Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., author comment AC1
https://doi.org/10.5194/gmd-2021-53-AC1, 2021

Reply on RC1

Tobias Stacke and Stefan Hagemann

Author 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-AC1, 2021

We thank the reviewer for the critical and constructive discussions on our manuscript. Here, we will repeat the reviewer's comments (italic font) and response directly below (standard font).

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.

We thank you very much for the appreciation of our work. We fully agree to the challenges you mentioned that were posed by this project. As the critical comments are repeated in the detailed comments section, we will discuss them there.

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.

With the choice of title we did not intent to imply that our model would be the only one

written in Python, but rather wanted to provide context for our choice of the model name. For this reason, we made no dedicated effort to find and review other GHMs that use Python. Nonetheless, we will add a short paragraph on the general state of model development to the documentation and think about options to rephrase our title.

We like to emphasize, that we did not limit ourselves to citing only our own modeling papers. In most of the cases, we provided references to studies, which resulted from international inter-comparison projects (WATCH and ISIMIP) where a large number of GHMs, including the MPI-HM, took part. The cited papers also provide plenty of information about the scientific performance of the MPI-HM compared to other GHMs. Considering that HydroPy is not a new model in terms of its hydrological equations but a remake of the MPI-HM, we do not plan to repeat these comparisons with other models at this stage. Instead, we demonstrated HydroPy's similarity with MPI-HM, which implicitly shows that the evaluations done for MPI-HM are still valid for HydroPy. In addition, we evaluated HydroPy against observational data, because this was not yet done for MPI-HM in previous publications. Of course, we will modify the text to make readers aware of the available model evaluation studies on MPI-HM. However, we feel that exercises like standardized model inter-comparisons for complex models like GHMs are a scientific project by themselves and beyond the scope of a model description paper.

(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"

Thanks for proposing a section on the challenges encountered during this project. This is a very good idea and we will add a paragraph on this to our manuscript. In summary, we find that such efforts are difficult as usually no (third party) funding is available for modernizing a scientific software. This leads to a lack of resources, especially of personnel with the required skill set and dedicated working time, and limits the model development perspective to the end of a scientist's contract. Furthermore, there is a difference between a 'good' model from the perspectives of scientific researchers and software engineers, as there has to be a balance between how easy model routines can be understood and modified versus the pure technical model efficiency in terms of computational resources and scalability. And finally, as the reviewer already mentioned, the direct scientific gain is rather low although publication of such effort in journals like GMD finally allow the scientists to earn some merits with technical work.

(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.

No, we published HydroPy only on Zenodo. The reason is directly related to our answer to remark 2. Zenodo can be used as a repository for source code (and data) and even allows to add new versions whenever we are able to work on model improvements. This suits our main interest, which is to support transparency in science by making our source code public. In contrast, platforms like github are more tailored towards interactive model/software development together with the respective community. Of course, we are aware that such collaboration can improve software significantly, as very skilled people might join the effort. However, our limited resources do not allow for any reliable model support beyond providing a manual with setup information. Furthermore, the direction of

model development for HydroPy is determined by the projects we can get funding for and not by the needs and wishes of the community. While this situation is not satisfying for us either, we feel that publication on Zenodo enables us to make most of our limited resources.

(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.

You are right: these claims are not supported in the manuscript. Actually, our statement concerning the infrastructure and model design were not meant in an objective way, but rather subjectively in comparison to the model structure and setup of the MPI-HM model. We fully acknowledge that there are probably concepts around to improve the structure of the code and its maintainability. However (see response to remark 2), we do not have a software engineer in our team and, thus, are limited to the scientist's point of view. Nonetheless, we agree that some information should be added to the manuscript. We are not aware of metrics to objectively evaluate the quality of our model design, but we will add a paragraph on our reasoning behind the code structure and our intended workflow to add new processes.

(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.

We deliberately omitted this information from the figure as we wanted to present a general overview over the hydrological processes realized in the model. Adding specific variables to the figure requires much more (and therefore smaller) text and interaction descriptions. However, we will try to add as much of these information as possible without making the figure too confusing.

