Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., referee comment RC1
https://doi.org/10.5194/gmd-2021-419-RC1, 2022

Comment on gmd-2021-419

Anonymous Referee #1

Referee comment on "UniFHy v0.1.1: a community modelling framework for the terrestrial water cycle in Python" by Thibault Hallouin et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-419-RC1, 2022

This paper addresses the topic of component modeling in the hydrological and land surface modeling communities. This is an important and current topic—in the US, it has received attention from government agencies such as NSF, DOE, NOAA, and USGS.

In the paper, the authors introduce `unifhy`, a component-based framework for hydrological modeling. The framework is written in Python. In the framework, a model is constructed from three components, representing the physics of the land surface, the subsurface/soil, and open water. Components can be swapped, either for another component (written as a Python class with a prescribed inheritance representing the physics of the process and satisfying the required inputs and outputs), or for a data file (a data component), or for a null component, which turns off an individual process. Users can write their own components; e.g., to implement a custom algorithm, and the authors demonstrate this by pulling from a library of components they've created. Components can be written not only in Python but also in languages such as C, C++, or Fortran, then wrapped into Python using tools such as Cython or pybind11. When the model state is evolved through time, information is exchanged between components at common multiples of time intervals using a controlled vocabulary. Interpolation between component grids is handled by the ESMF regridder. A model can be constructed from code or from a configuration file. The framework allows checkpointing, so a user can start, stop, and resume a simulation. The authors demonstrate the utility of the framework through a matrix of case studies, combining different components with data from different catchments.

The techniques used for modeling and software development are valid; in particular, the software design is well-planned. I appreciated the clear and descriptive UML diagram of the framework architecture (Figure 2) as a complement to the text. The framework makes good use of existing and well-known libraries (cf-python, netCDF4, pyproj) and standards (CF, BMI); this is the modern and accepted way to build an open source software package. Limiting the framework to only three components is a clever idea. It simplifies and focuses the modeling problem. A tradeoff is that it may be difficult for a user to write a component that satisfies the needed physics and input/output parameters. This issue may be ameliorated by the template provided by the authors on GitHub, as well as the

choice of Python as the core language—it's free, open source software, it's easy to learn, and it's used widely across disciplines in the physical sciences.

This isn't technically part of the review, but I wanted to complement the authors on their code—it's well-organized and well-written; there's testing on multiple levels, continuous integration, licensing, issue tracking, documentation, and tutorials. It has everything I want to see in an open source project. I installed unifhy locally and experimented with the code in the tutorial, although I didn't complete it because I didn't have appropriate input data. It would be good to have the input data hosted somewhere for users of the tutorial.

I have one major concern. As a researcher familiar with Landlab, I find that unifhy is very similar to Landlab, from concept to code. Because of this, I feel that the paper does not currently meet items 2 and 3 from the GMD review criteria:

- Does the paper present novel concepts, ideas, tools, or data?
- Does the paper represent a sufficiently substantial advance in modelling science?

What I would like from the authors is a justification for how unifhy differs from Landlab, what new ideas are incorporated into its design and UX, and how unifhy represents an advance over or a novel alternate to Landlab in modeling. I don't think this demand is onerous—an extra paragraph or so should suffice. I'm imagining the perspective of a new graduate student in hydrology reading this paper, and I'd like them to understand how unifhy builds on the previous contributions of others.

I enjoyed reading this paper, and I want it to be published, but the authors haven't quite shown that unifhy is a unique contribution to science.

Minor corrections:

- Line 30: For Landlab, please also cite Barnhart et al. 2020: https://doi.org/10.5194/esurf-2020-12 (this is a PDF link)
- Line 31: Basic Model Interface instead of Basic Modelling Interface