GMD submission by Knoben et. al.

Modular Assessment of Rainfall-Runoff Models Toolbox (MARRMoT) v1.1: an open- source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations

General response

We thank the editor and reviewers for their consideration of our manuscript and the obvious care with which the reviewers have scrutinized our work. Their comments are valuable and encouraging.

Their main points can be summarized as follows:

(1) Correspondence between MARRMoT models and original models is not sufficiently addressed;
(2) Table 1 should be appended to include additional information that specifies how we changed models from their original publication, and for what temporal resolution the models were originally intended;
(3) There is a possible mistake in the model smoothing code, resulting from a typing error in the source of the smoothing equation;
(4) There is an omission of code in the IHACRES model which is not explained in our documentation;
(5) There are various typing errors and sentences in need of clarification.

We discuss the reviewers' comments in the remainder of this document. Our responses are given in blue, and changes in the text are indicated in **bold** where relevant. Line numbers in our responses refer to the 'track changes' document.

We also fixed various spelling mistakes in the Supplementary Materials. These are not explicitly mentioned in the following comments.

The reviewers have highlighted one inaccuracy and one omission in the MARRMoT code. We have adapted the code following their recommendations and released MARRMoT version 1.1 which includes these changes. The manuscript title and DOI's have been updated to reflect this.

Kind regards,

On behalf of all co-authors,

Wouter Knoben

Reviewer 1 - P. Kraft

General Comments

(1) **Reviewer comment:** The authors have completed a great task: translating 46 model structures into a clean system, where equations and solver are separated and suitable to be used with an implicit solver scheme is a great accomplishment. I tried quite a while ago something similar for only a few models with CMF, which has a similar base structure as MARMMoT. I gave up to mimic existing models, since the abundant mixture of model code, flux equation and ad hoc solution schemes in existing models makes it extremely difficult and tiring to translate them into a clear set of ordinary differential equation. This translation of existing models into a common scheme is a new feature of MARMMoT that is not available for more abstract model building frameworks like SUPERFLEX, CMF and SUMMA.

Author response: Thank you for these kind words. To clarify, we created the MARRMoT models by only using model description papers or user manuals where papers where unavailable. We have not used any "original" model code to base our models on.

We note that reviewer comments 2 to 6 can all be related back to this point: we chose to base all MARRMoT models on written documentation only, not on existing computer code. Our reasons for doing this are as follows: (1) written documentation is traceable through the cited sources, which allows MARRMoT users to compare our documentation and code to the original work. (2) Computer code is often not available, which is a practical constraint on our ability to use existing code. (3) Multiple different versions of a certain models can be found (that still use the same name), with limited or no traceability or version control. This makes it difficult to decide which set of computer code can be considered the 'original' model. Therefore we rely on published documentation only.

We have introduced a new section 2.1 Scope (in response to reviewer comment 9), which includes the following clarification:

P4 L 22: "**MARRMoT models are based on written documentation only, not on existing computer code. This choice is motivated by our aim to produce traceable code and by several practical concerns. The documentation we base our models on is traceable through our cited sources. Computer code of hydrologic models tends to be less traceable than their documentation: code might be unavailable, code might not be accompanied by a persistent identifier, or multiple versions of the same model (using the same model name) might be available which complicates finding the 'original' computer code. This is supported by various authors who developed the original models: "Today many versions of the HBV model exist, and new codes are constantly developed by different groups …" (Lindström et al., 1997) and " … TOPMODEL is not a single model structure […] but more a set of conceptual tools" (Beven et al., 1995).**"

(2) **Reviewer comment:** For this reason, I would be very happy to see this study published and I agree with the authors concerning the great potential of such a unified model collection for future studies. However, for future applications, the users must know how much the newly constructed models really resemble the original work. The authors state, that they needed to make assumptions, changed processing orders and smoothed discontinuities to make existing models fitting into the new structure, but the discussion about the effects of these changes in chapter 5.3.3 is shallow and not covered by data. If I use "m_37_hbv_15p_5s", how similar are the results compared with the original HBV-96?

Author response: We agree that section 5.3.3. does not contain any comparison of "original" models and our MARRMoT models but think our intent of this section might not have been sufficiently clear. We think that such a comparison is impossible to make, because for many of our models no original code is available, and for other models too many different versions (all with the same model name) can be found online and at institutes. From personal experience, I have worked with three models that all claim to be (based on) HBV-96 but all three were certainly different. Hence we have not made any comparisons between MARRMoT models and other models inspired by the same original publication (more on this in response to the reviewers next paragraph). Instead, we intended section 5.3.3. as a caution against the assumption that our models are the same as other sets of code out there: our models will be different from any other model codes that are inspired by the same source material, and only by studying both the original papers and our MARRMoT implementation of such papers do we expect that users can fully understand why our code looks the way it does. This is currently covered in section 5.3.3:
P13 L5: "We strongly recommend readers to compare the original publication of each model with the version given in this toolbox, to place results from the MARRMoT models in a proper context of earlier work with these models."

