

# UFS Weather Model Test Cases

By: Gillian Petro



# The Unified Forecast System (UFS)

- Comprehensive Earth modeling system
  - Community-based
  - Coupled
- Source for NOAA's operational models
- Used for research and development within the weather modeling community
- Contributions encouraged!



# The UFS Weather Model (WM)

- Model for short- and medium-range research and operational forecasts
- Based on [Finite-Volume Cubed-Sphere \(FV3\) dynamical core](#)
- Includes:
  - Ocean, wave, land, aerosol, sea ice components
  - Mediators/couplers
  - The [Common-Community Physics Package \(CCPP\)](#)
  - [Stochastic physics](#)
  - Unified Post Processor – inline post-processing component
  - Regression testing (RT) framework

# Hierarchical Testing Framework

- Supports [Hierarchical System Development \(HSD\)](#)
- Multiple entry points to development
  - Atmospheric physics
  - Ocean and ice dynamics
  - Data assimilation for land models
- Multiple levels of complexity
  - Unit, integration, and regression testing
  - Science & software testing
- Operationally relevant and idealized configurations

# Resources

- [UFS WM Code](#)
- [UFS WM User's Guide](#)
- [GitHub Discussions \(Q&A\)](#)
- [GitHub Issues](#)
- [FV3 Technical Documentation](#)
- [FV3 Scientific Description](#)
- [EPIC Weather Model page](#)
- [GitHub Wiki](#)
- [WM RT Data Bucket](#)
- [HTF Data Bucket](#)

The image shows two overlapping screenshots. The top screenshot is the UFS Weather Model User's Guide website. It features a blue header with the title 'UFS Weather Model Users Guide' and a search bar. A navigation menu on the left lists sections from 1 to 12. The main content area displays a 'Welcome to the UFS Weather Model User's Guide' message with a table of contents. A large banner image with the text 'UFS Weather Model' is visible. Below the banner, there are links for 'Get Started', 'Documentation & User Support', 'Developer Support', and 'Releases'. A sidebar on the right contains sections for 'Get Started with the UFS WM', 'Resources', and 'For Developers'.

The bottom screenshot is a GitHub repository page for 'ufs-community / ufs-weather-model'. It shows the repository's home page with a search bar and navigation tabs for Code, Issues (149), Pull requests (39), Discussions, Actions, Projects (8), Wiki, and Security (2). The main content area displays a 'Welcome to the UFS Weather Model wiki!' message, a 'Home' section with a 'New page' button, and a list of 'Getting Started' steps: 1. Download Prerequisite Libraries, 2. Download the Code, 3. Download Data, and 4. Run the Test Case. The repository description states that it contains the model code and external links for building the Unified Forecast System (UFS) atmospheric model (fv3atm) and associated components.



# July 2020 CAPE Case

- **The Problem:** Biases in the GFS v16 cause [low CAPE predictions in the summer](#).
- **Configuration:**
  - Atmosphere-only forecast
  - 24-hour forecast
  - C48 (~200km) resolution
  - 127 vertical levels
  - Start date: 2020-07-23 at 0z
  - FV3\_GFS\_v16 physics
  - Default values from the WM's [default\\_vars.sh](#) export\_fv3\_v16 function
- **Goals:**
  - Get acquainted with the UFS WM
  - **Experiment!** Use different initial conditions, a coupled land surface model (LSM), or other warm-season cases



# Baroclinic Wave Case

- Case outlined in [Jablonowski and Williamson \(2006\)](#).
- **Purpose:** Evaluate accuracy of atmospheric models in simulating a baroclinic wave, which forms in the Northern Hemisphere and influences weather patterns.
- **Configuration:**
  - Atmosphere-only
  - [Dycore](#)-only (no physics)
  - C192 (~50km) resolution
  - 127 vertical levels
  - Default values from [default\\_vars.sh](#) — export\_fv3 and export\_tiled functions
  - Start date: 2019-12-03 at 0z
  - Recommended duration: 120-240 hours (5-10 days)



