

## Envisioning the Future of Community Physics

Grant Firl, Ligia Bernardet, Lulin Xue, Dustin Swales, Laura Fowler, Courtney Peverly, Ming Xue, and Fanglin Yang

### KEYWORDS:

Numerical weather prediction/forecasting; Parameterization; Subgrid-scale processes

### CCPP Visioning Workshop

**What:** A large cadre of physics developers, model developers interested in physics–dynamics coupling, code managers, and computational scientists met virtually for the Common Community Physics Package (CCPP) Visioning Workshop to discuss the current status and future of the project. Key topics included the need for additional functionality, preparing for physics parameterizations of the future, and improving interinstitutional cooperation for collective physics development.

**When:** 15–17 August 2023

**Where:** Online

<https://doi.org/10.1175/BAMS-D-24-0001.1>

Corresponding author: Grant Firl, [grant.firl@noaa.gov](mailto:grant.firl@noaa.gov)

In final form 8 January 2024

© 2024 American Meteorological Society. This published article is licensed under the terms of the default AMS reuse license. For information regarding reuse of this content and general copyright information, consult the AMS Copyright Policy ([www.ametsoc.org/PUBSReuseLicenses](http://www.ametsoc.org/PUBSReuseLicenses)).

**AFFILIATIONS:** **Firl**—Cooperative Institute for Research in the Atmosphere, Fort Collins, and NOAA/Global Systems Laboratory, and Developmental Testbed Center, Boulder, Colorado; **Bernardet and Swales**—NOAA/Global Systems Laboratory, and Developmental Testbed Center, Boulder, Colorado; **L. Xue**—Developmental Testbed Center, and NCAR Research Applications Laboratory, Boulder, Colorado; **Fowler**—NCAR Mesoscale and Microscale Meteorology Laboratory, Boulder, Colorado; **Peverly**—NCAR Climate and Global Dynamics Laboratory, Boulder, Colorado; **M. Xue**—University of Oklahoma, Norman, Oklahoma; **Yang**—NOAA/Environmental Modeling Center, College Park, Maryland

**O**n 15–17 August 2023, 81 scientists, engineers, and code managers in the atmospheric model development community gathered virtually to discuss the current status and the future direction of the Common Community Physics Package (CCPP). The CCPP consists of two pieces: a repository of physical parameterizations with standardized interfaces that comply with a set of agreed-upon rules, and a software framework for autogenerating the interface between the CCPP-compliant physics schemes and a host model (Heinzeller et al. 2023; Bernardet et al. 2023, manuscript submitted to *Bull. Amer. Meteor. Soc.*). The CCPP is currently being used operationally at NOAA within the Hurricane Analysis and Forecast System v1.0 as of June 2023 and is slated to be used in the next operational implementations of the GFS, the GEFS, and the Rapid Refresh Forecast System. Other prominent U.S.-based modeling centers considering adopting the CCPP are NRL [through the Navy Environmental Prediction System Using the Nonhydrostatic Core (NEPTUNE)] and NCAR through the System for Integrated Modeling of the Atmosphere (SIMA).

Although the CCPP has reached a basic state of maturity within several NOAA modeling applications and was written with a goal of host model agnosticism, its use beyond NOAA models likely requires new functionality. In addition, planning for future implementations of physics codes including extra-columnar algorithms (“three-dimensional physics”), full utilization of diverse (CPU/GPU) compute architectures, and coupling with chemistry/aerosol modules has become necessary and is a prudent next step for continued development of the CCPP. Prior to the workshop, the organizing committee created and distributed a survey to likely participants to solicit input regarding the list of topics to be covered during the event in order to prioritize and to be able to address the most relevant topics for all participant groups. The survey results heavily influenced the final agenda which was split into three half-days. The first day contained CCPP overview content to familiarize newcomers with the basic tenets of the CCPP and to serve as review for more advanced participants. The final two days consisted primarily of discussion sessions aiming to answer remaining questions and inform and prioritize future development through an open venue.

### **CCPP overview and review**

The workshop began with four presentations from Developmental Testbed Center (DTC) staff, starting with a broad programmatic overview of the CCPP. It covered the motivation for the CCPP’s existence, its development and use history, particularly how it relates to the Unified Forecast System (UFS; Jacobs 2021), the governance structure for the CCPP repositories, recent code updates, and current and near-future development activities. The second presentation provided an overview of some of the technical aspects of the CCPP. In particular, it explained how the CCPP fits within an existing host model’s infrastructure,

the requirements for a parameterization to be considered “CCPP-compliant,” how suites of parameterizations are assembled, how a host model uses the CCPP application programming interface (API), and finally how the CCPP Framework operates on host-side and physics-side metadata to autogenerate the physics “caps,” or drivers. Next, current code management practices of the CCPP repositories were presented, highlighting the extensive use of GitHub forks, pull requests, code reviews, releases and tagging, and regression testing on multiple compute platforms.

