Geosci. Model Dev., 13, 4305–4321, 2020
https://doi.org/10.5194/gmd-13-4305-2020
© Author(s) 2020. This work is distributed under
the Creative Commons Attribution 4.0 License.

Geoscientific Model Development
Open Access

Efficient ensemble data assimilation for coupled models with the Parallel Data Assimilation Framework: example of AWI-CM (AWI-CM-PDAF 1.0)

Lars Nerger, Qi Tang, and Longjiang Mu

Alfred-Wegener-Institut, Helmholtz-Zentrum für Polar- und Meeresforschung (AWI), Bremerhaven, Germany

Correspondence: Lars Nerger (lars.nerger@awi.de)

Received: 11 June 2019 – Discussion started: 1 August 2019
Revised: 13 May 2020 – Accepted: 4 August 2020 – Published: 16 September 2020

Abstract. Data assimilation integrates information from observational measurements with numerical models. When used with coupled models of Earth system compartments, e.g., the atmosphere and the ocean, consistent joint states can be estimated. A common approach for data assimilation is ensemble-based methods which utilize an ensemble of state realizations to estimate the state and its uncertainty. These methods are far more costly to compute than a single coupled model because of the required integration of the ensemble. However, with uncoupled models, the ensemble methods also have been shown to exhibit a particularly good scaling behavior. This study discusses an approach to augment a coupled model with data assimilation functionality provided by the Parallel Data Assimilation Framework (PDAF). Using only minimal changes in the codes of the different compartment models, a particularly efficient data assimilation system is generated that utilizes parallelization and in-memory data transfers between the models and the data assimilation functions and hence avoids most of the file reading and writing, as well as model restarts during the data assimilation process. This study explains the required modifications to the programs with the example of the coupled atmosphere–sea-ice–ocean model AWI-CM (AWI Climate Model). Using the case of the assimilation of oceanic observations shows that the data assimilation leads only to small overheads in computing time of about 15 % compared to the model without data assimilation and a very good parallel scalability. The model-agnostic structure of the assimilation software ensures a separation of concerns in which the development of data assimilation methods can be separated from the model application.

1 Introduction

Data assimilation (DA) methods are used to combine observational information with models. A common application is to apply DA to estimate an initial state that is used to start a forecast system as is common practice at weather and marine forecasting centers. The most widely used class of ensemble DA methods are ensemble-based Kalman filters (EnKFs) like the local ensemble transform Kalman filter (LETKF; Hunt et al., 2007), the deterministic ensemble Kalman filter (DEnKF; Sakov and Oke, 2008), or the local error-subspace transform Kalman filter (LESTKF; Nerger et al., 2012b). Commonly, DA is applied to separate models simulating, for example, the atmospheric dynamics or the ocean circulation. However, in recent years coupled models of different Earth system compartments have become more common. In this case, the compartment models frequently exchange information at the interface of the model domains to influence the integration of the other model compartment. For example, in coupled atmosphere–ocean models the fluxes through the ocean surface are dynamically computed based on the physical state of both the atmosphere and the ocean, and these are exchanged in between both compartments. For model initialization, DA should be applied to each of the compartments. Here, DA can either be performed separately in the different compartment domains, commonly called weakly coupled DA, or it can be performed in a joint update, called strongly coupled DA. Only strongly coupled DA is expected to provide fully dynamically consistent state estimates.

A recent overview of methods and issues in coupled DA is provided by Penny et al. (2017). By now the weakly coupled assimilation is the common choice for assimilation into coupled models and recent studies assess the effect of this assimilation approach. For atmosphere–ocean coupled models, different studies either assimilated observations of one compartment into the observed compartment (e.g., Kunii et al., 2017; Mu et al., 2020) or assimilated observations of each compartment into the corresponding observed compartments (e.g., Zhang et al., 2007; Liu et al., 2013; Han et al., 2013; Chang et al., 2013; Lea et al., 2015; Karspeck et al., 2018; Browne et al., 2019). The research question considered in these studies is usually to what extent the assimilation into a coupled model can improve predictions in comparison to the assimilation into uncoupled models. Partly, the mentioned studies used twin experiments assimilating synthetic observations to assess the DA behavior.

Strongly coupled DA is a much younger approach, which is not yet well established. Open questions for strongly coupled DA are, for example, how to account for the different temporal and spatial scales in the atmosphere and the ocean. Strongly coupled DA is complicated by the fact that DA systems for the ocean and atmosphere have usually been developed separately and often use different DA methods. For example, Laloyaux et al. (2016) used a 3D variational DA in the ocean but 4D variational DA in the atmosphere. The methodology led to a quasi-strongly coupled DA. Frolov et al. (2016) proposed an interface-solver approach for variational DA methods, which leads to a particular solution for the variables close to the interface. Strongly coupled DA was applied by Sluka et al. (2016) in a twin experiment using an EnKF with dynamically estimated covariances between the atmosphere and ocean in a low-resolution coupled model. For coupled ocean–biogeochemical models, Yu et al. (2018) discussed strongly coupled DA in an idealized configuration. Further, Goodliff et al. (2019) discussed the strongly coupled DA for a coastal ocean–biogeochemical model assimilating real observations of sea surface temperature. This study pointed to the further complication of the choice of variable (linear or logarithmic concentrations for the biogeochemical compartment) for strongly coupled assimilation.

Ensemble-based Kalman filters (but also the nonlinear particle filters) can be formulated to work entirely on state vectors. A state vector is the collection of all model fields at all model grid points in the form of a vector. When one computes the observed part of the state vector, applying the so-called "observation operator", one needs to know how a field is stored in the state vector. However, the core part of the filter, which computes the corrected state vector (the so-called "analysis state") taking into account the observational information, does not need to know how the state vector is constructed. This property is also important for coupled DA, where the state vector will be distributed over different compartments, such as the atmosphere and the ocean.

The possibility to implement most parts of a filter algorithm in a generic model-agnostic way has motivated the implementation of software frameworks for ensemble DA. While the frameworks use very similar filter methods, they differ strongly in the strategy of how the coupling between model and DA software is achieved. As described by Nerger et al. (2012b), one can distinguish between offline and online DA coupling. In offline-coupled DA one uses separate programs for the model and the assimilation and performs the data transfer between both through disk files. In online-coupled DA one performs in-memory data transfer, usually by parallel communication, and hence avoids the use of disk files. In addition, online-coupled DA avoids the need to stop and restart a model for DA. The Data Assimilation Research Testbed (DART; Anderson et al., 2009) uses file transfers and separate programs for the ensemble integration and the filter analysis steps, which are run consecutively. The framework "Employing Message Passing Interface for Researching Ensembles" (EMPIRE; Browne and Wilson, 2015) uses parallel communication between separate programs for model and DA. These programs are run in parallel and the information transfer is performed through parallel communication, which avoids data transfers using files. The Parallel Data Assimilation Framework (PDAF; Nerger et al., 2005; Nerger et al., 2012b, http://pdaf.awi.de, last access: 14 September 2020) supports both online- and offline-coupled DA. For the online-coupled DA, PDAF also uses parallel communication. However, in contrast to EMPIRE, the model usually is augmented by the DA functionality; i.e., model and DA are compiled into a joint program.