We emphasized this caution in section 5.3.3., and the differences we introduced between original publications and MARRMoT models will hopefully be much clearer in the revised manuscript, where we now address these changes in Table 1 like the reviewers suggested. Changes to section 5.3.3.:
P13 L 7: **"We emphasize that our models are based on publications that describe existing models, not on existing computer code. Thus, we neither guarantee nor expect that our code performs exactly like the original version of each model's code (if indeed such a version exists and can be found and agreed upon for any given model). We hope that by making MARRMoT available as open source code, future studies can go beyond simply stating the model name without publishing any model code, and instead can refer to an open-source, traceable version of the model(s) used."**

We have further changed Figure 2, so that the header of the model column now reads "Original model that is the basis the MARRMoT implementation" instead of "Models".

(3) **Reviewer comment:** What kind of quality control did you use to ensure the correctness of the translation? From my experience with abstract model formulations in CMF even extremely small changes can lead to suprising strong changes of the overall behavior, therefore I deem a more detailed discussion on the effects needed for a better article.

Author response: We acknowledge that small changes in model structure or code can have large impacts on model behaviour. However, we also think that it is practically impossible to track down a version of each model where we can confidently claim that that bit of code is indeed the original code of that model. This is supported by various authors who developed the 'original' models, who state things such as "Today many versions of the HBV model exist, and new codes are constantly developed by different groups …" (Lindström et al., 1997) and " … TOPMODEL is not a single model structure […] but more a set of conceptual tools" (Beven et al., 1995). Moreover, even if it is possible to locate a version of each model that can be considered the true original, the code might not be available (any longer).

We currently provide a test case example of MARRMoT model performance (section 4 in the paper) but we believe that measuring this against a baseline of "original" models is unfortunately practically impossible. We have clarified that MARRMoT models are based on documentation only (not on computer code) in the caption of Table 1 and section 5.3.3, to clarify that we do not possess the necessary computer code for more in-depth comparison. We have also (briefly) summarized the

discussion of this point in section 5.3.3 (see our response to comment 2). Changes to Table 1 caption:

P30: "Table 1: MARRMoT models. **Model IDs are used throughout this paper and the MARRMoT documentation. MARRMoT function names include a longer identifier that either refers to the name of the original model (e.g. m05_ihacres_7p_1s) or to the area of original application (e.g. m_01_collie1_1p_1s which was used in the Collie River basin). The column "Main changes" specifies structural changes between the MARRMoT model and the original model description (note that MARRMoT models are created solely based on the cited sources and not on any computer code). Not mentioned are cases where (i) model equations needed to be modified to account for the time step size at which the model is used; (ii) Ordinary Differential Equations were not given in the original source; (iii) cases where modelled processes were only described qualitatively in the original source, without equations; (iv) cases where model equations where smoothed in their MARRMoT implementations (these can be traced through the overview of flux equations in Supporting Materials S3).**"

(4) **Reviewer comment:** The perfect solution would be to include a graph of RMSE (MARMMoT vs. original model result) for good parameter sets. If this requires too much work, I would at least expect such a comparison for 2 or 3 strongly changed models and for 1 or 2 lightly changed models in combination with an additional column in table 1, that indicate the deviation from the original model code for models.

Author response: We have adapted table 1 to include the changes we introduced between the MARRMoT version of each model and the description we base these models on (see our response to comment 3).

A full comparison of all MARRMoT models and their originals is indeed out of reach for the following reasons:

- As mentioned earlier (see our responses to comment 1-3), we have not used any computer code to create MARRMoT models for a variety of reasons. Finding the "official" version of all 46 MARRMoT models is practically impossible, which limits which MARRMoT models can be compared to "original" models;
- Several models can be freely downloaded but do not publicly share their source code. This allows us to compare the performance of these models with their MARRMoT equivalent, but still does not allow us to judge how well MARRMoT approximates the original documentation (i.e. the reviewer's concern is now applied to the downloaded model's code: we cannot be sure that the model's internal workings reflect its documentation);
- The fact that the numerical solving scheme used is not often mentioned in source documentation complicates this comparative analysis. A common solving scheme in hydrology is Explicit Euler, but we use an Implicit Euler scheme as the preferred option in MARRMoT. Even with the same model, using a different numerical approximation scheme will generally lead to different simulations. Because the scheme used in original models is generally not clarified, it is difficult to judge how similar we expect our MARRMoT model simulations to be to any simulations generated by "original" models.

To illustrate this point, we follow the reviewer's suggestion and compare a lightly changed model (MARRMoT m37, based on HBV-96) and a strongly changed model (MARRMoT m07, based on GR4J) with "official" alternatives (HBV Light and the R implementation of GR4J called airGR respectively). We've updated section 5.3.3. with three additional paragraphs and included a new Figure 4 that supports this comparison. Additions to 5.3.3:

P13 L 10: “We strongly recommend readers to compare the original publication of each model with the version given in this toolbox, to place results from the MARRMoT models in a proper context of earlier work with these models. **We emphasize that our models are based on publications that describe existing models, not on existing computer code. Thus, we neither guarantee nor expect that our code performs exactly like the original version of each model’s code (if indeed such a version exists and can be found and agreed upon for any given model).**

