# Graduate Student Test Report:
# Medium-Range Weather Application 1.0 Cloud Condensation Nuclei Experiment

*September 3, 2020*

*Report prepared by Cecelia DeLuca. The MRW Application 1.0 release and associated Graduate Student Test had dozens of contributors including the UFS Communication and Outreach Working Group, the Medium-Range Weather Application Release Team, software developers at the Developmental Testbed Center, the National Center for Atmospheric Research, the NOAA Environmental Modeling Center, George Mason University, and many other organizations.*

## About the Graduate Student Test

The Graduate Student Test (GST) is a measure of how successful the UFS project is in opening its code and development processes to the broader community.

The GST assess how easily a student can:

- Get code.
- Run code.
- Change code.
- Test code for correct operation.
- Evaluate code with standard diagnostic packages.
- Get documentation, user support, and training.
- Understand what is needed for their code changes to transition to operations.

There is not a single GST that applies to all codes, and there will be multiple GSTs developed to evaluate the usability of UFS. A particular GST may cover only a subset of the elements in the list above..

Why graduate students? The GST targets students because they are motivated to work with UFS and are important contributors, but may be new to its concepts, acronyms, and assumptions. This makes them an excellent test group for ensuring that UFS materials and processes are clear and understandable for everyone who wants to work with the UFS code.

# Medium Range Weather 1.0 - Cloud Condensation Nuclei Experiment Test Description

This GST assesses how easy it is to access, run, and make modifications to input parameters to perform a sensitivity experiment with the first official release of the Unified Forecast System (UFS) code, the Medium-Range Weather (MRW) 1.0 application. Preparation of the code base and test was a joint effort of the UFS Communication and Outreach Working Group, the MRW Release Team, and software developers at the Developmental Testbed Center, the National Center for Atmospheric Research, the NOAA Environmental Modeling Center, George Mason University, and many other organizations.

The model code used for this test is described here:
https://ufscommunity.org/news/medrangeweatherapp/

The basis for the model is a UFS atmosphere with the Finite Volume Cubed Sphere (FV3) dynamical core. The Common Infrastructure for Modeling Earth (CIME) case-control system manages the workflow.[1]

The model parameters that are changed for the experiment relate to cloud condensation nuclei (CCN). CCN are small particles suspended in the atmosphere on which water vapor condenses. The number of nuclei per cubic centimeter is used in the microphysics parameterization. With more CCN in the air, cloud water is distributed over a larger number of smaller drops. This hinders the development of precipitation processes, so droplets stay suspended and cloud coverage is higher. Even after just 48 hours, an increase can be seen in the cloud cover field. As the forecast progresses, nonlinear processes take place and can change this simple interpretation.

For the GST, participants were asked to download and build the MRW 1.0 application, run it for 48 hours, use the workflow software to make a change to the namelist that increases the concentration of CCN, and perform another 48 hour run with this change. Then they were asked to compare the total cloud cover field before and after the change. There was also an option to shorten the simulation length of the run to 12 hours for people who were performing the test on a laptop, so that the simulation completed more quickly. Participants were given 6 hours to complete the test, and were asked to fill out a questionnaire afterwards.[2]

**Summary:**

[1] *CIME Documentation*, see https://esmci.github.io/cime/versions/master/html/index.html
[2] The questionnaire was developed with the help of social scientists and communication experts on the UFS Communication and Outreach Working Group.

*Link to the test:*
https://github.com/ufs-community/ufs-mrweather-app/wiki/Graduate-Student-Test%3A-MR-Weather-App-CCN-Experiment
*Link to the questionnaire:*
https://forms.gle/Gy94izxqgV8S1gBL8
*Number of respondents:* 8
*Computing platforms supported:* See
https://github.com/ufs-community/ufs/wiki/Supported-Platforms-and-Compilers-for-UFS-Medium-Range-Weather-App-release-v1.0
*Test dates:* 3/11/2020 through 9/3/2020

## Comments and Conclusions

This test assessed the usability of the first UFS community release, the Medium-Range Weather Application 1.0. This application is available through GitHub, run using the CIME case-control system, and documented with a Users Guide[3], GitHub wiki[4], and other supporting documentation.