For coupled ensemble DA in hydrology, Kurtz et al. (2016) have combined PDAF with the coupled terrestrial model system TerrSysMP. To build the system, a wrapper was developed to perform the online coupling of model and DA software. The study shows that the resulting assimilation system is highly scalable and efficient. Karspeck et al. (2018) have discussed a coupled atmosphere–ocean DA system. They apply the DART software and perform weakly coupled DA using two separate ensemble-based filters for the ocean and atmosphere, which produce restart files for each model compartment. These are then used to initialize the ensemble integration of the coupled model.

Here, we discuss a strategy to build an online-coupled DA system for coupled models with the example of the coupled atmosphere–ocean model AWI-CM. The strategy enhances the one discussed in Nerger et al. (2012b) for an ocean-only model. The previous strategy is modified for the coupled DA and applied to the two separate programs for the atmosphere and ocean, which together build the coupled model AWI-CM (Sidorenko et al., 2015). The required modifications to the model source codes consist essentially of adding four subroutine calls in each of the two compartment models. Three of these subroutine calls connect the models to the DA functionality provided by PDAF, while the fourth is optional and provides timing and memory information. With this strategy, a wrapper that combines the compartment model into a single executable as used by Kurtz et al. (2016) can be avoided. We discuss the strategy for both weakly and strongly coupled DA but assess the parallel performance only for weakly coupled DA into the ocean, which is supported in the code version AWI-CM-PDAF V1.0. This is motivated by the fact that strongly coupled DA is not yet well established, and weakly coupled DA by itself is a topic of current research.

The remainder of the study is structured as follows. Section 2 discusses ensemble filters and their setup for coupled DA. The setup of a DA system is described in Sect. 3. Section 4 discusses the parallel performance of the DA system build by coupling AWI-CM and PDAF. Section 5 examines the assimilation behavior of an example application with AWI-CM. Implications of the chosen strategy for the coupled model and data assimilation are discussed in Sect. 6. Finally, conclusions are drawn in Sect. 7.

2 Ensemble filters

Ensemble DA (EnDA) methods use an ensemble of model state realizations to represent the state estimate (usually the ensemble mean) and the uncertainty of this estimate given by the ensemble spread. The filters perform two alternating phases. In the forecast phase the ensemble of model states is integrated with the numerical model until the time when observations are available. At this time, the analysis step is computed. It combines the information from the model state and the observations by taking into account the estimated error of the two information sources and computes an updated model state ensemble, which represents the analysis state estimate and its uncertainty.

The current most widely used ensemble filter methods are ensemble-based Kalman filters (Evensen, 1994; Houtekamer and Mitchell, 1998; Burgers et al., 1998). When incorporating the observations during the analysis step, these filters assume that the errors in the state and the observations are Gaussian distributed. This allows one to formulate the analysis step by just using the two leading moments of the distributions, namely, the mean and covariance matrix. Another class of EnDA methods are particle filters (e.g., van Leeuwen, 2009). While particle filters do not assume Gaussianity of error distributions, they are difficult to use with high-dimensional models, because particular adaptions are required to avoid the case when the ensemble collapses to a single member due to the so-called "curse of dimensionality" (see Snyder et al., 2008). Methods to make particle filters usable for high-dimension systems were reviewed by van Leeuwen et al. (2019). One strategy is to use the observational information already during the forecast phase to keep the ensemble states close to the observations. This approach requires that some DA functions are already executed during the forecast phase. The realization in the implementation strategy will be discussed in Sect. 3.2.

2.1 Filter algorithms

To be able to discuss the particularities of coupled DA with respect to ensemble filter, here the error-subspace transform Kalman filter (ESTKF; Nerger et al., 2012b) is reviewed. The ESTKF is an efficient formulation of the EnKF that has been applied in different studies to assimilate satellite data into sea-ice–ocean models (e.g., Kirchgessner et al., 2017; Mu et al., 2018; Androsov et al., 2019) and biogeochemical ocean models (e.g., Pradhan et al., 2019; Goodliff et al., 2019).

2.1.1 The ESTKF

In the analysis step at time t_k, the ESTKF transforms a forecast ensemble $\mathbf{X}_k^{\mathrm{f}}$ of N_{e} model states of size N_x stored in the columns of this matrix into a matrix of analysis states $\mathbf{X}_k^{\mathrm{a}}$ as

$$\mathbf{X}_k^{\mathrm{a}} = \overline{\boldsymbol{x}}_k^{\mathrm{f}} \mathbf{1}_{N_{\mathrm{e}}}^T + \mathbf{X}_k^{\mathrm{f}} \left(\boldsymbol{w}_k \mathbf{1}_{N_{\mathrm{e}}}^T + \tilde{\mathbf{W}}_k \right), \tag{1}$$

where $\overline{\boldsymbol{x}}_k^{\mathrm{f}}$ is the forecast ensemble mean state and $\mathbf{1}_{N_{\mathrm{e}}}$ is a vector of size N_{e} holding the value one in all elements. Further, $\boldsymbol{w}_k$ is a vector of size N_{e}, which transforms the ensemble mean and $\tilde{\mathbf{W}}$ is a matrix of size $N_{\mathrm{e}} \times N_{\mathrm{e}}$, which transforms the ensemble perturbations. Below, the time index k is omitted, as all computations in the analysis refer to time t_k.

The forecast ensemble represents an error subspace of dimension $N_{\mathrm{e}} - 1$, and the ESTKF computes the ensemble transformation matrix and vector in this subspace. Practically, one computes an error-subspace matrix by $\mathbf{L} = \mathbf{X}^{\mathrm{f}}\mathbf{T}$, where $\mathbf{T}$ is a projection matrix with $j = N_{\mathrm{e}}$ rows and $i = N_{\mathrm{e}} - 1$ columns defined by

$$\mathbf{T}_{j,i} = \begin{cases} 1 - \frac{1}{N_{\mathrm{e}}} \frac{1}{\frac{1}{\sqrt{N_{\mathrm{e}}}}+1} & \text{for } i = j, j < N_{\mathrm{e}} \\ -\frac{1}{N_{\mathrm{e}}} \frac{1}{\frac{1}{\sqrt{N_{\mathrm{e}}}}+1} & \text{for } i \neq j, j < N_{\mathrm{e}} \\ -\frac{1}{\sqrt{N_{\mathrm{e}}}} & \text{for } j = N_{\mathrm{e}}. \end{cases} \tag{2}$$

Below, the equations are written using $\mathbf{X}^{\mathrm{f}}$ and $\mathbf{T}$ rather than $\mathbf{L}$ as this leads to a more efficient formulation.

