

A development review 2017-2022

Arun Chawla Unifying Innovations in Forecasting workshop July 18 - July 22 2022



Contributing Authors

- Rusty Benson
- Cristiana Stan
- Jeffrey Whitaker
- Jun Wang
- Ligia Bernardet
- Philip Pegion
- Rahul Mahajan

- Rocky Dunlap
- Dom Heinzeller
- Raffaele Montuoro
- Barry Baker
- Bin Liu
- Mark Potts
- Stylianos Flampouris





What is our Aim?

- Reduce the number of independent modeling systems across our community
- Move to unified state-of-art coupled modeling system across our enterprise
- Share common infrastructure across organizations
- Provide easy access and entry point to end-to-end system
 - encourage community engagement in model development with a long-term goal to accelerate innovation
- Work together towards a goal of improving our weather forecasting skills





About UFS

- The Unified Forecast System (UFS) is a community-based, coupled, comprehensive Earth modeling system targeted towards improvements in NWS operational modeling suites and developing better understanding of earth system processes
- UFS is hosted on GitHub and follows an open development plan
- UFS can be configured into global or regional applications in coupled or uncoupled configurations (medium-range weather, short-range weather, hurricane, etc.)
- NOAA & NCAR have a MOA to advance user-friendly community infrastructure used in UFS, NCAR, and other modeling systems





ESMF/NUOPC



ESMF/NUOPC is the model coupling infrastructure used in the UFS.

The coupling infrastructure is the "glue" that allows the UFS to be built from a set of heterogeneous model components.

Each component has a **NUOPC** "**cap**" - a small software layer that standardizes the interfaces to the model component.

ESMF provides core functions including grid remapping, process/thread management, asynchronous I/O, and is the basis for the **CMEPS** Community Mediator and **CDEPS** "data" components.







Common Community Physics Package



 An open-source and publicly supported library of parameterizations with associated framework for use in weather models

 Facilitates research and development through interoperability, clear interfaces, and support for hierarchical development





UFS Infrastructure Working Group

- An Infrastructure Working Group was set up to identify how a unified infrastructure could be developed
- The aim was to identify a few guiding principles around which a community could develop and evolve
- Broadly proposed a 3 pronged approach
 - Develop community repositories
 - Create Data Portal to serve the community
 - Develop a unified workflow strategy
- An "umbrella repository" structure was emphasized to take advantage of the decentralized nature of existing community repositories





A 3-pronged approach for UFS applications





Umbrella Repository Strategy

UFS Weather Forecast app: new physics development







Code Management and support for UFS

- Multi year project funded by EPIC Program Office to
 - Setup open community repository infrastructures for UFS applications and models
 - Provide code management and testing support
 - Develop basic infrastructure to support UFS applications
- Leveraged in kind resources from multiple organizations (EMC, GFDL, NCAR, NASA, PSL, DOE)
- Precursor to EPIC coming online
- Aim to create a common collaborative environment where multiple organizations could develop the UFS applications





UFS on GitHub

https://github.com/ufs-community/ufs/wiki

Search or jump to 🕢 Pull requests Issues Marketplace Explore		♣ +• ▦•
☐ ufs-community / ufs	6 ★ Star 0 % Fork 0	
<> Code ① Issues 0 ① Pull requests 0 ① Actions 凹 Projects 0 💷 Wiki ① Security 👍 Insight	s 🔅 Settings	
Home ceceliadid edited this page 21 days ago - 25 revisions	Edit New Page	
Welcome to the UFS Community!	▶ Pages 🕡	
This is the central repository for software that is part of the Unified Forecast System (UFS). UFS is a community-based, coupled, comprehensive Earth modeling system. The UFS numerical applications span local to global domains and predictive time scales from sub-hourly analyses to seasonal predictions. UFS is designed to support the Weather Enterprise and to be the source system for NOAA's operational numerical weather prediction applications. You can read more about the Unified Forecast System on the UFS Portal.	Home Supported Platforms and Compilers Development Processes	
Applications and Components	 Developing with Gitflow 	
UFS applications are configurations that support particular forecast requirements. Each application usually combines a numerical model, data assimilation, post-processing, workflow, and other elements. Since these end-to-end applications must be structured in a very specific way for operational forecasts, it may be easier for others to use a simpler configuration - for example, one without a full data assimilation element application of the sample element of a papel structure of the sample component.	Creating a GitHub Account for Development Work Conventions and Policies	