The Medium-Range Weather Application GST was informed by a previous GST, focused on a prototype UFS subseasonal-to-seasonal (S2S) application. The S2S GST was also run using CIME, following a similar methodology, and established that participants were able to get, build, run, and perform an experiment on a complex coupled code within a similar 6-hour completion window. The earlier GST also helped to refine the questionnaire and experimental protocol.[5]

The primary finding from the Medium-Range Weather Application 1.0 GST is that the usability of the model, workflow, and documentation is very good. This assessment is based on the metric that 5 of the 8 testers could complete the GST experiment within a 6 hour window. The remaining participants were able to complete the test in under 12 hours. The assessment is also based on comments from participants about the ease of use of the application.

It is highly encouraging that the code proved portable, as participants ran the GST on Cheyenne, Stampede2, Hera, and Mac and Linux laptops.

Most of the participants found that the documentation included everything that they needed to get started, and was clear and easy to follow. Several participants noted that the MRW Users Guide enabled them to understand capabilities and options in the code outside of the

[3] UFS Medium-Range Weather Application Users Guide, see
https://ufs-mrweather-app.readthedocs.io/en/ufs-v1.0.0/
[4] UFS Medium-Range Weather Application wiki, see
https://github.com/ufs-community/ufs-mrweather-app/wiki
[5] *Graduate Student Test Report: CMEPS 0.5 - SST Experiment*, see
https://ufscommunity.org/wp-content/uploads/2020/02/gst_2001_CMEPS_0.5_report.pdf

prescriptive sequence in the GST. This represents a significant improvement over the S2S GST, in which there was very little additional documentation.

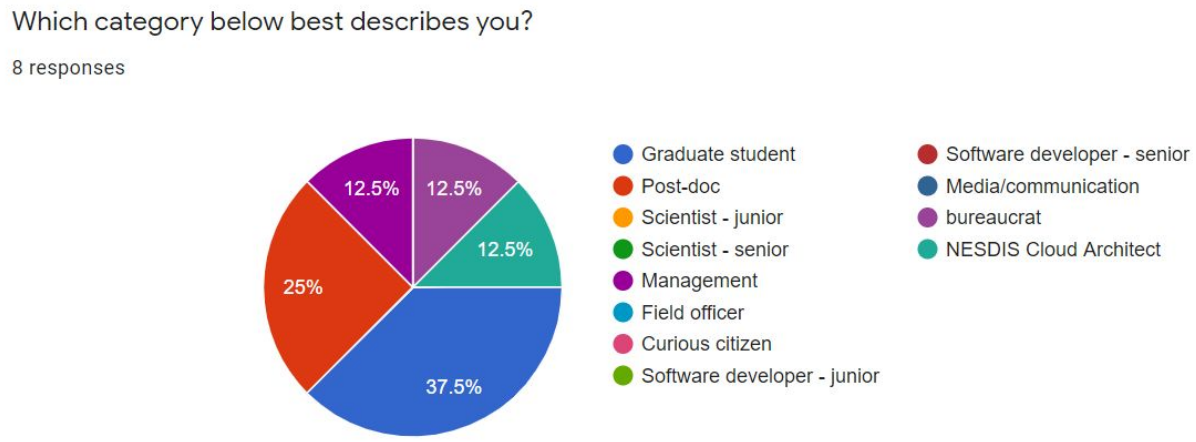
There are many ideas for improving the application in the responses. Some of the suggestions that appeared multiple times were:

- organize the documentation so there is less jumping around, and improve the documentation for the Mac and Linux builds
- provide more information on the Unified Post-Processor and more detail on data output options
- provide more comprehensive tutorials on using the UFS applications

Overall, the respondents were positive, and most expressed interest in using the Medium-Range Weather Application in their work. The last question, “Is there anything else you would like us to know?” inspired most of the respondents to write words of encouragement, such as, “A very good first step in the right direction. Major Kudos to the entire UFS team on getting it this far. Congratulations!” and “Great job. The AMS webinar was very helpful as well. Thanks.”