# Prerequisite Software

- Spack-stack contains:
  - National Centers for Environmental Prediction (NCEP) Libraries
  - External libraries
  - Earth System Modeling Framework (ESMF)
- Spack-stack Documentation: <https://spack-stack.readthedocs.io/en/latest/>

***spack-stack***  
***powered by***



# Get Data

- On [Tier-1 systems](#) at the INPUTDATA\_ROOT\* paths in [baseline\\_setup.yaml](#).
- By downloading the data directly from the [HTF data bucket](#) using wget.
  - wget  
https://noaa-ufs-htf-pds.s3.amazonaws.com/develop-20241115/HSD\_fix\_files\_and\_case\_data.tar.gz
  - tar xvfz HSD\_fix\_files\_and\_case\_data.tar.gz

```
develop ufs-weather-model / tests-dev / baseline_setup.yaml
Code Blame 104 lines (104 loc) · 4.05 KB
27 hercules:
28   QUEUE: batch
29   COMPILE_QUEUE: batch
30   PARTITION: hercules
31   dprefix: /work2/noaa/stmp/${USER}
32   DISKNM: /work/noaa/epic/hercules/UFS-WM_RT
33   STMP: /work2/noaa/stmp/${USER}
34   PTMP: /work2/noaa/stmp/${USER}
35   RUNDIR_ROOT:
36   SCHEDULER: slurm
37   INPUTDATA_ROOT: /work/noaa/epic/hercules/UFS-WM_RT/NEMSfv3gfs/input-data-20240501
38   INPUTDATA_ROOT_WW3: /work/noaa/epic/hercules/UFS-WM_RT/NEMSfv3gfs/input-data-20240501/WW3_input_data_20240214
39   INPUTDATA_ROOT_BMIC: /work/noaa/epic/hercules/UFS-WM_RT/NEMSfv3gfs/BM_IC-20220207
```

# Get Code

- To clone the **develop** branch, run:
  - `git clone --recursive -b develop`  
<https://github.com/ufs-community/ufs-weather-model.git>
  - `cd ufs-weather-model`
- I'm using the latest **develop** branch code. The commit at the time of recording for that is **ee3d52d**.
- Optional: save (or "export") the path to the UFS WM:
  - `export WM=$PWD`

# Baseline Configuration

- In [\\$WM/tests-dev/baseline\\_setup.yaml](#), adjust:
  - **dprefix:** Set this value to an existing directory where the user has write permissions.
  - **STMP:** Directory for baseline test output (typically `${dprefix}/stmp4`)
  - **PTMP:** Directory for runtime files (typically `${dprefix}/stmp2`)
- Additionally, update the input data directories if needed:
  - **INPUTDATA\_ROOT:**  
`/path/to/HSD_cases_data/NEMSfv3gfs/input-data-20240501`

# Running the Test Cases

- Run:

```
./ufs_test.sh -a <ACCOUNT> [-s] [-c] -k -r -n "<CASE_NAME>  
<COMPILER>"
```

- where:

- **<ACCOUNT>**: Account/project number for batch jobs.
- **<CASE\_NAME>**: Name of the test case (e.g., **2020\_CAPE** or **baroclinic\_wave** ).
- **<COMPILER>**: Compiler used for the tests ( **intel** or **gnu** ).
- **-s**: Syncs scripts from **./ufs-wm/tests** to **./ufs-wm/tests-dev** (only required on the first run)
- **-c**: Creates a new baseline (necessary until idealized case baselines are staged in the **UFS\_WM\_RT** directory).
- **-k**: Keeps runtime directories after test completion
- **l**: Runs test cases listed in a YAML file
- **-m**: Compares against existing baseline results (baseline must exist)
- **-n**: Runs a single test case
- **-r**: Uses Rocoto workflow manager