UFS is made up of multiple application repositories

Each repository is organized as a hierarchical "umbrella" repository that connects to different component repositories on GitHub





UFS Applications and Models

- A numerical model is an umbrella repository for earth system models that are connected together to provide forecasts
- Applications are UFS configurations that support particular forecast requirements
- Each application can combine a numerical model, data assimilation, pre- and post-processing, a workflow, and other elements



LINIELED EORECAST SYSTE



UFS Weather Model (UFS WM)

- A numerical model repository for the UFS coupled model that connects the atmospheric dycore (FV3), atmospheric physics (CCPP), a mediator (CMEPS), ocean (MOM6/HYCOM), waves (WW3), ice (CICE6), aerosol (GOCART), chemistry (CMAQ) and data model (CDEPS)
- Serves the following applications -- Medium range weather, S2S, Short range weather, Hurricane, Air Quality
- Can run multiple configurations -- Atmosphere-only mode, fully coupled, ocean-only mode, regional, hurricane mode (currently in development)
- Serves NWS operations
 - Transitioned to operations for GFS v15, GFS v16, GEFS v12
 - Development model for GFS v17, HAFS v1, RRFS v1 and seasonal SFS v1
- Common model core has been critical for developing a unified framework



UFS WM Coupling Capability

- The atmosphere (FV3ATM) NUOPC cap was refactored and updated to leverage the latest NUOPC features and to remove the legacy internal ESMF clock
- New run phases were added in FV3ATM to couple with aerosol components, write grid component calls were updated to allow coupling of the UFS WM within the JEDI framework
- New coupling fields were added to enable coupling FV3ATM with ocean, sea ice, wave, aerosols and JEDI
- Unified interface was developed to map the FV3ATM internal fields with different data structure in FV3ATM import and export states



LINIEIED EOREC



UFS WM components

- Many of the components that make up the UFS weather model are community developed and/or utilized within multiple modeling systems
 - WW3 wave model (UFS, GFDL, CESM, US Navy, USACE, UKMO, Environment Canada, etc)
 - MOM6 ocean model (UFS, GFDL, US Navy, CESM, etc)
 - HYCOM ocean model (UFS, US Navy)
 - FV3 dycore atmospheric dycore (UFS, GFDL, NASA, etc)
 - CCPP common community physics package (UFS, CESM, US Navy)
 - UPP unified post processing (UFS, NCAR)
 - CICE6 ice model (UFS, NASA, CESM, US Navy, DOE)
 - GOCART aerosol model (UFS, NASA)
 - CMAQ chemistry model (UFS, USEPA)
 - CMEPS and CDEPS community mediator and data models respectively (UFS, CESM)
- Code management of component repositories done by the respective communities, with care taken to ensure updates for UFS do not break other modeling systems served by the components and vice versa
- Continuous integration testing is critical to ensure development remains agile





UFS WM Build System

- Ufs-weather-model uses a configuration-based build system to select subcomponents and build executables.
- The valid configurations are: ATM, ATMAERO, ATMW, ATMAQ, S2S, S2SA, S2SW, S2SWA, HAFS, HAFSW, HAFS-ALL, NG-GODAS, UFSAQM. Below are the major configurations.

	FV3atm	MOM6	НҮСОМ	CICE6	WW3	AERO	CMAQ	CDEPS	CMEPS	Stochy Phys	FMS
АТМ	V									V	V
S2SWA	V	V		V	V	V			V	V	V
HAFS	V		V		V			V	V	V	V
NG-GO DAS		\checkmark		V				V	V	V	V
AQM	V						V		V	V	VI

INIFIED EORECAS



UFS WM Build System (cont.)

