

# The Coastal Modeling Framework

Code development and infrastructure

*Ufuk Turuncoglu & UFS Coastal Team\**  
NCAR/CGD/CSEG/ESMF



May 8, 2025

# UFS Coastal

It is a fork of UFS WM

There is no any formal schedule but synced with UFS WM head of develop every couple of months

ufs-weather-model Public  
forked from ufs-community/ufs-weather-model

feature/coastal\_app 9 Branches 1 Tag  
This branch is 359 commits ahead of, 11 commits behind ufs-community/ufs-weather-model:develop

uturuncoglu point forks ✓ 1466652 · last week 1,382 Commits

Branch	Description	Last Commit
.cicd	Merge remote-tracking branch 'origin/feature/coastal_ap...	3 months ago
.github	sync model	2 months ago
ADCIRC-interface	modified ADCIRC-interface/CMakeLists.txt to allow suppo...	last year
AQM @ 0a8a9a4	sync components	2 months ago
CDEPS-interface	more fix for issues	2 months ago
CICE-interface	sync components	4 months ago
CMEPS-interface	sync components	2 months ago
CMakeModules @ 9b7652c	more fix for issues	2 months ago
FV3 @ e5b040c	sync components	2 months ago
FVCOM-interface	fix fvcom build	2 months ago
GOCART @ 36e2ab4	sync components	2 months ago
GeoGate @ ec421b7	sync components	2 months ago
HYCOM-interface	Support IntelLLVM compiler (ufs-community#2224)	last year

About  
This repo is forked from ufs-weather-model, and contains the model code and external links needed to build the UFS coastal model executable and model components, including the ROMS, FVCOM, ADCIRC and SCHISM plus WaveWatch III model components.

github.com/oceanmodeling/ufs-coast...

Readme  
View license  
Activity  
Custom properties  
9 stars  
4 watching  
6 forks  
Repository

Releases  
1 tags  
Create a new release

Packages  
No packages published  
Publish your first package

It includes additional components:

ADCIRC  
FVCOM  
PaHM  
ROMS  
SCHISM

GeoGate

Slightly modified components:

CMEPS  
CDEPS  
CICE  
WW3



# Forks for sub-components (shared with UFS WM)

- CMEPS
  - It includes new coupling mode called as *coastal*. There is a plan to merge it with *ufs* coupling mode once UFS Coastal development is mature enough.
- CDEPS
  - One additional data mode called as *atmmesh* and also minor modification in ERA5 data mode (missing variables are added for CICE coupling)
- WW3
  - It has additional support to build WW3 tools and also running standalone WW3 configurations under UFS Coastal
- CICE6
  - Modification for updating open boundaries for regional coupling (PR [#174](#), CICE fork PR [#2](#))

# UFS Coastal RTs

- UFS Coastal inherits RT framework from UFS WM
- Includes additional UFS Coastal specific RTs to test the capability
  - There is at least one DATM coupled test for each new ocean model component
  - + coupled configurations like
    - DATM+SCHISM-WW3
    - DATM-DOCN-CICE and DATM-SCHISM-CICE (will be available soon)
  - The existing RTs are ported from old *CoastalApp*
    - BUT, we are in a transition to move more realistic configurations
      - *Duck NC* case is used for SCHISM coupled configurations (PR [#179](#))
  - Runs in every PR to the UFS Coastal

# Main philosophy to maintain UFS Coastal

- Since nothing is pushed to UFS WM level -> **RULE #1: DO NOT DIVERGE**
  - **Keep changes very minimal** in the UFS Coastal (or WM) infrastructure
  - **RULE #2: CAREFUL TESTING** if we need to change any component shared with UFW WM - tests need to pass
    - As an example, we tested modifications for new regional CICE configuration with global fully coupled S2S configuration
    - **RULE #3: ISOLATE CHANGES** as much as possible and create PR/s in the authoritative repositories if it is possible -> sync fork
  - **RULE #4: REGULARLY SYNC WITH UFS WM** authoritative repository
    - It involves fixing issues that are raised in the merge process (i.e. *use\_cmeps* flag added to WW3 coupling interface was creating issue in UFS Coastal)

# Challenges - Model Level

- **Challenge:** *UFS WM is evolving too fast. Actually it is a good thing for UFS WM, BUT*
  - Nothing is exposed to UFS WM in terms of UFS Coastal development
  - It is hard to test UFS Coastal with every PR that goes into UFS WM. There is not enough bandwidth (human+compute) in UFS Coastal team
- **Possible solutions?**
  - Automated testing capability (GitHub CI/CD, Jenkins)
  - More frequent and regular sync with UFS WM
    - Minimize the issues -> requires more work to fix conflicts/issues
  - Pushing existing development in UFS Coastal to UFS WM
    - This requires commitment from component developers (hard to achieve)
    - + requires decision in management level

# Challenges - Component Level

- **Challenge:** *Lack of testing in the component side*
  - The development in the component side is ongoing independently without enough testing with UFS Coastal (i.e. changes that is not backward compatible)
  - The component developers are not following best practices in terms of code development (“cap” and model) -> merge code without testing
  - It leads compile time and/or runtime issues in the UFS Coastal
  - Component developers does not have enough bandwidth to maintain the code
- **Possible solutions?**
  - Providing an easy to use work/testing environment (possible integrated with CI/CD) to component developers to test their model easily with UFS Coastal
    - Able to update component -> run -> report about results

# Challenges - External users/collaborators

- **Challenge:** *Request for help to bring new model configurations*
  - Might require experience in multiple model components (incl. coupling layer) and model infrastructure (RTs system etc.)
  - Requires vast amount of time to find the issues and fix them
  - Lack of experience in managing/working on multi-component GitHub repositories
- **Possible solutions?**
  - Tightly engaging with the community by providing training about overall infrastructure (incl. git and RT framework), components and their coupling layers
  - Sharing training videos and make those accessible by the community
  - Existing tests could be used as a reference -> need more diverse and realistic test cases defined under UFS Coastal

# Challenges - Application Level

- **Challenge:** *keep components of the application synced + supporting more configs.*
  - It uses UWTools as a workflow manager
  - Issues/changes in the model and/or workflow manager might break the application
  - Lack of regular testing in the application level -> porting is also challenging
- **Possible solutions?**
  - Need to develop robust automated testing capability (GitHub CI/CD, Jenkins) in the application layer too
  - Encourage users to test and use application -> not the model
  - Shared input data and configurations through the cloud -> [AWS](#)
  - Document workflow and test cases in a clear and understandable way

# Questions?

Ufuk Turuncoglu

[turuncu@ucar.edu](mailto:turuncu@ucar.edu)

You could also create discussion in our GitHub repository for further question/s and collaboration:

<https://github.com/oceanmodeling/ufs-weather-model/discussions>

