v15.1: Effective June 12, 2019. Service Change Notice 19-40. National Weather Service Headquarters, Silver Spring MD, https://www.weather.gov/media/notification/scn19-40gfs_v15_1.pdf

NOAA, 2020: Announcement of upgrade to the Global Ensemble Forecast System (GEFS), coupled with the Global Wave Ensemble System (GWES) and the NEMS GFS aerosol component (NGAC): Effective September 23, 2020 and request for comments. Service Change

Notice 20-75. National Weather Service Headquarters, Silver Spring MD, https://www.weather.gov/media/notification/SCN_20-75_GEFSv12_Changes.pdf

NOAA, 2021: Upgrade NCEP Global Forecast Systems (GFS) to v16: Effective March 17, 2021. Service Change Notice 21-20, Updated. National Weather Service Headquarters, Silver Spring MD, https://www.weather.gov/media/notification/scn_21-20_gfsv16.0_aaa_update.pdf [Accessed May 20, 2022]

Saha, S., Moorthi, S., Pan, H., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., Liu, H., Stokes, D., Grumbine, R., Gayno, G., Wang, J., Hou, Y., Chuang, H., Juang, H. H., Sela, J., Iredell, M., Treadon, R., Kleist, D., Van Delst, P., Keyser, D., Derber, J., Ek, M., Meng, J., Wei, H., Yang, R., Lord, S., van den Dool, H., Kumar, A., Wang, W., Long, C., Chelliah, M., Xue, Y., Huang, B., Schemm, J., Ebisuzaki, W., Lin, R., Xie, P., Chen, M., Zhou, S., Higgins, W., Zou, C., Liu, Q., Chen, Y., Han, Y., Cucurull, L., Reynolds, R. W., Rutledge, G., & Goldberg, M. (2010). The NCEP Climate Forecast System Reanalysis, *Bulletin of the American Meteorological Society*, *91*(8), 1015-1058.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y., Chuang, H., Iredell, M., Ek, M., Meng, J., Yang, R., Mendez, M. P., van den Dool, H., Zhang, Q., Wang, W., Chen, M., & Becker, E. (2014). The NCEP Climate Forecast System Version 2, *Journal of Climate*, *27*(6), 2185-2208.

Stefanova, L,., Meixner, J., Wang, J., Ray, S., Mehra, A.; Barlage, M., Bengtsson, L., Bhattacharjee, P., Bleck, R., Chawla, A., Green, B., Han, J., Li, W., Li, X., Montuoro, R., Moorthi, S., Stan, C., Sun, S., Worthen, D., Yang, F., Zheng, W. NOAA 2022: Description and Results from UFS Coupled Prototypes for Future Global, Ensemble and Seasonal Forecasts at NCEP. Series : Office note (National Centers for Environmental Prediction (U.S.)) ; 510. DOI : https://doi.org/10.25923/knxm-kz26

Theurich, G., C. DeLuca, T. Campbell, F. Liu, K. Saint, M. Vertenstein, J. Chen, R. Oehmke, J. Doyle, T. Whitcomb, A. Wallcraft, M. Iredell, T. Black, A.M. da Silva, T. Clune, R. Ferraro, P. Li, M. Kelley, I. Aleinov, V. Balaji, N. Zadeh, R. Jacob, B. Kirtman, F. Giraldo, D. McCarren, S. Sandgathe, S. Peckham, and R. Dunlap, IV, 2016: The Earth System Prediction Suite: Toward a coordinated U.S. modeling capability. Bull. Amer. Meteorol. Soc., **97**, no. 7, 1229-1247, doi:10.1175/BAMS-D-14-00164.1

United States Environmental Protection Agency. (2018). CMAQ (Version 5.2.1) [Software]. Available from https://doi.org/10.5281/zenodo.1212601

Zhou, L., Lin, S.-J., Chen, J.-H., Harris, L., Chen, X., & Rees, S. L. (2019). Toward convective-scale prediction within the next generation global prediction system. Bulletin of the American Meteorological Society, 100(7), 1225–1243. https://doi.org/10.1175/BAMS-D-17-0246.1