To illustrate this point, we compare performance of MARRMoT model m07 (based on the GR4J model) with the R implementation of GR4J (part of the airGR package; Coron et al., 2017, 2019), and we compare MARRMoT model m37 (based on HBV-96) with HBV Light (Seibert and Vis, 2012). MARRMoT m07 is an example of a model that has changed significantly from the original source as a result of combining the original documentation (Perrin et al., 2003) with a more recent state-space version of GR4J (Santos et al., 2018), while both MARRMoT m37 and HBV Light are similar to HBV-96. We thus expect larger deviations between simulations from MARRMoT m07 and airGR-GR4J than we expect between simulations from MARRMoT m37 and HBV-Light. In both cases, we selected 10000 parameter sets from MARRMoT’s parameter ranges through Latin Hypercube sampling. In the case of GR4J, both MARRMoT and airGR versions use the same 4 parameters. In case of HBV, the MARRMoT version has several additional snow parameters and a capillary rise parameter, while HBV Light has various elevation and input correction factors. These have all been fixed at values that effectively disable their impact on model simulations. We then simulated 5 years of streamflow in the earlier described Hickory Creek using both versions of both models. For comparison purposes, we use the Kling-Gupta Efficiency (KGE; Gupta et al., 2009) to express the similarity between simulations and observations. Error! Reference source not found. **shows the results of this comparison.**

Figure 4a shows that for the best performing parameter set in our sample (in terms of KGE value), the hydrographs generated by MARRMoT m37 and HBV Light are relatively similar. Figures 4c-4e show a decomposition of KGE values into its three constitutive components, that express the linear correlation (KGE_r), the ratio of simulated and observed standard deviations (KGE_a) and the ratio of simulated and observed means (KGE_b) respectively. For a given parameter set, MARRMoT m37 and HBV Light generate simulations that are relatively similar (i.e. close to the 1:1 line). HBV Light tends to produce more variable flows than MARRMoT m37 does (high standard deviation and mean of simulated flows). The reason for this is difficult to investigate because although HBV Light is freely available, its source code is not. Differences between both models’ equations and numerical approximation of these equations are likely explanations.

Figure 4b shows that for the best performing parameter set in our sample (in terms of KGE value), the hydrographs generated by MARRMoT m07 and airGR-GR4J are relatively different. Most notable, MARRMoT m07 recessions are much slower and higher than those from airGR-GR4J. Figures 4f-4h indicate that for parameter sets close to the optimal points (i.e. (0,0)), MARRMoT m07 and airGR-GR4J show similar performance. For parameter sets further away from the perfect simulation, MARRMoT m07 shows an increasing tendency to simulate more variable flows (higher standard deviation and mean components) than airGR-GR4J does. However, differences between MARRMoT m07 and airGR-GR4J are not unexpected because MARRMoT m07 also uses equations from state-space GR4J (Santos et al., 2018) and the models’ equations are thus not identical.

Concluding, we emphasize again that MARRMoT models are based on existing publications only and not on computer code. Differences with other models using the same name are unavoidable. We hope that by making MARRMoT available as open source code, future studies can go beyond

simply stating the model name without publishing any model code, and instead can refer to an open-source, traceable version of the model(s) used."

