Comment on gmd-2022-135

Anonymous Referee #1

Referee comment on "Modular Assessment of Rainfall–Runoff Models Toolbox (MARRMoT) v2.1: an object-oriented implementation of 47 established hydrological models for improved speed and readability" by Luca Trotter et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2022-135-RC1, 2022

The manuscript here presents a new version of MARRMoT with significant changes in structure and solver. The new version implemented some good programming practices of modular, object-oriented structure to improve code readability and debugging as well as to reduce the repetitiveness of the code. Benchmarked again the previous version, the new structure and solver seem to be stable and consistent for most of the sub-models (30 out of 36). Overall, the improved speed and readability in the new version echo the efforts toward reproducible and transparent research and coding.

The manuscript is generally concise, organized, and well-written. I don't have major concerns about the manuscript but some minor comments that I hope to help the manuscript clarify a few points and potentially reach a broader audience and a larger number of users.

1, Line 205, for new users or those who are not familiar with the MARRMoT, what is the model spin-up process, what are the spin-up criteria (discharge, water storage / soil moisture), and does the model automatically handle the repeating of climate forcing data?

2, does the model has the "hot-start" ability to continue running the model using saved model outputs from the last run?

3, it was not very clear whether the model comparison was made by comparing all NSE from 559 catchments or the median NSE? For example, does 0.29 mm/year the average difference for all 559 catchments?

4, Figure 3, what does the label (subscript) mean, 5p_5s, 8p_3s, it might confuse readers.

5, Figure 5, the ratio is easy to show speed improvement compared to the previous version. But what were the computational time and the time difference compared to their original counterparts that wrote in C or Fortran?

6, Line 300, I agree that adapting time-stepping schemes is critical and it might further improve speed and efficiency.