## Questionnaire Responses

### 1. Information about respondent




If you are a graduate student, what is your major area of study? 3 responses

Atmospheric Science

Climate Dynamics

Atmospheric Sciences

If you are a graduate student, what year are you in or what level of graduate work? (MS or Ph. D)3 responses

Entering 3rd year of PhD

Ph.D

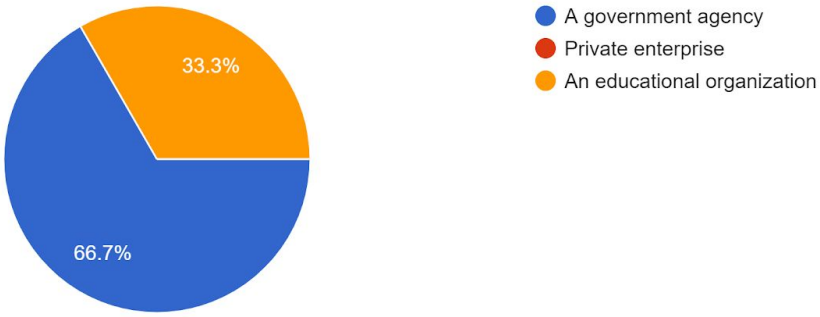
Year 6

Are you employed?

8 responses

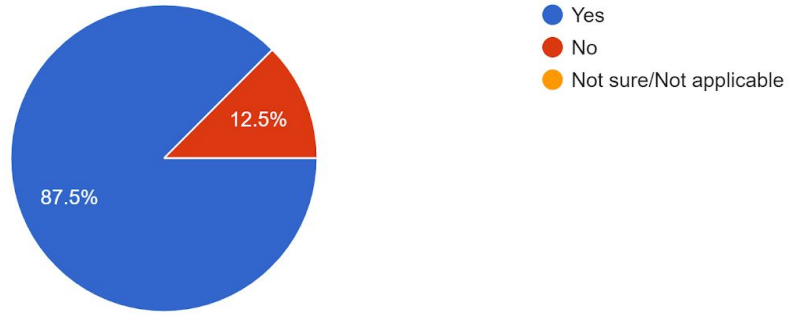

If yes, where are you employed?

6 responses



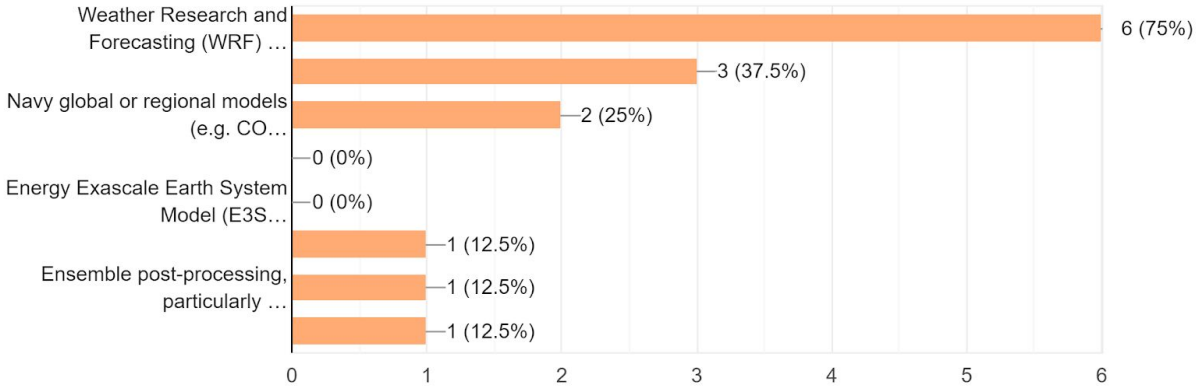
Do you use numerical models as part of your educational program or job?

