

***Interactive comment on* “Efficient ensemble data assimilation for coupled models with the Parallel Data Assimilation Framework: Example of AWI-CM” by Lars Nerger et al.**

Lars Nerger et al.

lars.nerger@awi.de

Received and published: 26 November 2019

Dear Dr. Liu,
thank you for commenting on our manuscript. Your comments helped to clarify some aspects in the manuscript. Please see our explanations below.

Dear authors, Thanks a lot for making this article of the latest improvement of PDAF online on GMD for discussion.

I am Li Liu from Tsinghua University, China, leading the development of C-Coupler that is a Chinese coupler family for Earth system modelling. I am very interested in

[Printer-friendly version](#)

[Discussion paper](#)



the software framework for ensemble data assimilation, even leading a research in this topic. So I am very interested in your PDAF work, and have learned a lot from its documentations, source codes, and this article.

After reading this article, I have the following concerns:

1. After downloading and then reading the latest available code version of PDAF from your website, I guess that it does not fully include the implementation for this article.

Response: In deed the model binding for AWI-CM which is described in the manuscript is not yet in the release package of PDAF. We will include it in the next release. To fulfill the requirements at GMD, we now provide the model binding code also at zenodo.org (<http://doi.org/10.5281/zenodo.3551667>).

2. Figure 3 of this article and the source code of PDAF may indicate that the certain order of processes in the MPI_COMM_WORLD among ensemble members of the coupled model as well as its component models is required, and different members of the same component model must have the same number of processes. For example, the IDs of processes of atm_member1, ocn_member1, atm_member2, ocn_member2, atm_member3 and ocn_member3, and the processes not involved in ensemble data assimilation, must be in an ascending/descending order. Is there any restriction about the processes not involved in ensemble data assimilation?

Response: Actually, the configuration of the parallelization is performed in the routine `init_parallel_pdaf`. The routine provides one common configuration, which we have found to be compatible with all models we have worked with so far. However, `init_parallel_pdaf` is also intended to be a template which can be adapted by the user (as is stated in the header of the file). The template

character of the file is also the reason why its name does not start with PDAF_ and why it is not part of the PDAF library itself. Thus, while the provided default configuration assumes a particular order of the processes, one can adapt `init_parallel_pdaf` for cases with different orders. In fact, we had to adapt the single-program variant of the routine, which is provided in the PDAF release package, for the coupled model with 2 executables. Section 3.3 describes this configuration.

3. Regarding P9L261_270, it seems unclear how to split the communicator for a set of processes exclusive from ensemble data assimilation when splitting `MPI_COMM_WORLD` into a group of `COMM_CPLMOD`. Is there any new modification in the APIs or input files of PDAF for this functionality?

Response: As described above, the only required change is in the routine `init_parallel_pdaf`. One required change was to account for the fact the the number of processes per model task is the number of processes for ECHAM and those for FESOM. These are read from a namelist file. However, no changes to the API of PDAF were required. Further, as PDAF itself is a software library, inputs are defined by the user and hence PDAF itself does not use input files.

4. Regarding the weakly-coupled application mentioned in P10L285_L288, it seems unclear how to generate separate `COMM_FILTER` for ECHAM and FESOM? Regarding this functionality, I guess that PDAF should know all component models of the coupled model and the number of processes of each component model, and know that weak coupling but not strong coupling is used. It may be interesting to know how PDAF is extended for this kind of input.

[Printer-friendly version](#)[Discussion paper](#)

Response: The difference in the weakly and strongly coupled assimilation is really just the different setup of `COMM_filter`. To distinguish these cases, we have introduced a flag `DA_couple_type` in the code, which allows to switch between both assimilation modes.

5. Regarding Figure 2, it is still unclear of the code flowchart of different component models in weak coupling. For example, given that only ECHAM is involved in data assimilation but FESOM is not, it is unclear whether only ECHAM calls `init_PDAF` and `Assimilate_PDAF`, or FESOM has to call these two APIs cooperatively?

Response: We have included in the flowchart only one sketch of the ‘Model with DA extension’ because the additional subroutine calls are added likewise in FESOM and ECHAM. This is now better clarified in Sec. 3.2.

6. PDAF requires the filter to use the same parallel decomposition with the model. Is it possible to introduce challenges when integrating an existing filter that already has its own parallel decomposition that may be different from the model. For example, a land surface model generally uses a round-robin parallel decomposition for load balance in parallelization, which may be not suitable for a filter or will introduce new code development or lower efficiency to the filter.

Response: Actually, PDAF was already applied with TerrSysMP, which includes a land surface model. The setup for this model was discussed by Kurtz et al. (2016). The assumption of the default setup of PDAF is that there is one pair of MPI communication calls between rank 0 and each other rank in `COMM_COUPLE`. Thus, communication patterns that would need several calls are not supported in the default setup of PDAF. In addition, always full sub-

[Printer-friendly version](#)[Discussion paper](#)

domains of state vectors are communicated. However, the number of processes in `COMM_COUPLE` does not need to be equal to the number of model tasks. For the default setup of PDAF, we have not attempted to provide a more general communication pattern in a generic form. In particular this would make the configuration phase much more difficult for users of PDAF while most users can use the default setup efficiently. Also, it should be possible to apply the filter to a round-robin distributed decomposition if all ensemble members use the same distribution. This is because the analysis step of PDAF does not assume compact sub-domains. However, for efficiency PDAF assumes that in the filter all ensemble members use the same parallel decomposition. If the default setup of PDAF is not directly usable, one could collect consistent sub-domains on the model processes and then communicate these to PDAF. Finally, one could also modify the communication pattern in the routines `PDAF_get_state` and `PDAF_gather_ens` of the PDAF library to adapt to a particular case. Please note that we have not included this detailed discussion in the manuscript, because we feel that this would be too much aimed for specialists.

Interactive comment on Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2019-167>, 2019.

[Printer-friendly version](#)[Discussion paper](#)