Dear Referee #1,

Thank you for providing good comments to improve the manuscript.
Specific comments:

1. Is there any statistics for the number of users of C language vs FORTRAN language to highlight the need of developing a C-based model system?

Reply: Here, "C language" is a misspelling, while C# language is correct. According to the TIOBE Programming Community Index (www.tiobe.com) for October 2024, C# is the fifth most popular of all major programming languages, with 5.6% of users, while FORTRAN is the ninth most popular, with 1.8% of users.

After our careful thinking, these statistics will not be present in the manuscript, because the number of the language users is not the main reason to reconstruct the model. The main reason is that C# is a modern and powerful language, and is easy to use (by taking advantage of many powerful code analysis tools for C#), while FORTRAN is a traditional old-style language, which is more difficult to use for many users, due to limited code analysis tools.

2. Table 1 caption and related text in the manuscript: please clarify that this is the Noah-MP model version in WRF-Hydro v3.0 not the latest community Noah-MP model version.

Reply: The caption is now changed to "Figure 1. The architectural diagram of NMP-Hydro (a) and the conversion of FORTRAN arrays to C# arrays (b). NMP-Hydro is a reconstructed replica of the version of Noah-MP that is coupled in WRF-Hydro 3.0. "

The related text in the manuscript is also minorly modified now. Generally, the original description has mentioned the version.

3. Does the river routing model have to run at the same spatial and temporal resolution as the main Noah-MP column model?

Reply: according to this comment, we have added following paragraph in section 3.2:

"*This module requires two additional inputs files, a river segment list file named 'ChannelOrder.txt' and a 'namelist.txt' file. The latter file is used to set parameters and the length of time step. Each river segment in the list file presents following information: its own index, the index of its next downstream river segment, the row number and the column number of the grid box (in Noah-MP's running domain) providing runoff input to the current segment, the length (m) of the current river segment, the two parameter values (K and X) of the Muskingum method, the area of the catchment of the current segment. Each river segment upstream of other segments must be listed ahead of all its downstream segments. The river segment list can be derived from both gridded river network or vectorized river network. The resolution of the river routing is determined by the original river network from which river segment list is derived. Therefore, the choice of using vector river network or gridded river network and the selection of spatial resolution are completely determined by the river segment list file which is provided by the users. The length of the temporal step of the river routing is required to be multiple times shorter than the time step for running the Noah-MP, and can also be designated by the users. In our application, the time step of routing is set to 600s or 900s, while the time step for Noah-MP LSM is set to 3 hours.* "

This description clearly states that the spatial resolution of the river routing model can be determined by the river segment list file. And the time step can be set by users. Therefore, both the spatial and temporal resolution can be very different to the Noah-MP column model.

4. What are the required input data for the river routing model in addition to those needed by Noah-MP column model?

Reply: "This module requires two additional inputs files, a river segment list file named 'ChannelOrder.txt' and a 'namelist.txt' file for setting parameters." This sentence is in the original manuscript.

5. What are the reasons for the difference between the two models? Theoretically speaking, they should produce exactly the same results due to the same equations and input data.

Reply: We agree with your comment that the models should produce exactly the same results due to the same equations and input data. However, the equations must be implemented by the programming code. The difference between the C# platform and the Fortran platform is complex, while the Noah-MP model is not a simple one but a model with tens thousands of lines, which made it very difficult for us to discern where the difference is coming from. Fortunately, during this revision, based a hard work on analyze the printed variable values, we find two major inconsistences in the code, and then a large part of the output differences is now removed. **The first previous inconsistence is missing the updating of the 'FICEOLD' variable in each time step; The second inconsistence is the wrong parameter translation of SNOWH2O function, due to the ambiguous parameter defining in the original FORTRAN code. See the improvement in Fig.1.**

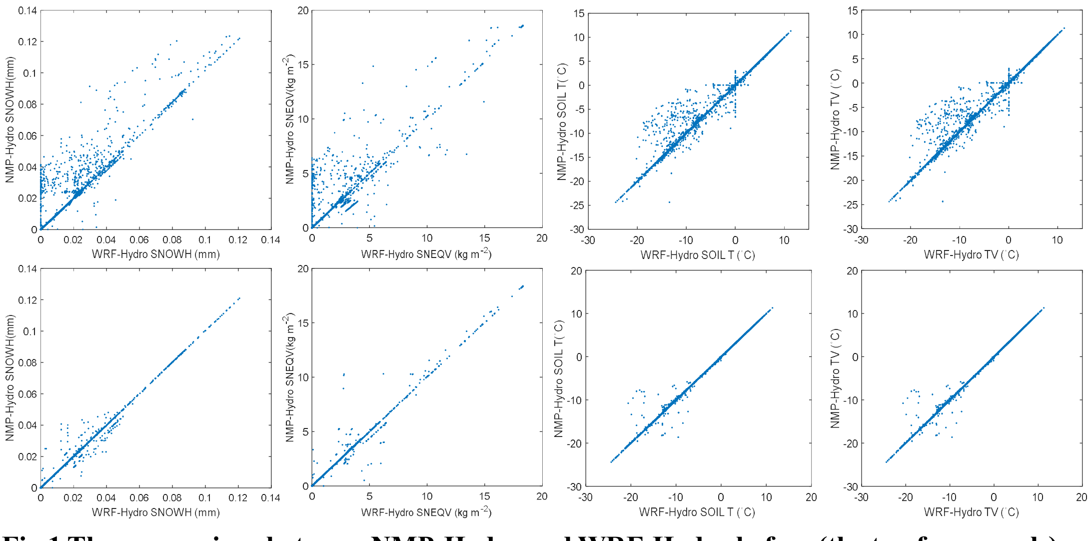

Fig.1 The comparison between NMP-Hydro and WRF-Hydro before (the top four panels) and after (the bottom for panels) this update of NMP-Hydro.