8 responses



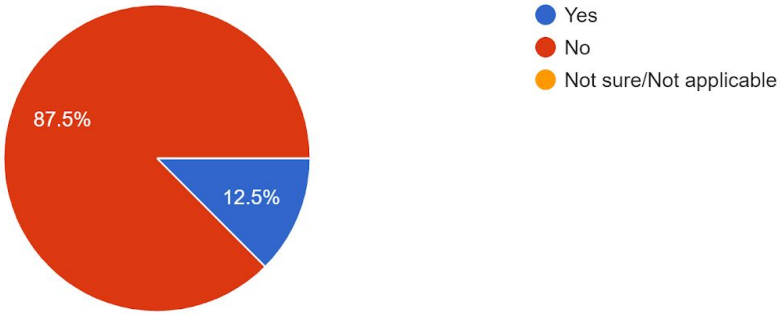

Have you run any of the following environmental models? Please check all that apply.

8 responses



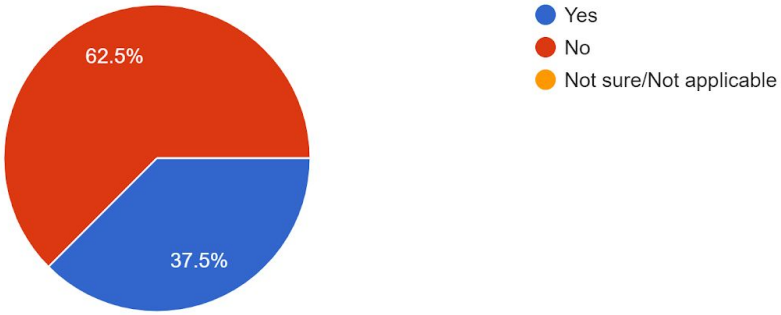
Are you currently involved in the UFS project?

8 responses



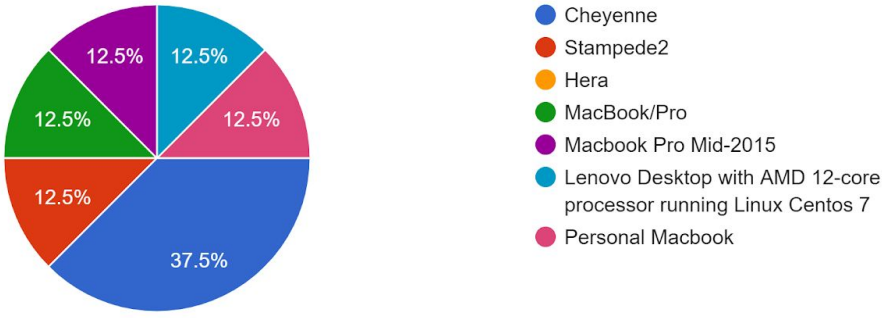

Have you participated in a UFS graduate student test before?