The last session of the first day featured a brief overview of the CCPP Single-Column Model (SCM) and a code walkthrough to demonstrate many of the topics discussed during the technical overview. The SCM overview covered the motivation for its existence, use, and continued maintenance, how one can force the model, and how it shares common components with NOAA’s UFS. The code walkthrough provided glimpses of all of the pieces that make the CCPP work. It demonstrated what is expected of parameterizations that operate within the CCPP system, how host models can configure and interact with the physics, how the CCPP framework ingests information from both the host and the physics to do its job of making the host–physics interface, and finally what the autogenerated pieces of code look like. The first day’s presentations fulfilled the goal of informing participants of existing capabilities and limitations of the CCPP while elevating the baseline awareness of workshop attendees to prime the discussions to be held on subsequent days of the workshop.

### **Improvement of existing practices**

The goal for the morning session of the second day of the workshop was to gather feedback from physics developers regarding potential improvements to current code management practices, documentation, support, training, and releases. Regarding code management, it was determined that one of the highest priorities is to develop a strategy for organization of the CCPP Physics repository. Switching from a flat directory structure to a system with subdirectories for each physical process and individual subdirectories for any host-specific “glue code” will help developers to understand the context of the code better. In addition, it was discussed whether to encourage more scheme developers to use git submodules to maintain a higher degree of control or independence. While this can be positive for some developers, allowing them to centralize their development across hosts, CCPP-compliant or otherwise, it can add a layer of complexity to the host’s code management. Another important issue is the ability to create code “snapshots” that are well-tested or otherwise have known characteristics. Although there is currently a system in place to create git tags for this purpose within the context of NOAA models, it is not clear whether this will be adequate to scale to additional host models. The prospect of creating tags with semantic versioning (i.e., major.minor.patch) every time the entire CCPP Physics repository changes substantially might be an improvement for higher temporal resolution of code snapshots. However, it is not clear whether such tags at the repository level will suffice in terms of code provenance granularity for all purposes; i.e., this does not solve the problem of more carefully tracking code changes at the individual scheme level.

One of the key tenets of the CCPP is to be able to identify variables by a “standard name” that is shared across physics parameterizations and hosts. A dictionary of these exists in its own GitHub repository for the purpose of managing a superset of standard names that are shared among physics parameterizations and host models. Since it is crucial for developers to have an easy-to-use dictionary, improvements were discussed, such as standing up a formal governance, calling out unavoidably host-specific names in the dictionary, and implementing a search function. The standard name is just one of the variable attributes currently found in the CCPP metadata; suggestions have been made to extend these further. For example, adding attributes that designate whether a variable is needed for restart capability, whether

a variable is only used in diagnostics, and whether a variable is permanent (needs to retain its value between time steps) or ephemeral (can be reset, reinitialized, or otherwise forgotten between time steps) were discussed.

Given the community development paradigm of the CCPP, several topics related to community conflict resolution were discussed. One concern is that the CCPP makes it too easy for users to use physics parameterizations out-of-context. It was discussed that the best practice for a scheme's documentation is for the developer to provide use guidelines and restrictions. For example, information about the intended horizontal grid size, the host application from which the scheme originated, vertical coordinate information, and acceptable ranges of tunable parameters would all help to dissuade "misuse" and improper dissemination of results obtained thereof. Second, as more groups contribute to and use the CCPP, proper coordination becomes paramount. An overarching entity to provide such coordination could help to recognize and then to eliminate or otherwise address code divergence when it occurs.

### **Preparation for next-generation physics parameterizations**

Significant work has been and continues to be put into aerosol and chemistry codes and their interaction with physics parameterizations. It has so far remained an open question as to how the CCPP can better support interaction with these types of codes, whose represented processes can be tightly coupled with physics. There seems to be two competing approaches to couple aerosol and chemistry codes with a host model and its physics—one where the relatively looser coupling happens through the host model and one where the relatively tighter coupling happens through the physics. Examples of both approaches can be found within the NOAA modeling ecosystem. The CCPP is better positioned to interact with the tighter coupling paradigm, given its built-in flexibility, although some concerns have been raised related to the use of immutable standard names for context-sensitive variables used in some chemistry modules. A path forward was discussed for creating a CCPP-sanctioned abstract derived data type to ameliorate this concern.

One of the existing limitations of the CCPP is its assumption of columnar-only physics schemes. Parameterizations that have horizontal components, i.e., "three-dimensional physics," are something that the CCPP may need to support, and, therefore, were a discussion topic during the workshop. One of the outcomes of the discussion was that, in general, extending support for three-dimensional physics was deemed to be challenging and to have questionable cost–benefit metrics. Horizontal grid geometry, indexing, and domain decomposition issues are likely too host-specific for the CCPP to efficiently handle. Should it be deemed appropriate to eventually support three-dimensional schemes, it was suggested that the CCPP could define an abstract base class and associated methods. Because host models are required to know about the particular grid geometry being used, they would be required to extend such a base class and its methods in order to provide functions like supplying halo indices and geometry, for providing area averaging functions, or for de-staggering variables to potentially enable some three-dimensional operations from within CCPP physics schemes.