6. What is the difference in the computational efficiency between the C-based and FORTRAN-based model codes?

Reply: We must admit that the program developed by C# language (not the C language in the above misspelling) is usually slower than that developed by Fortran, however C# is not a slow language. Such comparison between the two languages can be seen in many benchmark comparisons on the internet.

According to our experience, for running a time step of non-parallel WRF-Hydro (the original Noah-MP model) and the non-parallel NMP-Hydro (our newly developed model) on our laptops, the former seems taking less time than the latter. But the parallel running of NMP-Hydro is faster than a non-parallel WRF-Hydro. However, benchmarking comparison is difficult for us to made because the two models usually running on different platforms (operation systems) and computers.

Pursuing higher computational efficiency is not in our goal for reconstructing the Noah-MP model. Therefore, we will not present much discussion on this topic. For most cases, the computational efficiency is not a critical issue, because the difference is always small and acceptable.

7. There are a few important state variables that were not compared between the two models, including soil moisture, soil temperature, snow water equivalent, and snow depth.

Reply: There are indeed many important state variables need to concern. **In this revision, we have provided the comparison of soil water content, soil temperature, snow water equivalent and snow depth in supplementary material.** These variables generally show similar effects as the variables that are presented in the main manuscript. Corresponding description was now added to the manuscript.

Due to our limited energy, we cannot provide the tests on all of them everywhere. Consider that these variables are interconnected, the benchmark differences for other variables actually can be indirectly reflected by the results of those presented variables.

8. Lines 250-264: What are the causes for these large differences in runoff, temperature, radiation, and exchange coefficient?

Reply: During this revision, we find two code inconsistences through artificial analyzation on the printed variable values, which removed most part of the differences. However, there is still some large differences, which is related to the difference between the states of no-snow layer and having-snow layer. This snow-layer difference may be caused by some accumulated floating-point error on the depth of snow.

9. Lines 271-273: These differences seem too large for precision errors. Usually, if it is single precision, they would only differ in the 7th digit after the decimal point. Did the authors see any difference between the two models for all output fields in the first few (e.g., 10) model timesteps? If not, then maybe the precision error growth in a longer-period run would contribute to such difference. What model timestep did the authors use? Would reducing the model timestep (i.e., smaller numerical integration errors) lead to more consistency between the two models?

Reply: Thanks for your recommendation to solve the difference issue of the two models.

However, actually, during our efforts of more than five years (Dec. 2018 to now (Sep. 2024)), we have compared the two models in step-by-step running on multiple grid-boxes, only in the first 3-time steps. We find that mostly the differences are very small. We found that there were significant calculation errors (small but cannot be regarded as wrong) after multiple iterations in the VEG-FLUX function (especially in TV and TG calculations), but due to the iteration, it was difficult to figure out whether the error came from because there were many variables in the function and the code was lengthy and iterative. However, according to the final comparison of output results, in fact, both TV and TG have very small errors between the two models.

All the LSM simulation is based on a 3-hourly timestep. We do not believe that reducing the model time step can get better consistency, because the Noah-MP is not a partial differential equation-based model (unlike that in climate models) and there is no numerical integration concept here.

Fortunately, during this revision, based a hard work on analyze the printed variable values, we find two major inconsistences in the code, and then a large part of the output differences is now removed. **The first previous inconsistence is missing the updating of the 'FICEOLD' variable in each time step; The second inconsistence is the wrong parameter translation of SNOWH2O function, due to the ambiguous parameter defining in the original FORTRAN code.**

10. Is there two-way feedback between Noah-MP and the river routing scheme, or is it just Noah-MP affecting river routing results? Did the authors also see difference between the two Noah-MP models without activating river routing scheme?

Reply: There is no feedback from the river routing module to the Noah-MP LSM, therefore, the difference is unrelated to the river routing module.

11. It would be helpful if the authors could discuss a bit the future plans for applying and/or further improving the NMP-Hydro model and potential connection to the broader Noah-MP community.

Reply: According to this recommendation, a new paragraph was added to the 'conclusion' section:

"This new software can *run on Windows system platforms. Its C# code can be analyzed and visually browsed using many modern intelligent tools such as those in Sharpdevelop (https://github.com/icsharpcode/SharpDevelop) or Microsoft Visual Studio. The code of NMP-Hydro is easier to analyze, study and modify, which in turn will attract more users and promote the future development of the Noah-MP model. The current version of NMP-Hydro can serve as a good model for simulating land surface processes in climate change and ecohydrology research. Although NMP-Hydro cannot be coupled with the WRF model, it can still be used as a prototype system of Noah-MP to improve the Noah-MP schemes. Any new improvements in NMP-Hydro can easily be updated to other FORTRAN based Noah-MP. Future plans for the development of NMP-Hydro include (1) investigating whether the remaining differences between NMP-Hydro and the original WRF-Hydro 3.0 are caused by floating-point errors or other bugs in the code; (2) providing a single-column run mode and incorporating a genetic algorithm-based parameter optimization module; (3) extending the functionality for modelling dynamic vegetation by designing new schemes or optimizing parameters."*

We have revised the manuscript in a deeper extent, according to your comment.

Here, we upload a supplement file, but the revised manuscript is not allowed to upload here (this is required by the system. I do not understand why). Maybe we can upload the revised manuscript in a later chance.
Thank you very much.

Yonghe Liu