A model state vector $\boldsymbol{x}^{\mathrm{f}}$ and the vector of observations $\boldsymbol{y}$ with dimension N_y are related by the observation operator $\mathbf{H}$ by

$$\boldsymbol{y} = \mathbf{H}\left(\boldsymbol{x}^{\mathrm{f}}\right) + \boldsymbol{\epsilon}, \tag{3}$$

where $\boldsymbol{\epsilon}$ is the vector of observation errors, which are assumed to be a white Gaussian-distributed random process with observation error covariance matrix $\mathbf{R}$. For the analysis step, a transform matrix in the error subspace is computed as

$$\mathbf{A}^{-1} = \rho(N_{\mathrm{e}} - 1)\mathbf{I} + (\mathbf{H}\mathbf{X}^{\mathrm{f}}\mathbf{T})^T \mathbf{R}^{-1} \mathbf{H}\mathbf{X}^{\mathrm{f}}\mathbf{T} \ . \tag{4}$$

This matrix provides ensemble weights in the error subspace. The factor ρ with $0 < \rho \leq 1$ is called the "forgetting factor"

(Pham et al., 1998), and it is used to inflate the forecast error covariance matrix. The weight vector $\boldsymbol{w}_k$ and matrix $\tilde{\mathbf{W}}$ are now given by

$$\boldsymbol{w} = \mathbf{TA}\left(\mathbf{HX}^{\mathrm{f}}\mathbf{T}\right)^{T}\mathbf{R}^{-1}\left(\boldsymbol{y} - \mathbf{H}\overline{\boldsymbol{x}}^{\mathrm{f}}\right) \;, \tag{5}$$

$$\tilde{\mathbf{W}} = \sqrt{N_{\mathrm{e}} - 1}\mathbf{TA}^{1/2}\mathbf{T}^{T}, \tag{6}$$

where $\mathbf{A}^{1/2}$ is the symmetric square root which is computed from the eigenvalue decomposition $\mathbf{USU}^{T} = \mathbf{A}^{-1}$ such that $\mathbf{A}^{1/2} = \mathbf{US}^{-1/2}\mathbf{U}^{T}$. Likewise, $\mathbf{A}$ in Eq. (5) is computed as $\mathbf{A} = \mathbf{US}^{-1}\mathbf{U}^{T}$.

For high-dimensional models, a localized analysis is computed following Nerger et al. (2006). Here, each vertical column of the model grid is updated independently by a local analysis step. For updating a column, only observations within a horizontal influence radius l are taken into account. Thus, the observation operator is local and computes an observation vector within the influence radius l from the global model state. Further, each observation is weighted according to its distance from the water column to down-weight observations at larger distances (Hunt et al., 2007). The weight is applied by modifying matrix $\mathbf{R}^{-1}$ in Eqs. (4) and (5). The localization weight for the observations is computed from a correlation function with compact support given by a fifth-order polynomial with a shape similar to a Gaussian function (Gaspari and Cohn, 1999). The localization leads to individual transformation weights $\boldsymbol{w}_k$ and $\tilde{\mathbf{W}}$ for each local analysis domain.

2.2 Weakly coupled ensemble filtering

In weakly coupled DA, the EnKF is applied in the coupled model to a single compartment or separately to several of the compartments. Given that the analysis is separate for each involved compartment, the filter is applied as in a single-compartment model. Thus, in practice several EnKFs compute the analyses updates independently before the next forecast phase is started with the updated fields from the different compartments.

2.3 Strongly coupled ensemble filtering

To discuss strongly coupled filtering, let us assume a two-compartment system (perhaps the atmosphere and the ocean). Let $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{O}}$ denote the separate state vector in each compartment. For strongly coupled DA, both are joined into a single state vector $\boldsymbol{x}_{\mathrm{C}}$.

Using the joint forecast ensemble $\mathbf{X}_{\mathrm{C}}^{\mathrm{f}}$ in Eq. (1) of the ESTKF, one sees that the same ensemble weights $\boldsymbol{w}$ and $\tilde{\mathbf{W}}$ are applied to both $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{O}}$. The weights are computed using Eqs. (4) to (6). These equations involve the observed ensemble $\mathbf{HX}_{\mathrm{C}}^{\mathrm{f}}$, the observation vector $\boldsymbol{y}$, and the observation error covariance matrix $\mathbf{R}$. Thus, for strongly coupled DA, the updated weights depend on which compartment is observed. If there are observations of both compartments, they are jointly used to compute the weights. If only one compartment is observed, e.g., having only ocean observations $\boldsymbol{y}_{\mathrm{O}}$, then we also have $\mathbf{HX}_{\mathrm{C}}^{\mathrm{f}} = (\mathbf{HX}^{\mathrm{f}})_{\mathrm{O}}$, and the weights are only computed from these observations. Thus, through Eq. (1), the algorithm can directly update both compartments, $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{O}}$, using observations of just one compartment.

An interesting aspect is that when one runs separate assimilation systems for the two compartments with the same filter methodology, one can compute a strongly coupled analysis by only exchanging the parts of $\boldsymbol{y}$, $\mathbf{HX}^{\mathrm{f}}$, and $\mathbf{R}$ in between both compartments and then initializing the vectors containing observational information from all compartments in the assimilation system of each compartment. If there are only observations in one of the compartments, one can also compute the weights in that compartment and provide them to the other compartment. Given that $\boldsymbol{y}$ and $\mathbf{R}$ are initialized from information that is usually stored in files, one can also let the DA code coupled into each compartment model read these data and only exchange the necessary parts of $\mathbf{HX}^{\mathrm{f}}$. While this discussion shows that technically it is straightforward to apply strongly coupled DA with these filter methods, one has to account for the model parallelization, which is discussed in Section 3.3.

3 Setup of data assimilation program

This section describes the assimilation framework and the setup of the DA program. First an overview of PDAF is given (Sect. 3.1). The code modifications for online coupling are described in Sect. 3.2, and the modifications of the parallelization are described in Sect. 3.3. Finally, Sect. 3.4 explains the aspect of the call-back functions.

The setup builds on the strategy introduced by Nerger and Hiller (2013). Here, the discussion focuses on the particularities when using a coupled model consisting of separate executable programs for each compartment. While we here describe the features for both weakly and strongly coupled DA, AWI-CM-PDAF in version 1.0 is only coded with weakly coupled DA in the ocean. This is motivated by the fact that the weakly coupled DA in a coupled climate model has already different properties than DA in an uncoupled model. In particular, the initial errors in the coupled AWI-CM are much larger than in a simulation of FESOM (Finite Element Sea ice-Ocean Model) using atmospheric forcing. Mainly this is because in FESOM the forcing introduces information about the weather conditions, while AWI-CM only represents the climate state. Thus studying weakly coupled DA, which is still used in most applications, has a value on its own. Strongly coupled DA will be supported in the AWI-CM-PDAF model binding in the future.

3.1 Parallel Data Assimilation Framework (PDAF)

PDAF (Nerger and Hiller, 2013, http://pdaf.awi.de, last access: 14 September 2020) is a free open-source software that was developed to simplify the implementation and application of ensemble DA methods. PDAF provides a generic framework containing fully implemented and parallelized ensemble filter and smoother algorithms like the LETKF (Hunt et al., 2007), the ESTKF (Nerger et al., 2012b), or the nonlinear NETF method (Tödter and Ahrens, 2015) and related smoothers (e.g., Nerger et al., 2014; Kirchgessner et al., 2017). Further, it provides functionality to adapt a model parallelization for parallel ensemble forecasts as well as routines for parallel communication linking the model and filters. Analogous to many large-scale geoscientific simulation models, PDAF is implemented in Fortran and is parallelized using the Message Passing Interface standard (MPI; Gropp et al., 1994) as well as OpenMP (OpenMP, 2008). This ensures optimal compatibility with these models, while it is still usable with models coded, for example, in the programming language C.