The UFS WM uses CMake build system.

- When new model components are integrated, the CMake build capability is required.
 CMake capability was implemented through collaboration and added in some authoritative component repositories.
- Several physics configurations (CCPP physics suite files) can be built within the same executable to simplify direct comparisons. The same executable can be used to run both coupled and atmosphere-only simulations.
- Several executables can be built simultaneously with different configurations of components for regression testing.
- FMS infrastructure was moved out from the model code to become a prebuilt library. This saves time to build the executable. The average saving time is ~10%.
- See UFS weather model build talk by Brian Curtis on July 21 (10 am 12 pm session) for more details





CI/CD and regression test framework updates

- Automatic regression test capability was developed through GitHub API to enable CI/CD on most tier-1 platforms.
- Github actions now check component status. Tests were extended to regional and coupled applications.
- Operational Requirement Test (ORT) framework was developed. All submitted new features are required to run ORT for automated decomposition, threading, restart and debug testing. [See poster from Denise Worthen on Wed July 20th 4 6 pm for more details]
- Regression tests cover a range of configurations:
 - Global atmosphere-only for resolutions from 200km-13km and different physics features
 - Regional with latest RRFS configuration
 - HAFS including atm standalone, nested and coupled tests
 - Coupled with the latest prototype configuration
 - NG-GODAS with different data atmosphere forcing
 - Weekly for resources consuming (significant compute and time) tests



All developer interactions managed through GitHub Issues

Filte	rs → Q is:issue is:open 🛇 Labels 18 🗘	Milestones 0	New issue
	① 47 Open ✓ 129 Closed Author → Label → Projects → Milest	ones - Assignee -	Sort -
	Having more flexible Cmake build system enhancement #416 opened 2 days ago by uturuncoglu		Ç 9
	① Update modulefiles directory to use pio/2.5.2 in ufs-weather-model enhancement #415 opened 2 days ago by binli2337	Ť	
	① diag_table changes model results in coupled run bug #413 opened 4 days ago by junwang-noaa		\Box
	① specifying diag_table for the coupled model benchmark and control tests enhancement #412 opened 4 days ago by DeniseWorthen		
	① Cannot turn off OpenMP if hpc-stack (NCEPLIBS) is built with OpenMP bug #410 opened 6 days ago by climbfuji	1	
	Regional restart run is creating different results. Hug #408 opened 9 days ago by RatkoVasic-NOAA		
	mushy cice physics enhancement #402 opened 14 days ago by rgrumbine		
	Revising oro data with full lakes only enhancement #388 opened 23 days ago by shansun6		Ç 2
	ocean stochastic physics enhancement #371 opened on Jan 14 by pjpegion	§7 1	
	① Fully coupled test case (cpld_bmarkfrac_wave_v16) not thread-safe [bug #367 opened on Jan 13 by JessicaMeixner-NOAA		ς:
	① the coupled model requires FV3 and CMEPS updates for frac grid normalization enhancement #366 opened on Jan 13 by DeniseWorthen	۹	
	Add regional DEBUG regression test enhancement #361 opened on Jan 11 by RatkoVasic-NOAA	171 🍩	
	Incorporation the changes in fv3atm and ccpp/physics for the NSST model in the coupled system #350 opened on Dec 29, 2020 by XuLi-NOAA	Ů	Ç
	The new CCCPP suite FV3_GFS_v16b_ugwpv1 and regression tests for GFSv16b-127L/C96 configuration enhancement #347 opened on Dec 27, 2020 by ValeryYudin-NOAA	SD 1	
	add new suite file for regional fv3 test enhancement #344 opened on Dec 23, 2020 by junwang-noaa	9	Ç
-			



Active Wiki for Developers

н	nme

Jun Wang edited this page on Oct 22, 2020 · 100 revisions

Welcome to the UFS Weather Model wiki!

The ufs-weather-model repository contains the model code and external links needed to build the Unified Forecast System (UFS) atmosphere model and associated components, including the WaveWatch III model. This weather model is used in several of the UFS applications, including the medium-range weather application and the short-range weather application.