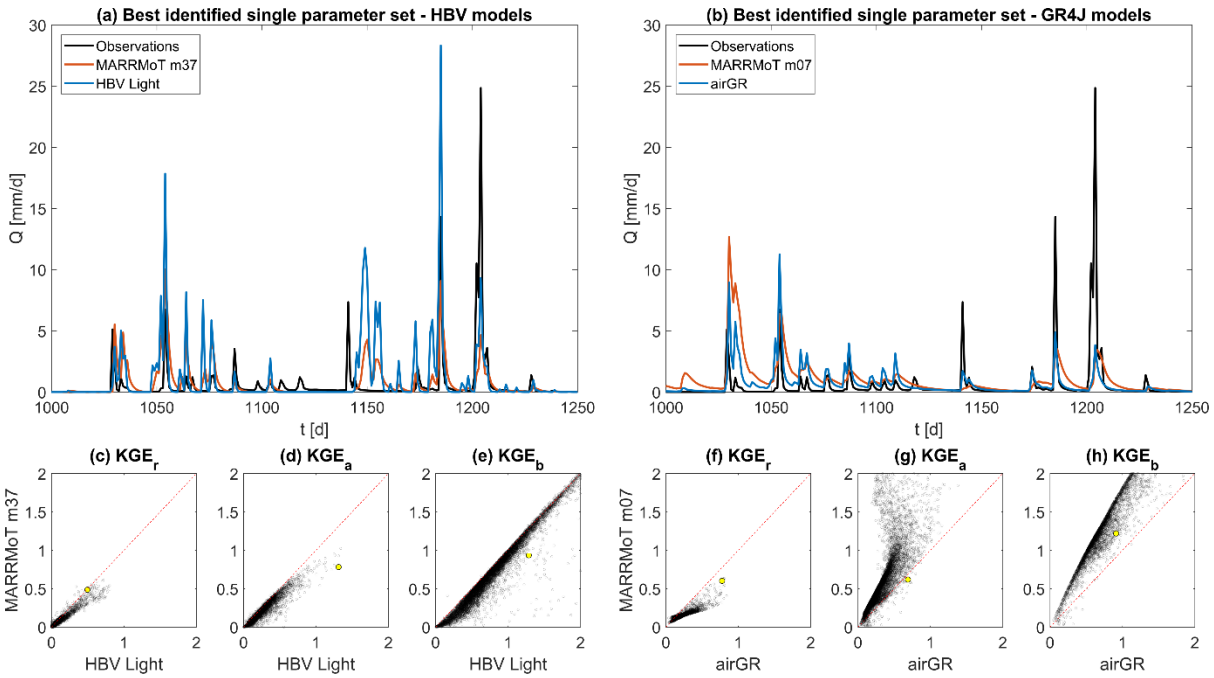

Figure 4: Comparison of two MARRMoT models and freely available model codes based on the same source material. (a) Close up of hydrographs generated by MARRMoT m37 and HBV Light using the same parameter values for their shared parameters. (b) Close up of hydrographs generated by MARRMoT m07 and airGR-GR4J using the same parameter values. (c-e) Constitutive components of the Kling-Gupta Efficiency (KGE) obtained by HBV Light and MARRMoT m37 for 10000 parameter sets in a single catchment. The yellow dot indicates the parameter set used to generate figure a. (f-h). Constitutive components of the KGE obtained by airGR-GR4J and MARRMoT m07 for 10000 parameter sets in a single catchment. The yellow dot indicates the parameter set used to generate figure b.

(5) **Reviewer comment:** Another concern that I have stems from the following: Deviations between equation implementations in code and their published version in "math" can easily differ – the authors suffered from this problem themselves. Hence I would be very interested if they found differences between model publication and implemented model code in their list of original models and how they dealt with such differences.

Author response: Unfortunately we cannot comment on this particular concern, because we did not use any existing computer code to inform MARRMoT modelling choices.

(6) **Reviewer comment:** And finally, what kind of quality control measures they took, to ensure that their implementation is in fact equivalent to the original implementation und does not differ strongly by new bugs or the correction quirks from the original model.

Author response: Because our MARRMoT code is only based on publications and not on examples of model code, no such quality control measures where possible. We cannot guarantee that no bugs are present in our framework. However, if bugs are present, our framework assures that each model that relies on the bugged element suffers from the same bug or quirk. This at least makes the comparison between models within MARRMoT fair. This is already mentioned as an important aspect of this work in the introduction:

P3 L30: “Due to the code being open source, transparency and repeatability of research is encouraged, additions to the framework are possible, and the community can find and correct any mistakes.”

And also in section 3.1 (relevant section emphasised):
P8 L7 “Flux functions are kept separate from the model functions, and each model calls several flux functions as needed. This allows for consistency across models (if errors are present in any flux function, at least they are the same in all models), easy implementation of new flux equations and facilitation of studies that are specifically interested in differences between various mathematical equations that all represent the same flux or process.”

Every model function has a built-in water balance check (output is reported on user request). This has been used to test each model’s water balance accounting during development and no errors were found. We’ve included this information in section 3.2:
P8 L24: “**The model code as currently provided was extensively checked for water balance errors during development, using multiple parameter sets for each model, both randomly sampled and using all combinations of extreme values using MARRMoT’s provided parameter ranges. These errors were generally in the order of 1E-12 or smaller, showing that the water balance is properly accounted for in each model.**”

Specific Comments

(7) **Reviewer comment,** P3 L 14: FARM does not fit into this listing, please remove

Author response: Agreed upon re-reading the paper, removed from the listing.

(8) **Reviewer comment,** P3 L 18: CMF falls for this comparison rather into the same category as SUPERFLEX, since you can build conceptual models as well as physical models and things in between. A reference for a “SUPERFLEX”-like usage of CMF is Jehn et al. 2018 https://doi.org/10.5194/hess-22-4565-2018

Author response: Thank you for this correction. We currently don’t list any example applications of the cited model comparison frameworks but will re-visit Jehn et al (2018) in follow-up publications that use MARRMoT. Changes:
P3 L 17: “… models (e.g. **CMF**, SUPERFLEX), or …“. Removed CMF from P3 L18.

(9) **Reviewer comment,** P3 L 32: The benefits of MARMMoT are explained a bit too enthusiastic – especially the “best practice” part about the solvers, in comparison to the discussion on that topic. I would also expect a clear note about the boundaries of MARMMoT’s scope (eg. only lumped models, no internal ET calculation etc.)

Author response: This is fair. We have toned down the language somewhat (“best practices” are now “good practices”) and added a new subsection to section 2 that outlines the scope and limitations of MARRMoT. Changes:
P1 L12: “several **good** practices **of model development**: …”
P3 L26: “… **good** practices for numerical model solving …”

P4 L9: "several other **good** practices for model development …"
P7 L2: "following the **good model development** practices outlined …"
P4 8: "… framework (Clark et al., 2008). **Section 2.1 gives a brief outline of the project scope and design philosophy.** MARRMoT follows several other **good** practices for model development which are briefly described **in sub-sections 2.2 to 2.5**.

