Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., referee comment RC1
https://doi.org/10.5194/gmd-2021-53-RC1, 2021

Comment on gmd-2021-53

Anonymous Referee #1

Referee comment on "HydroPy (v1.0): a new global hydrology model written in Python" by Tobias Stacke and Stefan Hagemann, Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-53-RC1, 2021

The model description paper "HydroPy (v1.0): A new global hydrology model written in Python" describes a reimplementation of the global hydrological model MPI-HM.
Overall, I certainly support the publication of this study; however, many questions remain unanswered, requiring a major revision of the current manuscript.

First of all, I think it is tremendous that the authors approached the challenge of rewriting a scientific software! A lot of software used in the community is stuck in the last century. Such an endeavour, reimplementing a research software, requires work that generally does not result in any measurable new science but still is extremely important and currently undervalued. While the authors made some reasonable choices in technology (choice of language and libraries), they completely ignore the work that has been done on other modelling projects. Furthermore, they claim benefits of the reimplementation that are not supported by any evaluation.

Detailed Comments:

(1) The title struck me as very odd. There are other global models as well that are written in Python, for example, PCR-GLOBWB; which also has been described in a model description paper in this very journal and has not been cited! In general, much relevant literature on global hydrological models is ignored. Indeed, this is not supposed to be a review paper but only citing your own modelling papers is not something that acknowledges the work of others and advances science. This also relates to the evaluation, which also did not include any comparison whatsoever to other models. At least the differences should be discussed if a full out comparison is not possible or due to calibration limited.
(2) As already mentioned, I think a significant contribution is the reimplementation of a model. Sadly the authors only included minimal information and no discussion on the involved challenges etc. This would probably merit its own perspective piece in GMD, but still, it would benefit the community greatly if you would provide more insights on that process. Also, you should consider citing "Muller et al. Going open-source with a model

dinosaur and establishing model evaluation standards, EGU 2018"
(3) It is great that the authors published their code as OpenSource, but I don't quite understand why the authors choose to upload the code to Zenodo but not to a platform like github or bitbucket. Or is it available there as well? Then please add a link the code availability section. It would benefit the community greatly if the code and its further development process are more accessible. This is, of course, nothing that should influence a decision on if this manuscript should be published but still something worth noting.
(4) In the abstract, the authors claim that "the new model requires much less effort in maintenance and due to its flexible infrastructure, new processes can be easily implemented". I do not see any evidence for these claims. While the code is thoroughly documented, files with over a thousand code lines and 3-4 classes do not reflect a carefully designed software architecture with extensibility and maintainability in mind. The paper is not a computer science manuscript, and thus I have to credit that they provided a rather clean implementation. Yet, the authors need to either compute metrics that support their claims or discuss their software architecture in more detail explaining how it supports the integration of new processes. Did you use particular software patterns to ensure that? Or do you just hope to achieve that because you used a more modern language? I do not want to discredit the tremendous work that probably went into the implementation, but such claims need to be supported; otherwise, it is not much of a scientific publication. Furthermore, I urge the authors to take a look at the review guidelines of JOSS to improve their code further. Currently, you don't have any automated tests that would allow a 3rd party, and more importantly, you, to check if your software is working correctly. Again this is not a criterion that GMD is added to its guidelines and will not prevent me from supporting a publication.
(5) Please add the central variables to figure 1. It would help significantly understand how the implemented processes work together and keep an overview of all the variables.
(6) Please add a table of all variables, a short explanation and in which equation they are used and if they are available as output.
(7) You mention the land surface property data in the data availability section but not the output data used in the evaluations section of the model. Please make this available or state why it is not possible. See also a recent paper in GMD as a possible example: https://gmd.copernicus.org/articles/14/1037/2021/
(8) Line 71: How is f_snlq,max determined?
(9) Line 95: Please discuss the implications of not considering groundwater recharge (defuse and focused)
(10) Line 139: Unclear, please elaborate
(11) Line 175 ff: Why not as inline c code and compile with cython? That would make it more accessible. In general, could you provide some performance metrics? Is the new implementation much faster/slower?
(12) The whole comparison to MPI-HM is solely focused on the NSE. This does not prove that you are getting identical/similar results for the right reasons. Again automated tests would be a great addition. Furthermore, I greatly recommend a full sensitivity analysis using, for example, Morris.
(13) You should discuss the performance of your model with respect to other available models.