Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-409-RC2, 2021

Geoscientific Model Development Discussions
Open Access
EGU

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 #2

Received and published: 18 January 2021

The paper titled "SuperflexPy 1.2.0: an open source Python framework for building, testing and improving conceptual hydrological models" by M. Dal Molin et al. details the development and implementation details of a new flexible hydrological modelling framework. The framework is based upon an earlier code SUPERFLEX developed by the second and third authors (Fenicia et al., 2011), but re-built using object-oriented Python programming approaches.

The paper is generally well-written and appropriately structured, and the software is clearly the fruit of much labor and potentially worthy of its own publication. However, I am not sure that the authors have presented a sufficient argument as to the unique

Printer-friendly version

Discussion paper

value of this contribution. I have a number of addressable concerns mostly with respect to evaluation and assessment of the framework:

1) Most importantly, I think the authors miss out on an opportunity (and expectation) to distinguish this effort from other modelling frameworks cited herein. What makes SuperflexPy unique? What types of problems may it be applied to that other flexible frameworks cannot readily tackle? It would be useful to illustrate any perceived advantages via one or two case studies. In particular, the authors need to make a very strong case as to why this implementation is particularly advantageous relative to the original SUPERFLEX code, since the conceptualization seems very similar. Is it merely the object-oriented Python wrapper? If so, is this alone a sufficiently unique contribution for this journal? As part of this, they will necessarily have to discuss some of the strengths and weaknesses of existing modular hydrology tools and what role SuperflexPy takes in addressing perceived gaps. This is the most critical comment for the authors to address.

2) The authors refer a number of things that might someday be done using the framework but have not yet been implemented, which I found problematic. Specifically, they discuss transport of contaminants and isotopes and use of a more complex numerical solver. However, none of these advances have actually been implemented in this model. This content needs to be removed, as it is not a current advantage of the software tool, it is a hypothetical future advantage.

3) The evaluation of the computational efficiency lacks rigour. There are insufficient model details to evaluate the computational benefits of SuperflexPy or the specific Numba implementation, and the speedup is quantified as improving from “a couple of minutes” to “a few seconds” of runtime. A quantitative assessment is warranted here if the authors wish to make a defensible argument regarding computational efficiency.

4) The authors have made a unique choice of coupling all of the constitutive laws for fluxes into a single element, i.e., use of the UnsaturatedReservoir() element implies use

of the relationships in Eqns 6-8. This is quite different from what is seen in models such as SUMMA, MARMMoT, or RAVEN where the swappable "element" is the constitutive law rather than the collection of constitutive laws applied to one storage element. Can you justify this selection and/or discuss the implications of this approach as compared to the flux-based components? It seems like just swapping one of the constitutive laws for a storage element will often necessitate creating a new component. Likewise, I would like to see a defense of the use of a fixed number of layers which necessitates the use of “transparent elements” and a clarification of the role of these layers – why are they even necessary? What problem do they solve?

5) I recommend including a UML depicting the inheritance structure and currently implemented elements in the SuperflexPy code (or a subset of these in the UML with a list/table of elements elsewhere). It is very unclear how the breadth and quantity of options compares to other modular frameworks including the original SUPERFLEX.

I have included most of my minor comments in the attached PDF file.

Please also note the supplement to this comment:
https://gmd.copernicus.org/preprints/gmd-2020-409/gmd-2020-409-RC2-supplement.pdf

---

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

Printer-friendly version

Discussion paper