Manuscript

Responses to Reviewers

Dear editor and reviewers,

Thanks very much for taking your time to review this manuscript. We really appreciate all your comments and suggestions, which are very helpful for us to improve the manuscript. We have made a thorough revision to address the comments and questions. Please find our point-to-point responses below.

Reviewers' Comments to the Authors:

Reviewer 1

1. *I would like a little more detail of is contained within the "black box" placed in the SI to give those who are interested this information.*

Thanks for the suggestion. The black box in the manuscript refers to the execution of data assimilation, which is described in sections 2.1 and 2.4. We hope the reviewer will find the description is enough for readers to understand data assimilation. Moreover, sections 2.1 and 2.4 cite papers for readers to learn more about the method.

2. *I wonder about the trade-offs between the usually very fast exchange of information achieved when writing an interface vs the more user friendly approach described here? Not an objection to your approach but genuinely curious.*

This is a great question. We appreciate it. Generally, MIDA requires longer computation time than the embedded data assimilation (DA) algorithms. The time difference depends on how to call model simulation. Taking DALEC model in the first study as an example, the time cost for the embedded algorithm is 24 mins while MIDA takes 52 mins to finish DA. Thus, we recommend the embedded algorithm for complex models with high computational demand while MIDA is more suitable for beginners of DA users with models that are less complex. We have added this information about computation efficiency of MIDA in the discussion section of the manuscript (L569-572). Corresponding code is added to the Code Availability and is available in Zenodo repository (https://doi.org/10.5281/zenodo.4891319).

3. *What isn't quite so clear is the details of how MIDA knows which information in the existing model output files corresponds to its observations. For example, the namelist.txt must contain information on the variable names used to describe the observations and their corresponding output variable generated by the model? These must still vary depending on the model being used? A screen shot showing the interface which is populated with an example would make this really clear.*

Thanks the reviewer for the great questions and suggestion. The information in the model output files that corresponds to its observations is in an output configuration file (e.g., config.txt), which notifies MIDA how to map model outputs to the observations. Users need to prepare the configuration file because, as the reviewer has mentioned, the configurations (or mapping functions) vary depending on the model being used. The output configuration file is described in L333-342. As suggested, we added a screenshot of the output configuration file in Appendix B and also added a link of Appendix B in L342.

4. *The models need to be able to read the parameters from a file. The MIDA framework must then be able to write out the proposed parameters in a unique format for each model, is that correct?*

Yes, that is correct. We have described how MIDA writes new parameter values to a file 'ParameterValue.txt', from which the model reads to execute simulations in L260-262 and L321-325.

5. *L322-330: Could you add a link to further details in SI for this section? The reason I ask is that Haario et al., (2001) steps based on the weighted (e.g. beta) combination of the multivariate Gaussian and a minimum step size scaled by a value drawn from a Gaussian distribution of mean = 0, sd = 1. The multivariate Gaussian being derived from the covariance matrix for the parameters adjusted by an optimal scaling parameter (e.g. 2.38 / npars^0.5). The weighting between the two steps (beta ~0.05) and the minimum step size. So which of these variables (or something else entirely) for example is you "jump scaling"?*

We thank the reviewer for this suggestion. Haario et al. (2001) introduced an adapted Metropolis algorithm, in which the proposal distribution is tuned along the search according to the covariance calculated from previous samples. The Metropolis-Hasting (MH) algorithm in this study uses a fixed Gaussian proposal distribution, in which the covariance is provided from test runs. A parameter covariance is not provided, the MH algorithm uses a uniform proposal distribution instead following this equation: $C_{new} = C_{old} + r \times (C_{max} - C_{min})/D$, where r is a random number uniformly distributed in $[-0.5, 0.5]$, C_{max} and C_{min} are the maximum and minimum limits of parameters, respectively, D is a scalar controlling the proposing step size. Users can change the value of D in the 'namelist.txt' file. The above content has been described in L233-237. To avoid the misleading by the citation of Haario et al. (2001), we corrected the citation to Metropolis et al. (1953) and Hastings (1970) in L228. We also added a citation of Xu et al. (2006) in L237 as Appendix B of Xu et al. (2006) explained the MH algorithm in detail.

The paragraph in the original version manuscript (L322-330) mainly described how to adjust the acceptance rate, which is a critical index to assess the performance of DA. And more details can be found in Xu et al. (2006), of which we have cited. So, we believe these would be adequate for readers to understand the method.

