Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-409-RC3, 2021

Geoscientific Model Development Discussions

Interactive comment on "SuperflexPy 1.2.0: an open source Python framework for building, testing and improving conceptual hydrological models" *by* Marco Dal Molin et al.

Anonymous Referee #3

Received and published: 20 January 2021

The paper "SuperflexPy 1.2.0: An open source Python framework for building, testing and improving conceptual hydrological models" by Dal Molin et al. describes the development and implementation of a flexible hydrological modeling platform. SuperflexPy is based on the earlier SUPERFLEX model, but uses the Python language and adds new features.

The paper is well-written and generally well-structured, and I believe that SuperflexPy can provide a valuable and flexible tool for hydrologists. But I have some concerns with how the authors frame and present the model.

1. In section 1.3, the authors state that their 3rd aim is to provide a broad discussion

of how SuperflexPy contributes to the hydrological community and how it compares to existing modeling tools. The included discussion does very little to situate SuperflexPy with respect to existing models. The authors should include a discussion about the similarities and differences, as well as perceived advantages and disadvantages, of SuperflexPy with respect to existing model frameworks. Why should someone use this model rather than previously available tools?

2. On line 160, the authors state that a flexible model framework should cover substance transport modeling, but the version of SuperflexPy presented does not have modules to handle substance transport. It would be possible to add substance transport, but that's presented as a hypothetical rather than an existing part of the model framework.

3. Beginning on line 162, the authors outline three computational criteria that a model should meet: ease of use, including interoperability with external software, ease of modification, and computational efficiency. Of these three, I think that only ease of modification is adequately addressed in the manuscript. Installation and operation are discussed, as the authors outline the strengths of Python as an object-oriented and commonly used programming language. But they do not discuss operability with external software, model calibration, or uncertainty analysis. There are references to recently published manuscripts that seem to cover these topics in more detail, but I believe that more discussion is warranted.

Section 4.4 covers computational efficiency, but does not provide much quantification or comparison with other models. The authors state that Numpy and Numba provide a speed up compared to native Python of "factors up to 30" and state a difference between "a few seconds with the Numba implementation compared to a couple of minutes with native Python execution" but do not provide sufficient details about the structure or complexity of the model used to calculate these runtimes. A comparison to other modeling frameworks would be especially valuable.

4. I think that Sections 4.1, 4.2, 4.3, and potentially 4.5 should come before Section 3. These detail the computational structure of the model and would follow the initial description in Section 2 quite well. An expanded version of section 4.4 could then be incorporated into an expanded discussion comparing SuperflexPy to other model frameworks.

Specific line comments are included below:

125-128: This paper doesn't explore model comparison, and it's not clear to me how the latter half of this paragraph informs the manuscript.

178: The term granularity is unclear

224: The terms element and unit are somewhat unintuitive. It's clear how nodes connect in a network, but it's not clear why a unit should be made up of elements.

250. The acyclic directional graph seems like a significant drawback – backwater effects, capillary action, hyporheic flow, etc. are common hydrologic occurrences. How do other modeling platforms handle these?

270: The definition of Levels should be introduced prior to this

315: Topography should be topology

351: Are specific units required? A small discussion of how the model handles units would be helpful.

371-373: Unclear whether importing data as a Numpy array is required

436: If nodes share units, does that mean that the unit is duplicated in each node?

443-445: It's not clear to me why connectivity gaps would occur and why a transparent element is the only way to solve this issue.

500: Are there memory concerns with retaining history, particularly for complex systems or long runtimes?

528: Numba was mentioned before but is detailed for the first time here.

615:SuperflexPy used to construct and compare

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-409, 2020.