(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.

We will add such a table.

(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/

At the point of submission, we were not convinced that the raw model output is of major interest to the community and did not want to potentially waste storage on any public repository. Of course, we are happy to follow your request and we will add a link to our

manuscript.

(8) Line 71: How is f_snlq,max determined?

The value of 6% for maximum liquid snow water content is commonly used e.g. in Wigmosta et al 1994. We will add this information to the manuscript.

(9) Line 95: Please discuss the implications of not considering groundwater recharge (defuse and focused)

This might be a misunderstanding, thanks for making us aware of this. In our simple model structure, groundwater recharge R_gr is set equal to drainage and subsurface runoff and is used in the balance equations of soil moisture (Eq. 13) and shallow groundwater (Eq. 24). We will add groundwater recharge to the list of synonyms for drainage.

(10) Line 139: Unclear, please elaborate

In MPI-HM the vertical land surface water balance module and the routing module were strongly separated, each featuring storages which were restricted to be used within the respective module. The sole exception was the surface water storage resulting from ponding water on the land surface, and, hence, used to to represent small creeks, lakes and wetlands. This storage was part of the vertical land surface water balance but could also interact with the lateral river routing scheme. In HydroPy, we simplified the representation of lakes and wetlands and utilize storages that are already part of the routing scheme.

(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?

Prior to using fortran, we experimented with a cython implementation. However, the increase in performance was not as high as we hoped for and, thus, we switched to a fortran version. We tested this with simulations done on an office laptop (Intel Core i5 9400H with 8GB Ram). Without writing any output, simulations at 0.5 deg on a global domain for 1 model month took:

pure python	22.3 s (+- 160ms)
python + cython routing	18.9s (+- 111ms)
python + fortran routing	13.6s (+- 38ms)

Please note, this numbers are from an older model version and probably not up to date anymore. We will provide performance numbers for the pure python and fortran routing version in the revised manuscript.

(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.

This is done, because river discharge is essentially the target variable of our GHM and it depends on all prior computations done in the model. Thus, focusing on river discharge to demonstrate the similarity in model output between HydroPy and MPI-HM appears as the most logical choice to us. Of course, you are right that similar results could theoretically be achieved for the wrong reasons. However, considering that both models are based on the same physical hydrological equations, we think the likelihood for such a ‘false positive’

would be rather low. Additionally, while the (N)NSE admittedly plays a large role in our comparison of river discharge, it is by no means the sole focus. For river discharge we present climatological river discharge curves for catchments of interest to explain differences between both model setups (Fig. 11 in the manuscript) and before that we compare and discuss average water fluxes and storages which exist in both models (Fig. 9 in the manuscript).

Concerning automatic testing, we will keep this recommendation in mind. But as we don't have any software engineer available for such task, we expect the development of meaningful tests to take a certain amount of thinking and revision and won't start on this in the framework of the current study.

Similarly, we agree that sensitivity analyses like the Morris One-At-a-Time or Latin Hypercube are very useful tools to identify the most important parameters in a model and prioritize them concerning tuning, optimization or simplification. However, in this study we just want to do the first step, namely the re-implementation of the MPI-HM in a modern form as HydroPy. While improving on the original formulations is certainly on our agenda, it has to wait until our limited resources allow for this.

(13) You should discuss the performance of your model with respect to other available models.

We are not sure whether you refer to the scientific performance or the technical one. Concerning scientific performance, we already commented on this in our reply to remark (1). Of course, we will include our reasoning about the implicit comparison due to the similarity to MPI-HM at the end of Sect. 4.4. Concerning the technical performance, we can provide numbers about model runtime, but we don't see much value in comparing those to other models. GHMs vary strongly in complexity and the number of implemented processes. Whether a model has a higher (computational) efficiency compared to another would be only relevant if both are equal in their outcome as well as scope of applications. Thus, we cannot include such a comparison in our manuscript and furthermore argue that such a project would be a major study on its own and beyond the scope of a model documentation paper.