8 responses



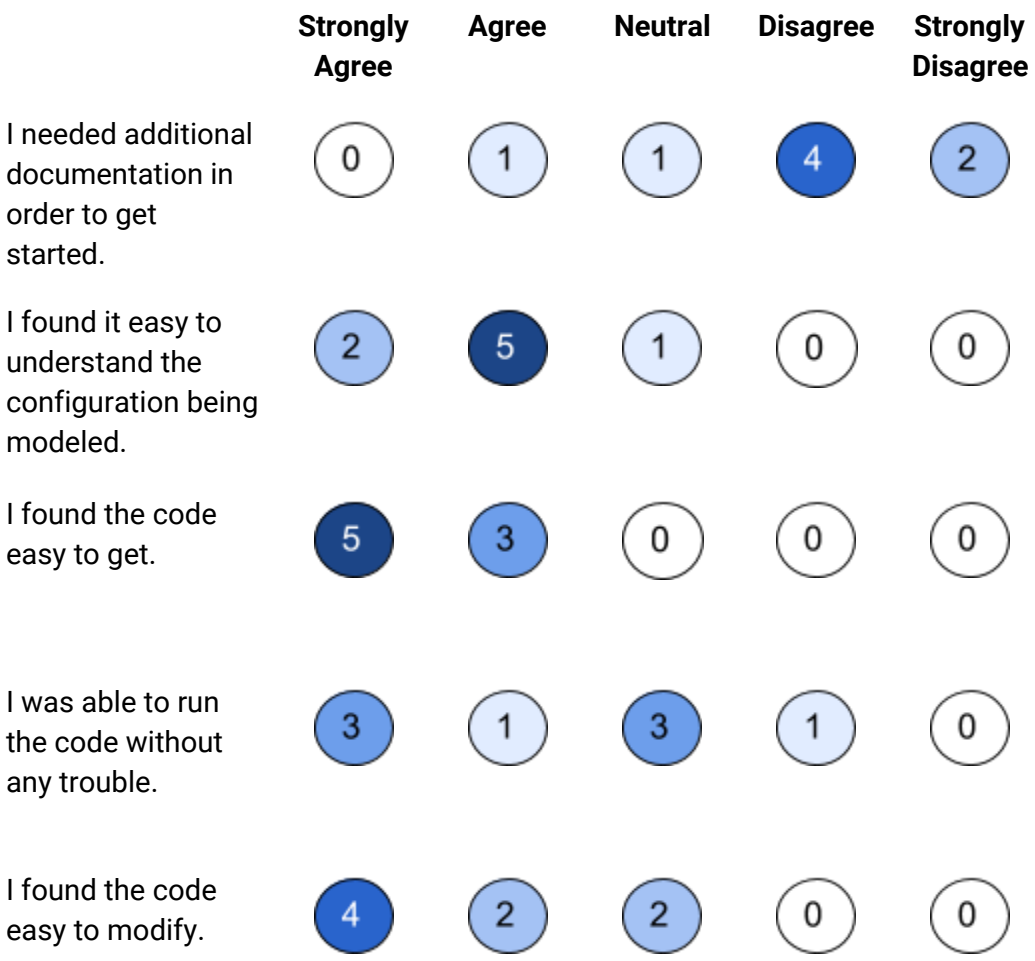

Which computer will you use to perform this test?

8 responses



## 2. Feedback on the test

The number in each circle is the number of people with the indicated response. The color is a visual aid that shows which answers had the most respondents, ranging from no responses (white) to 5 responses (dark blue).




## 3. Free response questions

Did the documentation provide sufficient information for you to understand the nature of the code you were asked to run and how it will be used? If not, what was missing?

8 responses

I thought the documentation/instructions did a great job in explaining what each option was choosing during the setup. I think it was also great to include a link to the User's Guide which went into more details about what the routines case.setup, case.build, etc. were doing.

---

Very easy. I was running on Mac. Any problems I had were from preexisting links to older libraries. When I did a clean build on a new Mac OS install, it worked just fine.

---

The documentation of the steps/code was very clear and straightforward. (There were, after all, very few steps)

---

Yes. I ran into a minor glitch here and there, but overall those were easy to overcome.

---

After the NCEPLIBS and NCEPLIBS-external, setting UFS sandbox was a bit tricky. The instructions were a little scattered in different places to set the correct Environment variables. The instructions were also centered around large clusters like Hera but not for macosx or linux. For macosx, I also had to relax several security settings (UFS.exe needed to be added to firewall settings to allow outbound/inbound connections). My initial environment was running python 3.8.2 and the sandbox does not seem to work with Python 3.

---

It's great that the instructions where clear enough for me to run the code successfully, but there are many other files in the directories I move through that I don't know what they do. The graduate student test was more like following a checklist of tasks rather than really understanding what I was doing. Perhaps future graduate student tests and tutorials could link to references in the User Guide.

---

The documentation provided the information needed to use the code and modify built in parameters.

---

The step I got stuck in is the project ID. The description of the test should remind/clarify what is project ID; a simple example will be helpful. I spent ~1 hour to figure out that project ID is the "group name" and it must be capital.

