

Reviewer 1

General comment

The authors have satisfactorily addressed all my previous comments. I only have a few minor suggestions and comments listed below. I recommend that the manuscript be accepted after resolving these minor points.

Minor comments:

1. Lines 121-123: Suggest rephrasing the sentence as “Other typical filtering algorithms, such as ensemble adjustment Kalman filter (EAKF, Anderson, 2001) and ensemble square root filters (EnSRF, Whitaker and Hamill, 2002), are not implemented in current releases but are planned to be included in future releases.”

Answer: We rephrase the sentence based on the reviewer’s suggestion with a slight adjustment. The last part of the sentence is written in a clearer fashion due to newer development in PDAF.

“not implemented in PDAF V2.3 used in this work, but were introduced in the newer release V3.0.”

2. Line 232: “with the OMI functionality, only three user-supplied functions need to be implemented.” It is not clear from the following text which three user-supplied functions are needed.

Answer: Thank you for the point. We rephrase the sentences and add the following content in Sect.2.4:

To handle different observations, with the OMI functionality, only three user-supplied functions need to be implemented:

Func 1: a function that provides observation information like observation values, errors and coordinates ($dim_obs = init_dim_obs(step, dim_obs)$) where its primarily purpose is to provide the dimension of observation vector (dim_obs) to PDAF,

Func 2: a function that provides the observation operator ($m_state_p = obs_op(step, dim_p, dim_obs, state_p, m_state_p)$), where the observation operator transforms the state vector ($state_p$) into observation space (m_state_p),

Func 3: a function that specifies the number of observations being assimilated in each local domain ($dim_obs_l = init_dim_obs_l(domain_p, step, dim_obs, dim_obs_l)$).

3. Lines 386-395 and 401-404: “In WCDA, each model component performs DA independently ... and define the local state vector as either the atmosphere or ocean variables.” and “Compared to the WCDA, ... and does not require special treatment.” These sentences do not fit well in the context of Section 4.1 “Skill of data assimilation.” They may be more appropriately placed in Section 3.2 “Experiment design” or somewhere else.

Answer: They are moved to Sect. 3.2 “Experiment design”.

4. Line 437: “the ratio of total computational time” should be “the ratio of computational time for ‘pre-post’.”

Answer: Done.

5. Line 441: “... but the ratio is only 2.04 and 3.58 for ‘distribute state’ and ‘collect state’ ...” The ratio should be 3.58 and 8.60 according to Table 2. However, this correction makes the reduction of pyPDAF overhead much less impressive, so the sentence may need to be further revised.

Answer: We corrected the mistake and slightly changed the wording:

The overhead in the pyPDAF system is less affected by the dimension of the state vector for the distribution and collection of state vector (labelled ‘distribute state’ and ‘collect state’) because these functions only exchange information between model and PDAF without intensive computation. For example, the pyPDAF system takes 3.82 and 8.89 times of computational time of the PDAF system for ‘distribute state’ and ‘collect state’ respectively on a 129×129 grid, which is similar to the ratio of 3.58 and 8.60 on a 2049×2049 grid.

6. Table 3: Please clarify why the wall clock times of all components do not sum to the “total” time shown in the last row of the table.

Answer: To match the total time, we added in Tab. 3 the omitted ‘local obs. search’ entry which does not affect our main results. We also added:

The similar computational time applies for the case when PDAF search for local observations for analysis local domains due to its intensive numerical computation.

7. Line 562: I am not sure where the number “70%” comes from.

Answer: This is a mistake from revision. We rephrase the sentence to:

In the scope of our specific experiment setup, compared with PDAF, our benchmark shows that, depending on the size of the state vector and ensemble, from 28% to around three times more time (see Tab. 2 and 4) is used by pyPDAF with the global filter while only 6% – 13% more time is required with a domain-localized filter when applying the Python DA system build with pyPDAF in a high-dimensional dynamical system.