The filter methods are model agnostic and only operate on abstract state vectors as described for the ESTKF in Sect. 2. This allows one to develop the DA methods independently from the model and to easily switch between different assimilation methods. Any operations specific to the model fields, the model grid, or to the assimilated observations are performed in program routines provided by the user based on existing template routines. The routines have a specified interface and are called by PDAF as call-back routines (i.e., the model code calls routines of PDAF), which then call the user routines. This call structure is outlined in Fig. 1. Here, an additional yellow "interface routine" is used in between the model code and the PDAF library routine. This interface routine is used to define parameters for the call to the PDAF library routines, so these do not need to be specified in the model code. Thus, only a single-line call to each interface routine is added to the model code, which keeps the changes to the model code to a minimum.

The motivation for this call structure is that the call-back routines exist in the context of the model (i.e., the user space) and can be implemented like model routines. In addition, the call-back routines can access static arrays allocated by the model, e.g., through Fortran modules or C header files. For example, this can be used to access arrays holding model fields or grid information. This structure can also be used in the case of an offline coupling using separate programs for the model and the analysis step. However, in this case the grid information is not already initialized by the model and has to be initialized by a separate routine. The interfaces and user routines provided by PDAF can also be used with models implemented in C or C++, or they can be combined with Python. For coupled models consisting of multiple executables, this call structure is used for each compartment model.

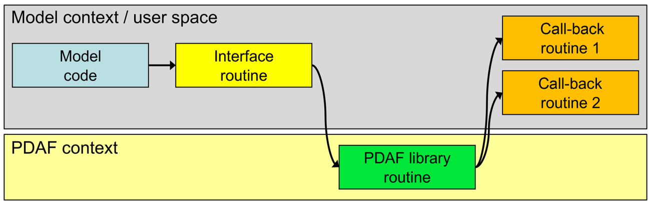

Figure 1. Call structure of PDAF. Calls to interface routines (yellow) are inserted into the model code (blue). The interface routines define parameters for PDAF and call PDAF library routines (green). These library routines call user-provided call-back routines. The model code, interface, and call-back routines operate in the model context and can hence exchange information indirectly, e.g., through Fortran modules. Likewise, the PDAF library routines share variables.

3.2 Augmenting a coupled model for ensemble data assimilation

Here, only the online coupling for DA is discussed. As described before, the offline coupling uses separate programs for the model and the DA program and model restart files to transfer information about the model states between both programs. Generally, the same code for the user routines can be used for online- and offline-coupled DA. The difference is that in the online coupling, model information like the model grid is initialized by the model code and usually stored in, for example, Fortran modules. For offline-coupled DA, one could use the same variable names and the same names for the modules. Thus, one would need to implement routines that initialize these variables.

The strategy to augment a coupled model with DA functionality is exemplified here using the AWI Climate Model (AWI-CM; Sidorenko et al., 2015). The model consists of the atmospheric model ECHAM6 (Stevens et al., 2013), which includes the land surface model JSBACH, and the Finite Element Sea ice-Ocean Model (FESOM; Danilov et al., 2004; Wang et al., 2008). Both models are coupled using the coupler library OASIS3-MCT (Ocean Atmosphere Sea Ice Soil – Model Coupling Toolkit; Valcke, 2013). OASIS3-MCT computes the fluxes between the ocean and the atmosphere and performs the interpolation between both model grids. The coupled model consists of two separate programs for ECHAM and FESOM, which are jointly started on the computer so that they can exchange data via the Message Passing Interface (MPI; Gropp et al., 1994). OASIS3-MCT is linked into each program as a library. For further details on the model, we refer to Sidorenko et al. (2015).

The online coupling for DA was already discussed in Nerger and Hiller (2013) for an earlier version of the ocean model used in the AWI-CM. Here, an updated coupling strategy is discussed that requires less changes to the model code. While the general strategy for online coupling of the DA is the same as in the previous study, we provide here a full de-

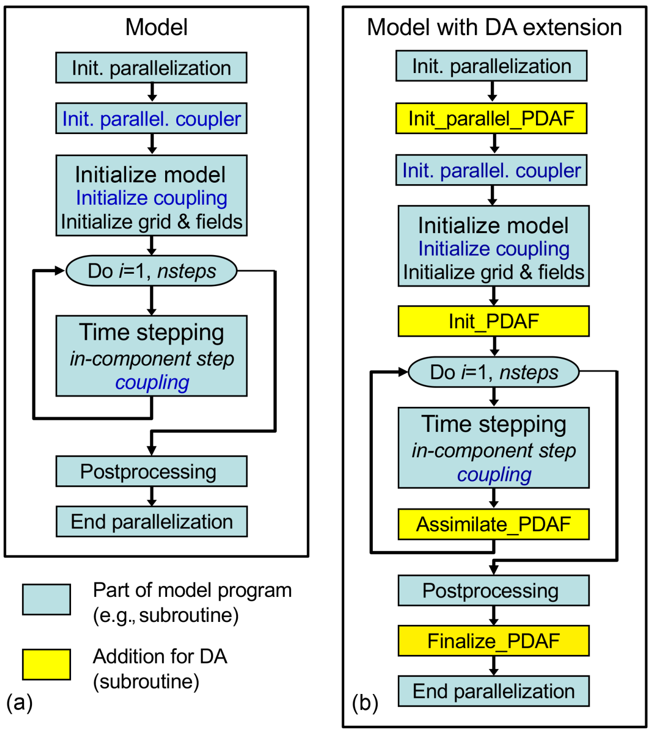

Figure 2. General program flow: **(a)** abstract original program without data assimilation; **(b)** program augmented for data assimilation. The blue color marks coupling routines whose parallelization needed to be adapted for the data assimilation. Each of the two coupled compartment models were augmented in this way.

scription for completeness. Further, we discuss the particularities of the coupled model.

Figure 2 shows the general program flow and the necessary extension of the code for adding the DA functionality. The different boxes can (but are not required to) be subroutine calls. The figure is valid for any of the two programs of the coupled model system. Without references to the coupler, it would also be valid for a single-compartment model.

The left-hand side of Fig. 2 (Fig. 2a) shows the typical flow of a coupled compartment model. Here, at the very beginning of the program, the parallelization is initialized ("init. parallelization"). After this step, all involved processes of the program are active (for the parallelization aspects see Sect. 3.3). Subsequently, the OASIS coupler initializes the parallelization for the coupled model by separating the processes for ECHAM and FESOM. Thus, after this point, the coupler can distinguish the different model compartments. Now, the model itself is initialized; e.g., the model grid for each compartment is initialized and the initial fields are read from files. Further, information for the coupling will be initialized like the grid configuration, which is required by the coupler to interpolate data in between the different model grids. This completes the model initialization; then the time stepping is computed. During time stepping, the coupler exchanges the interface information between the different compartments. After time stepping, some postprocessing can be performed, e.g., writing time averages or restart files to disk.