2.1 Scope
MARRMoT's scope is limited to conceptual hydrological models and the code currently includes no spatial discretization of inputs or catchment response. Models are expected to be used in a lumped fashion, although users could create their own interface to use MARRMoT code to represent within-catchment variability using multiple lumped model structures. Required model inputs are standardized across all MARRMoT models and every model only requires time series of precipitation and potential evapotranspiration, and optionally of temperature (used by certain snow modules). Model outputs are equally standardized and provide time series of simulated flow and total evaporation fluxes, and optionally time series of model states and internal fluxes. The models are set up such that they can use a user-specified time step size (e.g. daily, hourly) which is currently effectively the temporal resolution of the forcing data. Models and flux equations internally account for this time step size, so that parameter values can use consistent units, regardless of the temporal resolution of the forcing data. The main goal of this set up is ease-of-use, so that it is straightforward to switch between different model structures within an experiment."

(10) **Reviewer comment,** P5 L 5: Implicit schemes can fail if the time step size is too large for the non-linear solver to converge. How does the solver in MARRMoT deal with this? Is there an internal dynamic time step?

Author response: The internal time step of each model is equal to the temporal resolution of the forcing data. A user can choose (inside each model function) to turn on a progress display and messages from the solver which will indicate whether such a problem has occurred. Adaptive sub-stepping can resolve this issue, but this is currently not implemented (see section 5.3.5. Possible extensions). We've updated the text to clarify this. Changes:
P5 L29: "Note that fixed time step size refers to the use of a single time step size throughout a simulation **(i.e. no adaptive sub-stepping is used; see section 5.3.5)** and does not prescribe the time step size **(e.g. hourly, daily)**."

(11) **Reviewer comment,** P5 eq. 2: The equation is wrong, must be changed to $Q_o = Q_i\left(1 - \phi(S, S_{max}, \rho_s, \varepsilon)\right)$
(see source code: eg. infiltration_3.m:21,23,25 interflow_11.m:23,25,27), otherwise the names inflow and outflow do not make sense. An adhoc implementation of this equation shows that the parameter $\varepsilon = 5$ does not prevent $S > S_{max}$. A longer discussion on that can be found at this gist: https://gist.github.com/philippkraft/aae02d23fbdad62f98a413ab04fe6d83

Author response: Thank you for this in-depth analysis. We agree with your assessment and have changed both the text in the manuscript and the smoothing code. We think this is a code change of sufficient magnitude to warrant a new release and have incremented MARRMoT's version number to 1.1. The code DOI's in the manuscript have been changed in response. Text changes:
P6 L11: $Q_o = Q_{in}\left(\mathbf{1} - \Phi(S, S_{max}, \rho_S, \varepsilon)\right)$

(12) **Reviewer comment,** P 11 L 7: Implicit solvers are usually error controlled. Which kind of tolerances (relative and absolute) are used in the solver? And how does the solver react, if a solution within the error boundaries is not found? I understand the text, that at least fsolve can return values with an unspecific error tolerance (I guess in situations where some convergence criteria are missed)

Author response: All solvers (*fzero*, *fsolve*, *lsqnonlin*) use default settings in our example workflows (default accuracy is 1E-6 mm function values, which equates to an accuracy of 1E-6 mm in our case), but users can easily choose other options if they wish. All solvers output a squared norm of residuals ("resnorm"), which is the value of the objective function at the solver-provided solution. The correct solution is found when resnorm = 0. MARRMoT contains basic error control in the form of user-specified tolerance ("resnorm_tolerance" in our workflow examples). If the "resnorm" of a given solution is larger than the user-specified tolerance, various actions can be triggered:

First, the model function uses a more robust solver (*lsqnonlin*) to see if that succeeds where the faster solver (*fzero* or *fsolve*, depending on number of stores) failed. This happens inside a sub-function called "rerunSolver". Next, within "rerunSolver" *lsqnonlin* attempts to solve the model equations at this time step for a user-specified number of times ("resnorm_maxiter" in our workflow examples). Each attempt is started from different initial guesses:
(1) zero storage values;
(2) very high storage values (beyond the store maximum) ;
(3) storage values where *fzero* or *fsolve* got stuck;
(4) storage values of the previous time step;
(5) maximum storage values (if provided);
(6) random storage values.

If no solution is found, the "rerunSolver" function outputs an error flag (which the user can check for) and uses the last found (non-optimal) solution.