One benefit of the columnar nature of most atmospheric physics is that they lend themselves well to parallelization. Until recently, the CCPP has primarily been concerned with coarse-grain parallelism using threaded, multinode CPU compute architectures. Utilizing fine-grain architecture with graphical processing units (GPUs), as well as arbitrarily heterogeneous (between CPU and GPU) compute architectures has become a priority. The workshop featured a plenary discussion on this topic and some emphasis was placed on the choice of accelerator technology, e.g., CUDA versus openACC. Although some schemes within the CCPP Physics repository already have openACC directives for GPU acceleration, it remains unclear as to which technology is most prudent to support. OpenACC directives probably have a lower barrier-to-entry to physics developers, while CUDA is seen as likely

more performant and less vendor-locked. Code management questions surrounding whose responsibility (developer versus CCPP code maintainers) should enable and maintain execution on GPUs, how GPU-enabled code can be regression tested, and what computational penalties there may be for the desired heterogeneity remain open.

Significant time in breakout sessions was devoted to discussing a broad range of potential new functionality to be added to the CCPP. A through line that arose was the desire to enable greater consistency among schemes in a suite. One step toward this goal is for the CCPP to provide a module for common atmospheric physics functions. Such a module could be used by all CCPP schemes to provide consistent methods for common functions that need to be called at multiple points within a suite's execution, like the calculation of saturation vapor pressure, for example. Since the goal is consistency across the entire model and many hosts likely already carry some functions of this type (including in their postprocessing), care must be taken to ensure consistency and code reuse across all model components to realize the full benefits. A similar well-received idea is for the CCPP to provide functions to automatically convert variables, e.g., from temperature to potential temperature.

Another new functionality discussed relates to flexibility for time and space discretization. The CCPP could be made an even more useful tool for physics-dynamics coupling research by adding functionality to flexibly choose which schemes (or groups of schemes) in a suite should use process- versus time-split (i.e., parallel versus serial) time coupling. For vertical discretization, a longstanding issue has been that some modeling groups define arrays and loops as surface-up and others define them as top-of-atmosphere-down. The CCPP should be able to handle schemes written for either orientation, either by providing automatic variable flipping or requiring that schemes write vertical orientation-agnostic loops. For horizontal discretization, two discussion topics arose. One was related to the use of subcolumns within physics (e.g., for fractional surface tiles, multiple convective plumes, etc.) which often require splitting and aggregation of subcolumns and/or multiple scheme calls. It was agreed that the current, manual coding approach, with associated variable proliferation to handle these situations, is not portable between hosts and generates significant code maintenance issues. A generalized approach handled by the CCPP could add a lot of value for physics developers and code maintainers and could have an added benefit of contributing to a computational load balancing solution. A second horizontal grid issue is for the CCPP Framework to add the ability for schemes within a suite to run on different horizontal grids (e.g., radiation on a coarser grid). While this topic elicited interest, it was not considered high priority.

## Conclusions

A significant debt of gratitude is owed to the participants of the workshop. The gathering enjoyed relatively broad participation from four NCAR laboratories (Research Application Laboratory, Mesoscale and Microscale Meteorology, Climate and Global Dynamics, and Atmospheric Chemistry Observations and Modeling) and six NOAA laboratories and centers (Physical Sciences Laboratory, Global Systems Laboratory, Environmental Modeling Center, Geophysical Fluid Dynamics Laboratory, Chemical Sciences Laboratory, and the National Severe Storms Laboratory), making up approximately two-thirds of the group. The final one-third of participants hailed from the U.S. DOE Pacific Northwest National Laboratory, the Joint Center for Satellite Data Assimilation, the United States Naval Research Laboratory, the Brazilian National Institute for Space Research, the Stevens Institute of Technology, the University of Maryland, the Central University of Rajasthan, and the Norwegian Meteorological Institute. The strong cross-institutional engagement in the CCPP project as evidenced by this workshop is a fine example of the spirit of scientific collaboration and signals a robust inclination among the community to collaborate on physics development. Through the continued

development of this software, the goal of easily sharing physics codes across institutions and their models should “lift all boats” toward the ultimate aims of scientific discovery and ever-more-useful weather forecasts.

**Acknowledgments.** This work was supported by the Developmental Testbed Center with funding from the National Oceanic and Atmospheric Administration Oceanic and Atmospheric Research through Contract W8R2Q80. The Developmental Testbed Center is funded by the National Oceanic and Atmospheric Administration, United States Air Force, and the National Center for Atmospheric Research. The National Center for Atmospheric Research is a major facility sponsored by the National Science Foundation under Cooperative Agreement 1852977. This work was also supported by the Bipartisan Infrastructure Law. The corresponding author further acknowledges support from National Oceanic and Atmospheric Administration Award NA 19OAR4320073, a cooperative agreement between NOAA and the Cooperative Institute for Research in the Atmosphere.

## References

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

Jacobs, N. A., 2021: Open innovation and the case for community model development. *Bull. Amer. Meteor. Soc.*, **102**, E2002–E2011, <https://doi.org/10.1175/BAMS-D-21-0030.1>.