This wiki includes a Getting Started section that leads you through a quick example. It also includes a Graduate Student Test based on that example. The Graduate Student Test is a way for students, postdocs and others to provide feedback on the usability of the code by running through a series of experiments and then filling out a questionnaire about the experience.

Documentation and User Support

The UFS Weather Model User's Guide has more comprehensive information than this wiki, including links to scientific descriptions of the code and more thorough technical instructions.

Depending on what you are doing, you may need different versions of the User's Guide:

Version	Description
Tag ufs- v1.1.0	Documentation for the most recent release and the version currently used in the Getting Started section and Graduate Student Test.
Latest	Documentation for the head of the development. This may have gaps and errors.

You can also get expert help through a user support forum set up specifically for the UFS Weather Model.

UFS Weather Model Support Forum

The ufs-weather-model Hierarchical Repository Structure

The ufs-weather-model repository supports several UFS applications. It contains atmosphere and wave components and some infrastructure components. Each of these components has its own repository. All the repositories are currently located in GitHub with public access to the broad community. Below is the ufs-weather-model hierarchical repository structure:

Hierarchy of repositories	Authoritative repo URL
ufs-weather-model	https://github.com/ufs-community/ufs-weather-model
I FMS	https://github.com/NOAA-GFDL/FMS
I NEMS	https://github.com/NOAA-EMC/NEMS
└── NCEPLIBS-pyprodutil	https://github.com/NOAA-EMC/NCEPLIBS-pyprodutil

Pages 22

Home

Edit New Page

Creating a GitHub Account for Development Work

Getting Started

1. Download the Code 2. Compile the Model 3. Download a Test Case 4. Run the Test Case 5. Check Your Results

Developer Information

- Commit Queue
- Templates for issues and pull requests available
- Making code changes in the UFS weather model using the GitHub repository
- Building model
- Running regression tests
- Running unit tests
- Regression Test Policy for Weather Model Platforms and Compilers
- · Porting to a new machine

UFS

Applications (include workflow)

- UFS Medium Range Weather Application
- UFS Short Range Weather Application

Models

- UFS Weather Model
- UFS Hurricane Analysis and Forecast System Model

Utilities



UFS Code Repository and Documentation

• The ufs-weather-model repository can be found at: https://github.com/ufs-community/ufs-weather-model

• The latest configuration documentation for UFS Weather Model can be found at:

https://ufs-weather-model.readthedocs.io/en/latest/





Examples of Infrastructure Development Integrating GOCART

• 2.50 • 1.80 • 1.40 Hot Log

0.60 0.40

- GOCART aerosol model integrated to UFS through collaboration with NASA developers
- New run phases added in FV3ATM cap to couple GOCART at each forecast time step

- FV3ATM is currently one way coupled to GOCART (radiation feedback planned for year 3 of UFS R2O project)
- Coupling of aerosols still limited to atmosphere via FV3ATM, though the atmosphere component itself is fully coupled to other earth system components





Examples of Infrastructure Development Multiple/Telescoping and Storm-Following Moving Nests

- Currently, UFS WM supports not only single domain global and regional configurations, but also multiple static and telescopic global/regional nesting configurations (mainly developed by GFDL and EMC).
- Recently, the storm-following moving nesting capability was also developed with close collaboration among AOML, EMC, GFDL, NCAR/ESMF collaborators, which is one of the major milestone for the UFS-HAFS application development and HAFSv1 Initial Operational Capability (IOC) implementation and transition.



