ParticleDA.jl v.1.0: A distributed particle filtering data assimilation package

Response 2

November 16, 2023

We would like to thank the reviewers and editor for their helpful comments and feedback. Our response to their points are outlined below. An updated version of the article with highlights of the changes made has been uploaded.

1 Topic Editor

> This manuscript is an excellent contribution to the DA and geoscience communities. As all reviewers think highly of this manuscript, I recommend that the manuscript needs significant revision to address reviewer 3's comments and suggestions. In particular, I agree with reviewer 3's comments that the authors should add a new experiment with nonlinear observation operator to emphasize the robustness of the package in dealing with nonlinear dynamics, which geophysical models commonly deal with. Also, the Lorenz model is a very common simple model for evaluating innovative DA algorithms. Therefore, the authors should follow the setup of the popular references and provide the analysis RMSE to give some basic idea about the performance of the new ParticleDA.jl v.1.0 package. Currently, the choice of the observation error variance is very small.

Author's response: Taking the suggestions on board section 5.1 has been added which showcases the performance of the bootstrap filter with a non-linear observation operator. Further, the performance of the locally optimal proposal for varying numbers of particles on the Lorenz 63 system is now highlighted in the updated Fig. 4, please be advised that the observation error variance has been increased in this case.

2 Reviewer 3

I appreciate the authors' efforts to build a parallelizable model-agnostic particle filter system, which holds promise for various geoscience applications. Nevertheless, I would like to raise some concerns about the state transition density, or the model error Q, in the following:

1. The difference of the particle weights between using the bootstrap and using the optimal proposal is determined by Q (e.g., see the reference below). Therefore, the way that Q is specified determines how much the optimal proposal emphasize the importance of the choice of Q.

   e.g., see Sections 9.2.2-9.2.3 in Evensen, Geir, Femke C. Vossepoel, and Peter Jan van Leeuwen. Data assimilation fundamentals: A unified formulation of the state and parameter estimation problem. Springer Nature, 2022.

   The particle filter algorithm itself is model-agnostic, while Q is not. Although this manuscript has provided examples illustrating the generation of Q, in general this is not trivial for many geophysical models. For example, not only do we need to consider the smoothness of the state variable, but also the physical constraints across different variable types. For example, the wind and pressure should largely satisfy the geostrophic balance relation in the AGCM. In addition, for many geoscience applications, Q can be state dependent. For example, the model errors for predicting a heat wave can be very different from predicting a hurricane. Probably beyond the scope of this study, the transition density can also be quite non-Gaussian, e.g., for modeling the convection process in weather prediction models.

   The user might need to build Q for their own model, which again, is not trivial for many geophysical models. Since this manuscript is submitted to GMD, I would recommend expand the discussions surrounding the construction of Q for geoscience applications (e.g., in the conclusion section).

Author's response: The author's recognise that this is a key ingredient and was a focus in the previous round of revision. Please see lines 110 - 112 where the importance of maintaining physical constraints and relationships of the underlying physics is highlighted. To further expand on this point we have added a paragraph in conclusion section on this, please see lines 502 - 509.

2. I would recommend elaborate more on how *get_covariance_state_noise* and *get_covariance_state_observation_given_previous_state* are being evaluated. Are there any computational challenges to evaluate these two functions for a very high-dimensional (e.g., $d_x \sim 10^9$) model?

Author's response: The authors agree that this is a key point, and we have added further detail to manuscript in Section 3.2 (*Model interface*) describing how the implementation allows exploiting sparsity in the observation operator H to keep the computational and memory costs manageable when working with models with high-dimensional state spaces.

2.1 Minor comments

1. For a spatially extended model, like a weather prediction model, the dimension can be as high as 10^9 . Will this be an issue when copying a state from one processor to another? How does the overall algorithm scale with the dimension of the model state d_x?

Author's response: This is a very important point with the key question focusing on how the package will perform when the spatial model is also built upon a distributed memory set up. For example weather prediction models traditionally rely on MPI (Message Passing Interface) to run across multiple compute nodes and how ParticleDA.jl performs in this case is the focus of ongoing development but is currently beyond the scope of this work.

2. I find it somewhat less convincing that the particle filter can work better than any existing linear and Gaussian DA methods in the experiments (e.g., from the results in Figs 8-9). Nevertheless, I do understand the primary goal of this manuscript is to showcase the capabilities of the package, instead of proposing a novel particle filter methodology and conduct comprehensive comparisons with existing methods, etc.

   Therefore, I do not insist but recommend, e.g., add a new experiment with non- linear observation operator, and/or compare the performance of PF and an ensemble Kalman filter against the ground truth in an OSSE (e.g., in Figure 8, you could also add a panel that shows the results from using an ensemble Kalman filter).

Author's response: As suggested an additional experiment (section 5.1) has been added with a non-linear observation operator for the Lorenz 63 system. The introduction of this non-linear operator influences the performance of the bootstrap proposal. We also agree that a comparison of the particle filter against an ensemble Kalman filter for the AGCM case is beyond the scope here, but references have been made to the performance of a Local Ensemble Kalman Transform Filter (LETKF) in a similar non-linear set-up (see lines 473-476).

3. Figure 8: is the unit of the time averaged error correct? An error in surface pressure exceeding 50 hPa seems unrealistically large.

Author's response: The units on the time averaged error are correct. The authors are aware that this is quite large in comparison to other studies but acknowledge that in this set up only the surface pressure is observed and on a very sparse observation network. The purpose of Fig. 9 (previously Fig. 8) is to emphasis the performance of the particle filter against an ensemble where no assimilation is occurring and to showcase the interoperability of the package with a pre-exisiting Fortran codebase. Please see the discussion in the previous round of revisions.