The right-hand side of Fig. 2 (Fig. 2b) shows the required additions to the model code as yellow boxes. These additions are calls to subroutines that interface between the model code and the DA framework. In this way, only single-line subroutine calls are added, which might be enclosed in preprocessor checks to allow users to activate or deactivate the data assimilation extension at compile time. The additions are done in the codes of both ECHAM and FESOM, and here we discuss them in general. The added subroutine calls have the following functionality:

- *Init_parallel_PDAF* modifies the parallelization of the model. Instead of integrating the state of a single model instance ("model task"), the model is modified to run an ensemble of model tasks. This routine is inserted directly after the parallelization is started. So all subsequent operations of the program will act in the modified parallelization. In the coupled model, this routine is executed before the parallelization of the coupler is initialized. In this way the coupler will also be initialized for an ensemble of model states.

- In *Init_PDAF*, the PDAF framework will be initialized. The routine is inserted into the model codes so that it is executed after all normal model initialization is completed, i.e., just before the time-stepping loop. The routine specifies parameters for the DA, which can be read from a configuration file. Then, the initialization routine for PDAF, named "PDAF_init" is called, which performs the PDAF-internal configuration and allocates the internal arrays, e.g., the array of the ensemble states. Further, the initial ensemble is read from input files. As this reading is model specific, it is performed by a user-provided routine that is called by PDAF as a callback routine (see Sect. 3.4). After the framework is initialized, the routine "PDAF_get_state" is called. This routine writes the information from the initial ensemble into the field arrays of the model. In addition, the length of the initial forecast phase, i.e., the number of time steps until the first analysis step, is initialized. For the coupled model, "PDAF_init" and "PDAF_get_state" are called in each compartment. However, some parameters are distinct: the time step size of ECHAM is 450 s, while it is 900 s for FESOM. Hence, the numbers of time steps in the forecast phase of 1 d are different in the compartments.

- *Assimilate_PDAF* is called at the end of each model time step. For this, it is inserted into the model codes of ECHAM and FESOM at the end of the time-stepping loop. The routine calls a filter-specific routine of PDAF that computes the analysis step of the selected filter method, e.g., "PDAF_assimilate_lestkf" for the localized ESTKF. This routine of PDAF also checks whether all time steps of a forecast phase have been computed. Only if this is true will the analysis step be executed; otherwise, the time stepping is continued. If additional operations for the DA are required during the time stepping, such as taking into account future observations in the case of the advanced equivalent-weights particle filter (EWPF; van Leeuwen, 2010) or collecting observed ensemble fields during the forecast phase for a four-dimensional filtering (Harlim and Hunt, 2007), these are also performed in this filter-specific routine. For the coupled model, the routine is called in both ECHAM and FESOM. Then, "PDAF_assimilate_lestkf" will check for the analysis time according to the individual number of time steps in the forecast phase. The analysis step will then be executed in each compartment according to the configuration of the assimilation. In the implementation of AWI-CM-PDAF 1.0, the analysis is only performed in FESOM. Thus, while "PDAF_assimilate_lestkf" is also called in ECHAM, it does not assimilate any data.

– *Finalize_PDAF* is called at the end of the program. The routine includes calls to the routine "PDAF_print_info", which print out information about execution times of different parts of the assimilation program as measured by PDAF as well as information about the memory allocated by PDAF.

Compared to the implementation strategy discussed in Nerger and Hiller (2013), in which the assimilation subroutine is only called after a defined number of time steps, this updated scheme allows users to perform DA operations during the time-stepping loop. To use this updated scheme, one has to execute the coupled model with enough processors so that all ensemble members can be run at the same time. This is nowadays easier than in the past, because the number of processor cores is much larger in current high-performance computers compared to the past.

Apart from the additional subroutine calls, a few changes were required in the source codes of ECHAM, FESOM, and OASIS3-MCT, which are related to the parallelization. These changes are discussed in Sect. 3.3.

3.3 Parallelization for coupled ensemble data assimilation

The modification of the model parallelization for ensemble DA is a core element of the DA online coupling. Here, the parallelization of AWI-CM and the required changes for the extension for the DA are described. For FESOM, as a single-compartment model, the adaption of the parallelization was described by Nerger et al. (2005) and Nerger and Hiller (2013). A similar parallelization was also described by Browne and Wilson (2015). For the online coupling of PDAF with the coupled model TerrSysMP, the setup of the parallelization was described by Kurtz et al. (2016). While for TerrSysMP, a different coupling strategy was used; the parallelization of the overall system is essentially the same as discussed here for AWI-CM. The parallelization for the DA is configured by the routine *init_parallel_pdaf*. In general this is a template routine, which can be adapted by the user according to particular needs. Nonetheless, by now the default setup in PDAF was directly usable in all single-compartment models to which PDAF was coupled. Compared to the default setup in PDAF for a single-compartment model, we have adapted the routine to account for the existence of two model compartments.

Like other large-scale models, AWI-CM is parallelized using the Message Passing Interface standard (MPI; Gropp et al., 1994). MPI allows one to compute a program using several processes with distributed memory. Thus, each process has only access to the data arrays that are allocated by this process. Data exchanges between processes are performed in the form of parallel communication; i.e., the data are explicitly sent by one process and received by another process. All parallel communication is performed within so-called communicators, which are groups of processes. When the parallel region of a program is initialized, the communicator MPI_COMM_WORLD is initialized, which contains all processes of the program. In the case of AWI-CM when the two executables for ECHAM and FESOM are jointly started, they share the same MPI_COMM_WORLD so that parallel communication between the processes running ECHAM and those running FESOM is possible. Further communicators can be defined by splitting MPI_COMM_WORLD. This is used to define groups of processes both for AWI-CM and for the extension with PDAF.

For AWI-CM without data assimilation extension, the parallelization is initialized by each program at the very beginning. Then, a routine of OASIS3-MCT is called which splits MPI_COMM_WORLD into two communicators: one for ECHAM (COMM_ECHAM) and one for FESOM (COMM_FESOM). These communicators are then used in each of the compartment models, and together they build one model task that integrates one realization of the coupled model state. MPI_COMM_WORLD is further used to define one process each for ECHAM and FESOM, which perform parallel communication to exchange flux information. Important here is that OASIS3-MCT is coded to use MPI_COMM_WORLD to define these communicators. Each of the compartment models then uses its group of processes for all compartment-internal operations. Each model uses a domain decomposition; i.e., each process computes a small region of the global domain in the atmosphere or the ocean. The distribution of the processes is exemplified in Fig. 3a for the case of six processes in MPI_COMM_WORLD. Here, the communicator is split into four processes for COMM_FESOM (green) and two for COMM_ECHAM (blue).

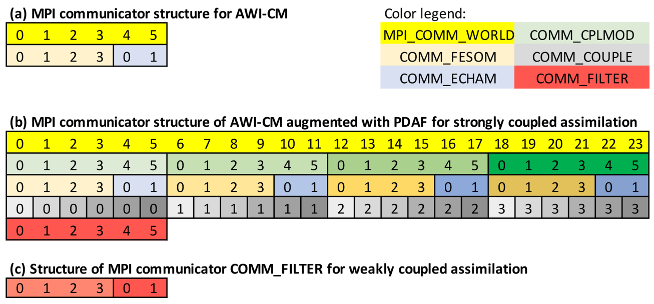

Figure 3. Example configuration of MPI communicators: **(a)** AWI-CM, **(b)** AWI-CM with PDAF extension for ensemble data assimilation. The colors and lines mark processes that are grouped as a communicator. Different shades of the same color mark the same communicator type (e.g., four orange communicators COMM_FESOM). For COMM_COUPLE, each communicator is spread over the model tasks. The numbers mark the rank index of a process in a communicator.