We have updated the text in the manuscript with a general note on where to specify solver settings and updated the User Manual with a more in-depth description:
P12 L15: "**Note that settings for these root-finding methods are specified within each model file because certain settings are model-dependent. Progress display is disabled for all three functions (*fzero*, *fsolve*, *lsqnonlin*) by default but can be enabled by the user. The model-dependent Jacobian matrix is specified for *fsolve* and *lsqnonlin*. The maximum number of function evaluations is capped at 1000 for *lsqnonlin*. All other root-finding options are left at default Matlab values (see Matlab documentation of the root-finding methods for further details). Users are encouraged to experiment with these settings to find those that work for their specific problem.**"

User Manual, P17: **"The function "rerunSolver" will attempt to find new solutions for the current time step that are within the accuracy threshold specified in "solver.resnorm_tolerance". It does this up to "solver.resnorm_maxiter" times, and restarts the solving procedure from different initial guesses each time. This provides better chances of finding a solution with the requested accuracy.**

Currently, two optional output arguments of "rerunSolver" are unused. Output argument 2 provides the final value of "resnorm" which the user can request and check to see whether the accuracy specified in "solver.resnorm_tolerance" has been achieved. Alternatively, the user can request output argument 3 ("flag") which returns 0 if the function "rerunSolver" returned a sufficiently accurate solution. "flag" will return -1 if "rerunSolver" has not been able to find a sufficiently accurate solution."

(13) **Reviewer comment,** P 11 L 23: Since the comparability of the MARMMoT functions with the original model is the major feature of this study, I would expect a discussion on this topic that is deeper, completer and more explicit. See general comment.

Author response: Answered in responses to comments 2-6 above.

(14) **Reviewer comment,** Table 1: Add columns that indicate the type and level of deviation between the original source and the implementation in MARMMoT (eg. spatial generalization, change of solution order, introduction of smoothing functions, generalization of timestep, etc)

Author response: We have amended Table 1 quite significantly based on both reviewers' comments, and this change has been made. See pages 30 and 31 in the revised document.

Reviewer 2
Anonymous (Referee)
Received and published: 2 April 2019

(15) **Reviewer comment:** This is an interesting and well-written paper. It is a timely contribution to making available open, flexible and easy-to-use platforms for hydrological modelling. One strength and originality of the platform is the use of space-state model formulations and a well adapted solver. The Supporting material also gathers a wealth of information, very useful for young (and less young) modellers wishing to better understand models' behaviour. I have only a few minor comments detailed below. I advise publication after minor revision and I congratulate the authors for this impressive amount of work.

Author response: Thank you for these kind words. Please find answers to your detailed comments below.

Detailed comments

(16) **Reviewer comment,** P1, L4: Should not it be "state-space"? This aspect, which appears in the tittle, is not really discussed in the text. Maybe the authors could explain a bit more the implications of considering state-space formulations.

Author response: Thank you, state-space is indeed correct. Changed in the title. We've also added a clarification to the text to point out where state-space formulations are discussed (this discussion was already there, but not clearly labelled as such). Changes:
P5 L2: "First, MARRMoT uses a distinct separation of model equations **as state-space formulations** and the numerical approach used to solve these equations."

(17) **Reviewer comment,** P2, L5-8: A number of earlier papers had discussed the issue of modelling steps and may be cited if deemed useful (Refsgaard and Henriksen, 2004; Refsgaard et al., 2005; Scholten et al., 2007).

Author response: Thank you for these recommendations. We consider especially the first paper (Refsgaard and Henriksen, 2004) to be useful for our model development context and have included it as a reference. Changes:

P2 L8: "(e.g. Beven, 2012; Clark and Kavetski, 2010; Gupta et al., 2012; **Refsgaard and Henriksen, 2004**)"

(18) **Reviewer comment,** P4, L9: Move "(ODEs)" to line 15.

Author response: Agreed, changed (now on P5 L8 due to the new section 2.1).

(19) **Reviewer comment,** P7, L7-10: Is it possible with the tool to apply a model on a set of catchments? This would be useful in the perspective of model testing on large sample.

Author response: Yes, that would require only a simple loop that loads data for various catchments and sends this to the model(s). We have amended the text to clarify that there are no barriers to using MARRMoT in multiple catchments:
P8 L5: "**These examples can easily be adapted to work with multiple catchments if desired.**"

(20) **Reviewer comment,** P7, L30: I was surprised that PDM, which is widely applied in the UK and elsewhere, is not part of the platform.

Author response: The core of PDM is a certain set up of the soil moisture routine, which is currently included in MARRMoT as part of the HyMOD model (MARRMoT ID: 29). The nature of this project is such that it is practically impossible to include all models, but an interested user should be able to start with our HyMOD code and (with help from the User Manual) modify this model to more closely resemble PDM. The option to do this is currently described in the introduction, section 2, section 3 and section 5.3.5 (relevant sentences emphasised):

P3 L30: "Due to the code being open source, transparency and repeatability of research is encouraged, additions to the framework are possible, and the community can find and correct any mistakes."

P4 L7: "and (ii) add new options to the framework"

P8 L 17: "User Manual: This document helps a user set up MARRMoT for use in either Matlab or Octave, outlines the inner workings of the standardized models, provides several workflow examples and provides examples on how to create a new flux equation or model."

P14 L22: "The MARRMoT User Manual therefore provides detailed guidance on creating new model and flux functions, and the code's location and licensing on Github allows these new models to be shared freely. Extensions to the framework are thus possible and encouraged."

(21) **Reviewer comment,** P9, L1-11: Actually, these findings are not really new and corroborates past studies in the literature which could be cited.

