

Interactive comment on “Rapid development of fast and flexible environmental models: The Mobius framework v1.0” by Magnus Dahler Norling et al.

Magnus Norling

magnus.dahler.norling@gmail.com

Received and published: 20 July 2020

* The goal was not to compare manual to autocalibration, but to show one possible workflow when calibrating the model and iterating on model structures (i.e. use autocalibration to "fine-tune" a manual calibration). We are not the authors of the autocalibration algorithms, and so the point is not to show off how good these are at reaching good solutions from random starting points. We are only showing one possible way these python libraries can be used with models built in Mobius.

* Future edits to the github repository should not break the experiments described in the paper. However, we also made an archived version of the repository corresponding

C1

to this paper, which is linked to in section 5 (line 376).

* The source files for the benchmarking are referenced in the paper, line 335 "Code used in these experiments can be found in the "Evaluation" subfolder of the Mobius repository"

* Description of the hardware used in benchmarking can be provided in the revised manuscript. Note that we only report the ratio between the run times of different implementations. The run times themselves, are highly hardware dependent, while the ratios are fairly stable across several different machines we tried. Number of cores is not going to be that relevant since one model run is single threaded (while you can parallelize if you want to run the model many times such as in MCMC).

* I'm not sure what precisely is asked for in the last comment. Optimization times are highly variable, depending on algorithm, the model used, input data etc. These together determine how many model evaluations are needed to reach convergence, and are independent of the model implementation (as long as the implementations give identical results). The model implementation only determines how fast each model evaluation is, and so that is what we focused on in the benchmarking. Timing for optimization will scale accordingly. Nevertheless, we did provide the timing for one particular optimization run on line 287 (using one specific algorithm and model setup).

Interactive comment on Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2020-26>, 2020.

C2