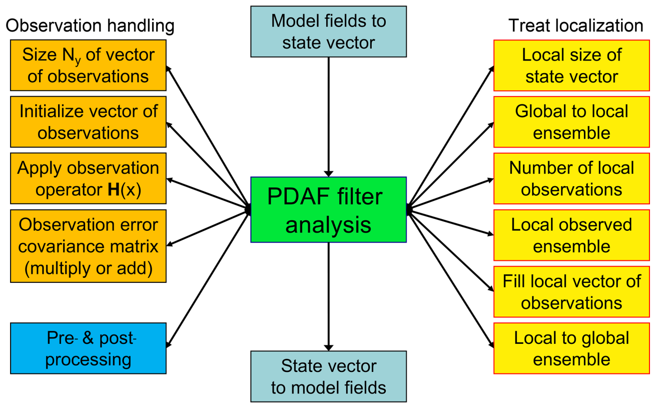
Observation handling
Size N_y of vector of observations
Initialize vector of observations
Apply observation operator **H**(x)
Observation error covariance matrix (multiply or add)
Pre- & post-processing
Model fields to state vector
PDAF filter analysis
State vector to model fields
Treat localization
Local size of state vector
Global to local ensemble
Number of local observations
Local observed ensemble
Fill local vector of observations
Local to global ensemble

Figure 4. PDAF filter analysis step and related call-back routines provided by the user. There are four types of routines: transfers between model fields and state vector (cyan), observation handling (orange), treatment of localization (yellow), and pre- and postprocessing (blue).

For the ensemble DA, the parallelization of AWI-CM is modified. Generally, the introduction of the ensemble adds one additional level of parallelization to a model, which allows one to concurrently compute the ensemble of model integrations, i.e., several concurrent model tasks. In AWI-CM augmented by the calls to PDAF, the routine *init_parallel_pdaf* modifies the parallelization. Namely, MPI_COMM_WORLD is split into a group of communicators for the coupled model tasks (COMM_CPLMOD), as exemplified for an ensemble of four model tasks in Fig. 3b indicated by the different color shading. Subsequently, OASIS3-MCT splits each communicator (COMM_CPLMOD) into a pair (COMM_ECHAM and COMM_FESOM) (third row in Fig. 3b). To be able to split COMM_CPLMOD, the source code of OASIS3-MCT needs to be modified by replacing MPI_COMM_WORLD by COMM_CPLMOD, because OASIS3-MCT uses MPI_COMM_WORLD as the basis for the communicator splitting (see also Kurtz et al., 2016, for the required modifications). With this configuration of the communicators, AWI-CM is able to integrate an ensemble of model states by computing all model tasks concurrently.

Two more communicators are defined in *init_parallel_pdaf* for the analysis step in PDAF. Here, a configuration is used that computes the filter analysis step on the first coupled model task using the same domain decomposition as the coupled model. Because the ESTKF (as any other ensemble Kalman filter) computes a combination of all ensemble members individually for each model grid point or for single vertical columns (Eq. 1), the ensemble information from all ensemble members is collected on the processes of the first model task, keeping the domain decomposition. For collecting the ensemble information, the communicator COMM_COUPLE groups all processes that compute the same subdomain in the coupled model. Thus, all processes that have the same rank index in, for example, COMM_FESOM are grouped into one communicator as shown in row 4 of Fig. 3b. Finally, the communicator COMM_FILTER (row 5 of Fig. 3b) is defined, which contains all processes of the first model task. Note that compared to the single-compartment case discussed in Nerger et al. (2005) and Nerger and Hiller (2013), the major change is that each model task is split into the communicators COMM_FESOM and COMM_ECHAM, which are, however, only used for the model integration. In addition, COMM_FILTER includes the processes of both model compartments of the first model task.

This configuration is used to perform strongly coupled DA, because it allows for the communication between processes of ECHAM with processes of FESOM. In a weakly coupled application of DA, COMM_FILTER is initialized so that two separate communicators are created: one for all subdomains of FESOM and another one for all subdomains of ECHAM as shown in Fig. 3c. In practice one can achieve this by using the already defined communicators COMM_FESOM and COMM_ECHAM of model task 1. Because these two communicators are initialized after executing *init_parallel_pdaf*, one has to overwrite COMM_FILTER afterwards in, for example, *init_PDAF*. With this configuration, the assimilation can be performed independently for both compartments.

3.4 Call-back routines for handling of model fields and observations

The call-back routines are called by PDAF to perform operations that are specific to the model or the observations. The operations performed in each routine are rather elementary to keep the complexity of the routines low. There are four different types of routines, which are displayed in Fig. 4:

- For *interfacing model fields and state vector* (cyan), there are two routines called before and after the analysis step. The first routine writes model fields into the state vector of PDAF, while the second initializes model fields from the state vector. These routines are executed by all processes that participate in the model integrations, and each routine acts on its process subdomain. For the coupled model, there are different routines for FESOM and ECHAM.

- The *observation handling* (orange) performs operations related to the observations. For example, a routine provides PDAF with the number of observations, which is obtained by reading the available observations and counting them. This routine allows PDAF to allocate arrays for the observed ensemble. Another routine is the implementation of the observation operator. Here, the routine is provided with a state vector $\boldsymbol{x}$ from the ensemble and has to return the observed state vector, i.e., $\mathbf{H}(\boldsymbol{x})$. For the coupled model, the routines are distinct for FESOM and ECHAM as, for example, the observation operator for an oceanic observation can only be applied in FESOM. For strongly coupled DA, the observation operator routine would also contain parallel communication that acts across the compartments. Thus, after obtaining the observations in a compartment, a cross-compartment observation vector is initialized using MPI communication.

- For *localization* (yellow), the localized analysis described in Sect. 2.1.1 requires several operations, which are provided by call-back routines. For example, a call-back routine needs to determine the dimension of a local state vector. For a single grid point, this would be the number of variables stored at this grid point. For a vertical column of the model grid, this would be the number of three-dimensional model fields times the number of model layers plus the number of two-dimensional model fields (like sea surface height or sea ice variables in FESOM). Then, after PDAF allocates the local state ensemble, a call-back routine is used to fill the local states from the full domain-decomposed state vector. (Likewise, there is a routine that writes a local state vector after the local analysis correction into the full state vector.) In addition, there is a routine that determines the number of observations within the influence radius around the vertical column and a routine to fill this local observation vector from a full observation vector.

- For *pre- and postprocessing* (blue), there is a pre- and postprocessing routine to give the user access to the ensemble before and after the analysis step. Here, one typically computes the ensemble mean and writes it into a file. Further, one could implement consistency checks, e.g., whether concentration variables have to be positive, and can perform a correction to the state variables if this is not fulfilled.