Author response: This is true, we have added a sentence to clarify that these results are only meant to illustrate what can easily be done with our framework and are not intended to be taken as original findings. Changes:
P10 L6: "**Note that our findings in this test case are not new, but the** test case highlights the power of multi-model comparison frameworks:"

(22) **Reviewer comment,** P9, L26: The sentence was not fully clear for me.

Author response: We've added clarification to this sentence. Changes:
P10 L27 (please note that this change is part of the citation field and these changes somehow do not show with the standard 'track changes' mark-up): "Furthermore, this toolbox lowers the threshold for model comparison studies and can help to diminish "legacy" reasons for model application **(i.e. choosing to use a certain model for reasons other than the model's perceived appropriateness for the task at hand, such as convenience or past experience;** Addor and Melsen, 2019)."

(23) **Reviewer comment,** P12, L22: I am unsure that this would be straightforward. Adaptive time-stepping means that model parameters are not time dependent, which is not always the case (?).

Author response: MARRMoT already allows a user-specified time step size (e.g. daily, hourly). This ability should transfer directly to adaptive time-stepping. This is clarified in the new section 2.1. Changes (relevant section emphasized):
P3 L 11: **2.1 Scope**
MARRMoT's scope is limited to conceptual hydrological models and the code currently includes no spatial discretization of inputs or catchment response. Models are expected to be used in a lumped fashion, although users could create their own interface to use MARRMoT code in a semi-distributed way. Required model inputs are standardized across all MARRMoT models and every model only requires time series of precipitation and potential evapotranspiration, and optionally of temperature (used by certain snow modules). Model outputs are equally standardized and provide time series of simulated flow and total evaporation fluxes, and optionally time series of model states and internal fluxes. The models are set up such that they can use a user-specified time step size (e.g. daily, hourly) which is currently effectively the temporal resolution of the forcing data. Models and flux equations internally account for this time step size, so that parameter values can use consistent units, regardless of the temporal resolution of the forcing data. The main goal of this set up is ease-of-use, so that it is straightforward to switch between different model structures within an experiment."

(24) **Reviewer comment,** P23: The use of "unnamed" for many models is not informative. Could the authors give more explicit names, for example by using the first letters of the first author's name of the cited publications?

Author response: We have amended the header in this table to read "Original model name" to highlight that these are not intended as names of the MARRMoT models (we'd prefer those to be referred to by either their function name (column 3 of this table) – or their ID (column 1)). We've made the following change to the table caption to clarify this:
P30: "Table 1. MARRMoT models. **Model IDs are used throughout this paper and the MARRMoT documentation. MARRMoT function names include a longer identifier that either refers to the name of the original model (e.g. m05_ihacres_7p_1s) or to the area of original application (e.g. m_01_collie1_1p_1s which was used in the Collie River basin). […]**"

(25) **Reviewer comment,** P23: Many models are not using a snow module, but could actually be used with such a module. To which extent snowmelt modules existing in other models could be used with these models?

Author response: This is possible and should be fairly straightforward. The User Manual contains guidance on adapting existing models (and creating new ones). This is currently mentioned in Section 3, section 3.2 and section 5.3.5 (see also our response to comment 20).

(26) **Reviewer comment,** Supplementary material, S2: A few models (e.g. S1, S3, S6) were not initially developed for short time steps (daily or shorter, as mentioned in P5,L5 of the article) and may be not directly applicable at these time steps. Typically, I am unsure a bucket model alone would perform well on most catchments at the daily time step. Should not this be clarified somewhere? Maybe the information on the original model time step development could be added in Table 1.

Author response: We have amended Table 1 quite significantly based on both reviewers' comments, and this change has been made (P30).

(27) **Reviewer comment,** Supplementary material, S2: When reading this document, I found it would be useful to have a summary table for each model showing all model parameters together (symbol, meaning, unit). Some model descriptions are quite long and this table would ease the overview on model parameters. The authors may consider adding these tables, except if it is too much work.

Author response: Thank you for this suggestion. We have added the requested tables to each model description. These changes to the Supplementary Materials are too numerous to copy here.

(28) **Reviewer comment,** Supplementary material, S2: Some models (e.g. #25 or 40 and maybe others) compute net rainfall as the difference between raw rainfall and potential evapotranspiration. This is actually equivalent to having an interception store with null capacity. Therefore I think these models should appear as having an interception store in Fig. 2. This process may also appear in S3 as an interception process.

Author response: Currently this type of behaviour is already covered by our flux "interception_2". We have slightly changed the description of this flux in S3 to reflect that it can serve as both an abstract interception store (i.e. when a fixed amount is taken from incoming precipitation) and as a null capacity store (i.e. when a variable amount is taken from incoming precipitation based on current potential evapotranspiration values). Changes:
Supplementary P127: "Interception excess after an **absolute** amount is intercepted"

With regard to Figure 2, we have tried to keep our models as close to the original documentation as possible. In the source material for model 25, no mention is made of this effective rainfall representing interception. Therefore we do not think that we should assign this interpretation in Fig. 2 to our MARRMoT version (it might also represent surface depression storage or a form of precipitation bias correction). However, model 7 uses this same structure of effective P and the source documentation does mention that this represents interception with null capacity. We had erroneously not labelled this model as such in Fig. 2, which we have now corrected. Changes:
P25, Fig. 2: added an interception marker to model m07

(29) **Reviewer comment,** Supplementary material, S2.1: This bucket model is also often used to represent interception (with evaporation at the potential rate), not only soil moisture. But this is generally only a part of a model.