Were you able to complete the test in 6 hours? If not, how long did it take?

8 responses

Yes, I was able to complete the test in about 4 hours.

---

90 minutes including download and compiling. +1h run time for 12h forecast Dorian case on MacBook with 8GB memory.

The test took me about 10 hours to complete, but this is mostly because I did it on my personal computer (a 2016 Macbook Pro with Catalina v10.15.3). The installation instructions for Mac's are great! I ran into some trouble because it was not an install on a clean OS (I had a bunch of pre-installed packages with Homebrew etc); eventually, I found that the issue was with my Homebrew-installed netCDF libraries. The software compiled just fine after I found this issue. The only other issue I ran into was some python3.x incompatibilities in the CCPP python scripts, which have been reported and fixed by the NCAR folks by this point. Each 48-hour simulation took ~3 hours on my Macbook.

---

Not really. I had some issues to track down such as upgrading my gcc/gfortran/g++ compilers. And the NCEPLIBS gave me a little trouble with the routine grib2_all_tables_module.f. And on my small workstation, it took ~10 hours to produce each 48 hour simulation. So no, I didn't hit the 6-hour mark, but over a "shelter in place" weekend I went from nothing to running the UFS application on my home system.

---

about 11 hours total from start to finish with some debugging of environment and downgrade to python 2.7

---

Cumulative time, yes, I completed the test in less than 6 hours. However, I took about a month to complete it (worked on it an hour here and there, but hey! It was easy to pick up where I left off).

---

Yes, all in all, I completed the test within 2 hours, however I am familiar with CESM and this has a very similar user interface as that.

---

Yes

What could be done to improve the user experience getting, running and changing this code?

7 responses

I think getting the code and running the code are very straight forward. The modification of the namelist through the file ufs_nl_ufsatm feels a bit foreign to me coming from WRF. I don't exactly know the workflow/creation process of the namelist, but I feel direct editing of the namelist is easier since most of the variables that are available to be edited are displayed.

---

Perhaps more on the plotting end. My GrADS build didn't like Catalina, so I installed NCL. It was pretty straightforward, and I'm rusty.

---

I think that the download/installation instructions are really great (the issues I had were on my end), so no recommendations there. Building and running the simulations was also very smooth. The only thing that would help the code-changing process would be more documentation of the various namelist parameters (see my comments below).

---

I thought it was great. Even building the GNU version was well explained in the document.

---

Recommend homebrew installation package for UFS (including NCEPLibs, and UFS sandbox) to ease deployment on macosx. Upgrade the scripts to run with python 3.8. Create consolidated instructions for the entire build instead of linking to other wikis/github, etc. More thorough discussion on Environment variable setup for macosx is definitely needed.

---

My biggest hurdle was getting access to a machine to work on. I did the test on Stampede2 and had trouble getting on the system at first. I never could figure out how to do x-forwarding to view the graphics I made.

---

If by changing the code, you mean making adjustments to optional parameters, I think this is a sufficient example of 'how' to do this, however a larger variety of modifications would give more scope to the options. As far as modifying source code, this test didn't give any examples of this. This may be a more specified group of users who would be interested in this, but I assume there is an easy way to accomplish this and could have been included for future tests.

## Could you use this code in your work? If so, how?

7 responses

I have used both FV3-SAR and FV3GFS to do research into coupling of lake models to these systems.

---

Good question. I wish I was using this code more versus other things at work.

---

With the addition of more documentation (see below) and capabilities (e.g., more grid resolutions), this model would certainly be useful in my S2S research.

---

Maybe...I'm thinking about that. I'm in management at this time, but this is amazing application for students and the community.

---

NESDIS provides a lot of pre-processing data to the models. Having the insights into how UFS works and data is preprocessed would definitely benefit gaining any efficiencies.

---

Not immediately. I am a new post-doc at the Naval Research Laboratory. The primary focus is working on the Neptune model. But there are times I have seen others run WRF for different reasons. This UFS system could be an alternative if the documentation continues to improve and it's easy to get the code compiled and running on a variety of machines.