4 Parallel performance of the coupled data assimilation system

4.1 Scalability

To assess the parallel performance of the assimilation system described above, AWI-CM is run here in the same global configuration as described by Sidorenko et al. (2015). The atmosphere uses a horizontal spectral resolution (T63, about 180 km) with 47 layers. The ocean model uses an unstructured triangular grid with 46 vertical layers. The horizontal resolution varies between 160 km in the open ocean, with a refinement to about 45 km in the equatorial region and close to the Antarctic continent, and 30 km north of 50° N. The models are run with a time step size of 450 s for ECHAM and 900 s for FESOM. The coupling by OASIS3-MCT is performed hourly.

In the initial implementation (AWI-CM-PDAF 1.0), the assimilation update is only performed as weakly coupled DA in the ocean compartment. The state vector for the assimilation is composed of the two-dimensional sea surface height, the three-dimensional model fields temperature, salinity, and the three velocity components. The DA is started on 1 January 2016, and satellite observations of the sea surface temperature obtained from the European Copernicus initiative (data set SST_GLO_SST_L3S_NRT_OBSERVATIONS_010_010 available at https://marine.copernicus.eu, last access: 14 September 2020), interpolated to the model grid, are assimilated daily. The assimilation is multivariate so that the SST (sea surface temperature) observations influence the full oceanic model state vector through the ensemble estimated cross-covariances that are used in the ESTKF. The initial ensemble was generated using second-order exact sampling (Pham et al., 1998) from the model variability of snapshots at each 5th day over 1 year. The ensemble mean was set to a model state for 1 January 2016 from a historical (climate) run of AWI-CM. No inflation was required in this experiment; i.e., a forgetting factor $\rho = 1.0$ (see Eq. 4) was used. Even though we only perform weakly coupled DA here, we expect that the compute performance would be similar in the case of strongly coupled DA, as is explained in Sect. 6.

For a fixed ensemble size but varying number of processes for ECHAM and FESOM, the scalability of the program is determined by the scalability of the models (see Nerger and Hiller, 2013). To access the scalability of the assimilation system for varying ensemble size, experiments over 10 d were conducted with varying ensemble sizes between $N_\mathrm{e} = 2$ and $N_\mathrm{e} = 46$. The length of these experiments is chosen to be long enough so that the execution time is representative to assess the scalability. However, the assimilation effect will be rather small for these 10 analysis steps. The number of processes for each model task was kept constant at 72 processes for ECHAM and 192 processes for the more costly FESOM. The experiments were conducted on the Cray XC40 system "Konrad" of the North-German Supercomputer Alliance (HLRN).

Figure 5 shows the execution times per model day for different parts of the assimilation program. Shown are the times for 24 h forecast phases including the time to collect and distribute the ensemble (DA coupling within the communicator COMM_COUPLE) for the analysis step. Also shown are the times for the analysis step (green), the execution of the pre- and post-step operations (red), and the DA coupling time (blue). The crosses show the times for each model task separately for the atmosphere and ocean; thus, there are $2N_\mathrm{e}$ black and blue crosses for each ensemble size. The blue and black lines show the maximum execution times. The overall execution time is dominated by the time to compute the forecasts. The combined time for the analysis and the pre- and post-step operations is only between 4 % and 7 % of the forecast time. For a given ensemble size, the black crosses show that the execution times for the forecast on the different model tasks vary. In the experiments, the longest forecast time was up to 16 % longer than the shortest time, which occurred for $N_\mathrm{e} = 24$. This variability is partly caused by the time for DA coupling (see discussion below) but also by the fact that the semi-implicit time stepping of FESOM leads to varying execution times. Further influences are the parallel communication within each compartment at each time step and the communication for the model coupling by OASIS3-MCT that is performed at each model hour. The execution time for these operations depends on how the overall program is distributed over the computer. As the computer is also used by other applications, it is likely that the application is widely spread over the computer so that even different compute racks are used. This can even lead to the situation when the processors for a single coupled model task of ECHAM and FESOM (but also a single model instance of ECHAM or FESOM) are not placed close to each other. If the processors are distant, e.g., in different racks, the communication over the network will be slower than for a compact placement of the processors. To this end, also the execution time will vary when an experiment for the same ensemble size is repeated. Nonetheless, repeated experiments showed that the timings in Fig. 5 are representative. Likewise, experiments in the new supercomputer system "Lise" of the HLRN showed similar timings, though the forecast time was reduced to about 27 s per model day compared to about 35 s shown in Fig. 5.

The variation of the forecast time when the ensemble size is changed is mainly caused by the varying time for the DA coupling. When the time for the DA coupling is subtracted from the forecast time, the variability is much reduced as the black dashed line shows. The variability in the dependence on the ensemble size is better visible when the execution time is normalized relative to the time for $N_\mathrm{e} = 2$, as is displayed in Fig. 6. The forecast time including DA coupling fluctuates and increases by up to 8 % for the largest ensemble with $N_\mathrm{e} = 46$ (black line). In contrast, the forecast time without DA coupling only increases by about 3.5 % (black dashed line). The time for the DA coupling (blue line) varies by a factor of 2.5. This large variation is due to the fact that here the communication happens in the communicators COMM_COUPLE, which are spread much wider over the computer than the communicators for each coupled model task (COMM_CPLMOD), as is visible in Fig. 3. However, even though the number of ensemble states to be gathered and scattered in the communication for the DA coupling varies between 2 and 46, there is no obvious systematic increase in the execution time. In particular, for $N_\mathrm{e} = 40$ the execution time is almost identical to that of $N_\mathrm{e} = 2$.

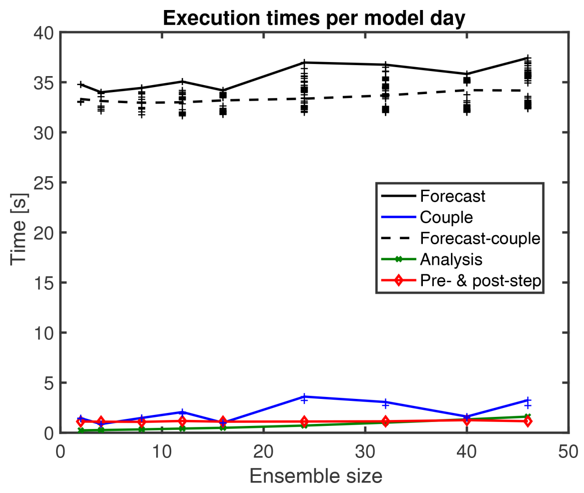

Figure 5. Execution times per model day for varying ensemble sizes for different parts of the assimilation program. The dominating forecast time includes the "coupling" time, which results in the time variations.

Further variation in the dependence on the ensemble size is visible for the pre- and post-step operations (red line). This variation is mainly due to the operations for writing the ensemble mean state into a file. In contrast, the analysis step shows a systematic time increase. The time for computing the analysis for $N_\mathrm{e} = 46$ is about 7 times as long as for $N_\mathrm{e} = 2$. This is expected from the computational complexity of the LESTKF algorithm (see Vetra-Carvalho et al., 2018). However, the LESTKF also performs MPI communication for gathering the observational information from different process domains. When repeating experiments with the same ensemble size, we found a variation of the execution time for the analysis step of up to 10 %.