6. *I like the inclusion of a screenshot of the software but I think it would be useful to have an example which has been filled in to help guide the potential user. Alternatively showing an example of the namelist.txt might be informative.*

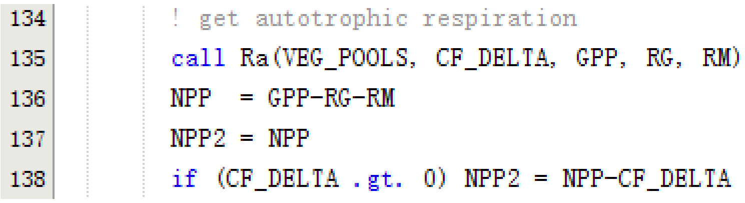
Thanks for the suggestion. We added a screenshot of the namelist.txt in Appendix C for the first case study with DALEC model. A link to the Appendix C is also provided in L320.

7. *This doesn't really impact the validity of the paper but just something I noticed and wanted to raise as it should really be clarified. The DALEC model is stated as having 5 C pools but also to having a Growing Degree Days phenology model. However, the Williams et al., (2005) model doesn't have phenology model (i.e. continuous allocation / evergreen). DALEC was split into deciduous and evergreen versions in Fox et al., (2009) as part of the reflex project adding a 6th pool and the GDD model. The example DALEC code provided on the MIDA Github shows a alternate version of the model where leaf C is not dependent on GPP (and thus the system is not mass balanced). This is a distraction from the main point of*

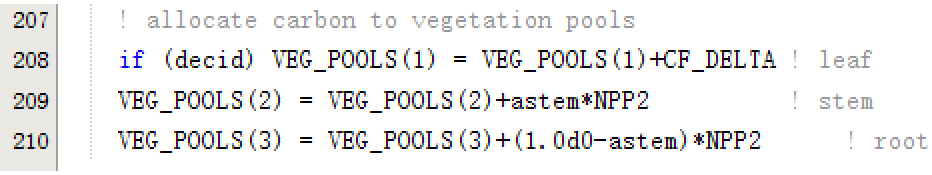
demonstrating your DA system. Please make the origin of the code clear as it doesn't match that found in the citations given.

We apologize for the inconsistence. The version of DALEC model we used in this study is the version described by Lu et al. (2017). It origins from Williams et al. (2005) but with some structural modifications. For example, the version of DALCE model by Lu et al. (2017) incorporates the phenology submodel developed by Ricciuto et al. (2011). Compared to the version of DALEC used in Fox et al. (2009), the model used in this study works for deciduous species and the plant labile pool is removed for simplification. We corrected the citation to Lu et al. (2017) in L375.

In the code for DALEC in this manuscript, GPP is first consumed in autotrophic respiration, i.e., growth respiration (RG) and maintenance respiration (EM), and then is allocated to three vegetation pools, i.e., foliage (VEG_POOLS(1)), wood (VEG_POOLS(2)), and root (VEG_POOLS(3)). The variable NPP2 in L138 in the code is NPP minus change in leaf mass (CF_DELTA) which is used to update foliage pool. NPP2 in L209-210 in the code is used to update wood and root pools. Therefore, the sum of the changes in the three vegetation pools equals to NPP. Therefore, the DALEC model in this study has C mass balance. More detailed information is in Lu et al. (2017) which we cited in L375.



```
134      ! get autotrophic respiration
135      call Ra(VEG_POOLS, CF_DELTA, GPP, RG, RM)
136      NPP  = GPP-RG-RM
137      NPP2 = NPP
138      if (CF_DELTA .gt. 0) NPP2 = NPP-CF_DELTA
```



```
207    ! allocate carbon to vegetation pools
208    if (decid) VEG_POOLS(1) = VEG_POOLS(1)+CF_DELTA ! leaf
209    VEG_POOLS(2) = VEG_POOLS(2)+astem*NPP2          ! stem
210    VEG_POOLS(3) = VEG_POOLS(3)+(1.0d0-astem)*NPP2      ! root
```

8. *L140: "DA is a statistical approach..." – there are many different algorithms for DA whether for state update or parameter estimation (in this case). I think it would be clearer refer to it as an "approach". I can see that you are trying to talk about your specific approach so maybe "The DA approach embeded within MIDA..."*

We apologize for the confusion. Currently, only Metropolis Hasting algorithm is embedded in MIDA, but MIDA is open to incorporate many other DA algorithms. Therefore, it would be more appropriate to use "approach" rather than "The DA approach embedded within MIDA". We have changed "DA is a statistical algorithm" to "DA is a statistical approach" as suggested in L141.

9. L176: "hinders" or "hides"?

It should be "hides". We have corrected this typo in L191.

10. L454: "This model simulates..."

We have changed as suggested in L475.