---

Absolutely, but would need to look up additional information and options not covered by the test and parts of my work would require me to make source code modifications.

## What are the highest priority additions you would make to the code to make it more useful to you?7 responses

I think getting more information on the ability of the post-processing would be very beneficial. From my previous experience with FV3-SAR and FV3GFS, there are a number of different output avenues and it is not instantly clear if/how variables are interpolated. It would be good to know for vector quantities if they are grid relative or earth relative, potentially adding a descriptor to the netcdf variable to clarify the vector field.

My highest priority is to make this more useful to everyone else!

---

The documentation of the basic workflow is great! Now, I want more documentation of levers/knobs that can be turned (e.g., physics options) to make this code more easily applied to scientific questions (e.g., what is the role of different physics schemes or options on atmospheric phenomenon X?). The model output is also a bit unintuitive to me. It would be nice if there were user-defined output streams (like in MPAS) where I could define a set of pre-calculated variables (e.g., 850hPa vorticity) I want output on a gaussian grid in its own custom-named output stream (e.g., "vort001.nc", "vort002.nc" etc.).

---

I'm interested in the UPP and how that can be modified for post-processing applications. I'm also looking forward to the short-term weather application.

---

A more consolidated deployment strategy for running the code on laptop (homebrew based). Also, having a pre-built Amazon Machine Image (AMI) with all the libraries environment variables and ufs sandbox installed for AWS environment on Ubuntu or Amazon Linux would be also help migrate users toward the cloud environment where input data sets could be consolidated in a data lake like solution.

---

I really believe the success of this (and any other similar project) is detailed and updated documentation. I like that you are using readthedocs to host documentation and I would love to see more tutorials and demonstrations. These tutorials should not just walk a person through steps (like this graduate student test does) but explain/teach what every step and command is for. Both the online WRF and Model Evaluation Toolkit (MET) tutorials are great examples and were very helpful to me when I needed to learn those software. A comprehensive suite of tutorials for UFS would really help graduate students want to learn UFS and use it in their research.

---

I am not sure I have sufficient information of everything that is offered in the code (simplified/complex physics packages, alternate dynamical core designs, etc.) in order to suggest an addition, I can only comment on what the test covered.

## Is there anything else you would like us to know?

7 responses

When running the command:
./create_newcase --case DORIAN_C96_GFSv15p2 --compset GFSv15p2 --res C96 --workflow ufs-mrweather
I had to include "--project " PROJECT_ID# for Cheyenne. This is likely an environmental variable that I do not have correctly setup at this time, but something to be aware of.

The command case.setup did not create the input.nml file for me to check in one of the steps. I believe this is not created until case.build.

I know this is typical when using different architectures, but there were some differences when visually comparing my output of column total cloud cover to the image provided in the instructions. Specifically the signal over Greenland was significantly different.

---

You folks are awesome! Keep up the great work!

---

This 1.0.0 release is a great step! Very impressed that I could run it on my laptop. It leaves me craving more documentation of the knobs that can be turned. For example, the number of namelist parameters in the input.nml file is quite large; is there documentation somewhere of what these various parameters do? Also, being a WRF user, I find the "xmlchange" methodology of changing run namelist parameters a bit inconvenient right now. I want to just open one file and see all the namelist options that I can change, rather than using "./xmlquery --listall" all the time (and then run ./xmlchange every time I want to change a parameter). I'm sure there's a good reason why it was done this way, though, and I just need to get used to it. Really nice work on this release!!

---

Great job. The AMS webinar was very helpful as well. Thanks.

---

A very good first step in the right direction. Major Kudos to the entire UFS team on getting it this far. Congratulations!

---

Thanks for being patient with me taking so long to get this completed. With COVID-19 my schedule and routine got messy...glad I was able to finish this and hope to learn more about the UFS as it develops!

---

Overall, I thought it was easy to use for someone like me who has experience running climate models.