Author response: True, as evidenced by several other models that use this concept. Although such interception modules generally treat evaporation as occurring at the potential rate for all store depths, and the model in S2.1 uses a linearly decreasing evaporation ratio instead. Therefore we have decided not to mention this particular point in the description in section S2.1.

(30) Supplementary material, S2.5: line 8: "This"

Author response: Changed, thank you.

(31) **Reviewer comment,** Supplementary material, S2.5: The original IHACRES model includes a pure time delay, which is very useful for model applications on large catchments. Why was it removed here? I guess it would be useful also in other models which are not able to introduce a delay between rainfall and streamflow.

Author response: Our implementation of IHACRES was based on Figure 1 in Littlewood et al (1997) and sections 2.1.1 to 2.1.3. The pure time delay is only mentioned in the summary paragraph of section 2 (2.1.5). Removing this time delay from the MARRMoT model was no conscious choice, we simply didn't see this bit of information. We have made the following changes:

- Created a new Unit Hydrograph function: uh_8_delay, following your suggestions in the comment below
- We have changed our IHACRES model function (m05) as follows:
    - Included the new routing component (new parameter, new routing code, new water balance code)
    - Changed the name of this function from "m05_ihacres_6p_1s" to "**m05_ihacres_7p_1s"** to reflect the additional parameter
    - Tested this code to ensure no water balance errors have been introduced
- We have updated the IHACRES parameter range function
    - New name to reflect the additional parameter
    - New parameter range
- We have updated Figure 1 to reflect the increased number of parameters in our IHACRES version
- We have updated Table 1 in the User Manual to reflect the additional UH option
- We have created a new section 4.8 in the Supporting Materials that outlines the new UH option
- We have updated the model description in section S2.5
- We have updated Table S3 (which shows an overview of recommended parameter ranges in MARRMoT) to include the new time delay in IHACRES

To reflect this substantial change to the MARRMoT code, we have decided to increment the version number to v1.1.

(32) **Reviewer comment,** Supplementary material, S4: The pure time delay mentioned in the previous comment could be introduced as another option of unit hydrograph. Actually, it can be easily coded as a UH, which would have only two non-zero ordinates. If td is the time delay (it can be a real value, not necessarily an integer value), the two non-zero ordinates would be respectively td-int(td) and 1-td+int(td).

Author response: Thank you for specifying the required code. We have implemented this as "UH_8_delay".

(33) **Reviewer comment,** Supplementary material, S4,7: Why a question mark before "Moore and Bell"?

Author response: The question mark is the result of an error in referencing. Thanks for pointing this out. This is now corrected. Changes:
Supplements P145: "References E.g. MCRM (**Bell et al, 2001;** Moore and Bell, 2001)"

(34) **Reviewer comment,** P150: Not sure I fully understand the note on the filling parameter.

Author response: This parameter can be used to let the depression store in model 36 fill according to an exponential rate, where the shape of this exponential profile is decided by this parameter's value. However, literature applications of model 36 tend to set this parameter to 1, because no information is available on which plausible ranges of this parameter can be based. We've changed the text slightly to clarify this:
Supplements, P150: "Controls the **exponential rate** of depression store inflow flux but is usually set at 1 because no studies **are available that can be used to set plausible ranges**"

Cited references (reviewers)

Refsgaard, J. C. and H. J. Henriksen. 2004. Modelling guidelines - terminology and guiding principles. Advances in Water Resources 27: 71-82.

Refsgaard, J. C., H. J. Henriksen, B. Harrar, H. Scholten, and A. Kassahun. 2005. Quality assurance in model based water management - review of existing practice and outline of new approaches. Environmental Modelling & Software, 20: 1201–1215.

Scholten, H., A. Kassahun, J.C. Refsgaard, T. Kargas, C. Gavardinas and A.J.M. Beulens, 2007. A methodology to support multidisciplinary model-based water management. Environmental Modelling & Software 22, 743-759.

Additional references (author response)

Addor, N. and Melsen, L. A.: Legacy, Rather Than Adequacy, Drives the Selection of Hydrological Models, Water Resour. Res., 55(1), 378–390, doi:10.1029/2018WR022958, 2019.
Beven, K., Lamb, R., Quinn, P., Romanowicz, R. and Freer, J.: TOPMODEL, in Computer Models of Watershed Hydrology, edited by V. P. Singh, pp. 627–668, Water Resources Publications, USA, Baton Rouge., 1995.
Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. a., Vrugt, J. a., Gupta, H. V., Wagener, T. and Hay, L. E.: Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models, Water Resour. Res., 44(12), doi:10.1029/2007WR006735, 2008.
Lindström, G., Johansson, B., Persson, M., Gardelin, M. and Bergström, S.: Development and test of the distributed HBV-96 hydrological model, J. Hydrol., 201, 272–288, doi:https://doi.org/10.1016/S0022-1694(97)00041-3, 1997.