Global parent with 4 static and telescopic nests



Regional parent with a storm-following moving nest



Examples of Infrastructure Development Asynchronous Multiple Grid Output

- Multiple grids now available for output in FV3ATM write-grid component
 - native cubed-sphere
 - standardized lat-lon resolutions of separate regions
- ESMF library updated to support multiple grids owned by different sets of forecast tasks and set up communication for the corresponding output grids on the write-grid components
- Regridding is conducted between forecast grids and corresponding write output grids
- Further development work to support moving nest outputs and improve optimization of the computation





Examples of Infrastructure Development Inline Post

- UPP (Unified Post Processor) is a software package developed to create useful atmospheric products from raw model output
- Inline post-processing capability added to create UPP output from memory
- Inline post capability extended to regional applications including RRFS and HAFS
- Special treatment to mask missing values on areas outside the computational domain
- Results from in-line and off-line post are not bitwise identical, but consistent with correlations over 0.99
- Inline post capability critical in I/O bound computations (large domains, cloud etc)

30-0 mb SPFH from LAM







Unified Library Stack

- UFS applications require a host of libraries for all the individual components (at last count over 30 and growing)
- For a system that can be ported on multiple platforms it is critical that there is a reliable and reasonably painless way to build all the necessary libraries and their dependencies
- spack-stack is a collaborative effort between NOAA-EMC and JCSDA to compile a consistent software stack for all UFS and JEDI applications across platforms: user workstations/laptops, HPCs, cloud platforms
- Builds on the open-source spack software manager developed at LLNL
- Current release: spack-stack 1.0.1 on Orion, Discover, AWS (Red Hat 8, Ubuntu 20) continuously expanded. Target: quarterly releases
- For more information, see Dom Heinzeller's talk on Wed 9.35 am ET





Public Releases

- Aim is to introduce the community to UFS and support it in developing new capabilities for the UFS.
- Comes with Graduate Student Tests (GST) to measure how easily a student can
 - download, build, change and test code for correctness
 - access documentation and user support
- Tested on multiple platforms (different HPC platforms, laptops, containers etc)
- Include extensive documentation
- Releases have transitioned from multi-agency activities to EPIC





Application Public Releases

- MRW Application release 1.0 3/11/2020
 - First UFS release for an atmosphere only global configuration
 - First attempt at a unified library build
- MRW Application release 1.1 10/8/2020
 - Minor release updating to Python 3 configuration in CCPP
 - Bug fixes
- SRW Application release 1.0 09/16/2021
- First release of Limited Area Modeling (LAM) capability with the UFS weather model
- SRW Application release 2.0 6/23/2022
 - Second release of LAM capability
 - First release under EPIC





Support Forums



- Publicly viewable support forum
- Registered users may post new topics and/or responses to existing topics
- Forums available for:
 - Build Dependencies
 - Initialization
 - Post-processing
 - UFS Weather Model
 - Medium-Range Weather Application
 - Short-Range Weather/Convection Allowing Application
 - General Discussion





Where we stand today

- A common ufs-community organization with multiple repositories is available on GitHub
- Repositories follow a community development paradigm where anyone who wishes to contribute can do so
- The UFS modeling system shares components with other modeling systems, and leverages developments where possible
- Multiple agencies and organizations are collaborating through these repositories
- EPIC charged with take the UFS community infrastructure to the next level





Challenges

- The UFS is made up of multiple community components, each having a large and diverse community. This necessitates a hierarchy of code managers that
 - have excellent communication skills
 - are very disciplined
 - have detailed procedures to ensure that agile development is not compromised
 - have excellent debugging and engineering skills
 - \circ \quad have an open mind to see multiple points of view
- As the community increases we need to have strict discipline in making contributions as well as excellent testing processes to maintain the quality of our code base. Code managers all across the UFS suite need to both develop standards as well as be empowered to diligently enforce them
- Each of the components that make up the UFS have developed independently. As a result, there are many duplicative libraries and multiple solutions for handling challenges like IO costs





Challenges (contd)

- Challenges laid out in the previous slide is only for the forward model. Data Assimilation capabilities (either weak or strongly coupled) add their own layers of complexity
 - New issues arise when Data Assimilation introduced to forward model capabilities
 - Science of strongly coupled Data Assimilation has many challenges
 - Significant investment in infrastructure needed to make data easily available to the community
- A unified workflow framework still eludes us
 - Each application is exploring its own workflow framework
 - A unified approach that can work across multiple applications and across varying HPC architectures requires sustained resources over a reasonable period of time