4.2 Performance tuning

To obtain the scalability discussed above, important optimization steps have been performed. First, it is important

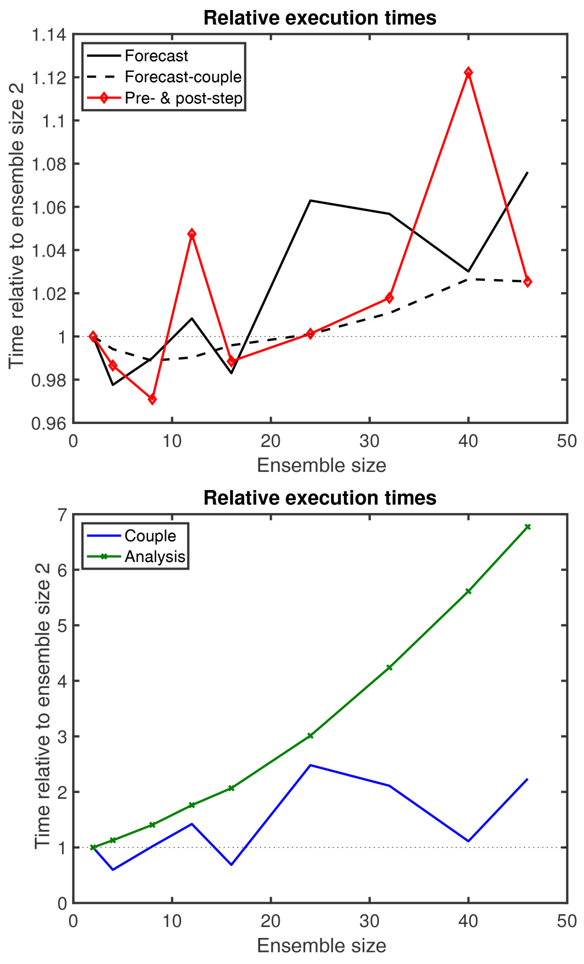

Figure 6. Execution times relative to ensemble size 2 for different parts of the assimilation program as a function of the ensemble size. The fluctuation in the time is caused by parallel communication and file operations. The analysis step shows a systematic time increase, while the time for DA coupling varies strongly.

that each coupled model instance is, as far as possible, placed compactly in the computer. Second, one has to carefully consider the disk operations performed by the ensemble of coupled model tasks.

For the first aspect, one has to adapt the run script. The coupled model is usually started with a command line like

```
mpirun -np  N_O fesom.x : -np N_A\
 echam.x
```

(or any other suitable starter for an MPI-parallel program) such that FESOM and ECHAM are run using N_O and N_A processes, respectively. For the DA, one could simply change this by replacing N_O by $N_e \times N_O$ and N_A by $N_e \times N_A$ to provide enough processes to run the ensemble. This is analogous to the approach used when running a single-compartment model. However, changing the command line in this way will first place all MPI tasks for the FESOM ensemble in the computer followed by all MPI tasks for the ECHAM ensemble. Accordingly, each ocean model will be placed distant from the atmospheric model to which it is coupled. Using this execution approach, the time for the forecasts discussed above increased by a factor of 4, when the ensemble size was increased from 2 to 46. For a more efficient execution, one has to ensure that the ocean–atmosphere pairs are placed close to each other. This is achieved with a command line such as

```
mpirun -np  N_O fesom.x : -np N_A  \
 echam.x: -np  N_O fesom.x  -np N_A\
 echam.x...
```

which contains as many FESOM-ECHAM pairs as there are ensemble members. With this approach, the time increase of the forecast was reduced to about 40 % for the increase from $N_e = 2$ to $N_e = 46$.

For the second issue regarding disk operations, one has to take into account that the direct outputs written by each coupled ensemble task are usually not relevant, because the assimilation focuses on the ensemble mean state. To this end, one generally wants to deactivate the outputs written by the individual models and replace them by outputs written by the pre- and post-step routine called by PDAF. If the model does not allow us to fully switch off the file output, it usually helps to set the output interval of a model to a high value (e.g., a year for a year-long assimilation experiment). However, in the case of AWI-CM this strategy still resulted in conflicts of the input/output operations so that the models from the different ensemble tasks tried to write into the same files, which serialized these operations and increased the execution time. To avoid these conflicts, it helped to distribute the execution of the different ensemble tasks to different directories, e.g.,

```
mpirun -np  N_O 01/fesom.x : -np N_A\
 01/echam.x : -np  02/N_O fesom.x :\
 -np N_A  02/echam.x...
```

combined with a prior operation in the run script to generate the directories and distribute the model executables and input files. This distribution avoids the case when two model tasks write into the same file and improves the performance of the ensemble DA application. In this configuration, the performance results of Sect. 4.1 were obtained. Another benefit of separate execution directories is that ensemble restarts can be easily realized. Given that each model task writes its own restart files in a separate directory, a model restart is possible from these files without any adaptions to the model code. Note that the approach of separate directories is also possible for the ensemble DA in the case of a single (uncoupled) model like a FESOM-only simulation using atmospheric forcing data as, for example, applied by Androsov et al. (2019).

5 Application example

Applications based on the AWI-CM-PDAF 1.0 code are given in Mu et al. (2020), where the focus is on the effect on sea ice, and Tang et al. (2020), who discuss the reaction of the atmosphere on assimilating ocean observations.

Here, we demonstrate the functionality of the data assimilation system in an experiment assimilating SST data over the year 2016. An ensemble of 46 states is used, which is the maximum size used in the scalability experiment discussed above. The assimilation is performed with a localization radius of 500 km using the regulated localization function by Nerger et al. (2012a). The same SST observations as in Sect. 4.1 are assimilated, which are treated as in Tang et al. (2020). The resolution of the observations is 0.1° and hence higher than the resolution of the model in most regions. Since the model grid is unstructured with varying resolution, super-observations are generated by averaging onto the model grid. The observation error standard deviation for the assimilation was set to 0.8 °C, and observations whose difference from the ensemble mean is more than 2 standard deviations are excluded from the assimilation. This approach excludes about 22 % of the observations at the initial first analysis step. The number of excluded observations shrinks during the course of the assimilation and after 1 month less than 5 % of the days observations are excluded. The assimilation further excludes observations at grid points for which the model contains sea ice because of the mismatch of the satellite data representing ice-free conditions, while ice is present on modeled ocean surface. Two experiments are performed: the experiment FREE runs the ensemble without assimilating observations, while the experiment DA-SST assimilates the SST data.

Figure 7 shows the root-mean-square error (RMSE) of the SST in the analysis step with respect to the assimilated observations over time. Given that the SST observations are assimilated, it is a necessary condition for the DA to reduce the deviation from these observations. At the initial analysis time (i.e., after 24 h), the RMSE is about 1.2 °C. In the free run, the RMSE increases first to about 1.4 °C and reaches nearly 1.6 °C a the end of the year. The assimilation in DA-SST strongly reduces the RMSE during the first 2 months. During this initial transient phase, the RMSE is reduced to about 0.45 °C. Afterwards, the RMSE remains nearly constant, which is a typical behavior. On average over the year 2016, the RMSE in the experiment DA-SST is 0.51 °C, while it is 1.38 °C for the free run.

To validate the assimilation with independent observations, temperature and salinity profiles from the EN4 data set (EN4.2.1) of the UK Met Office (Good et al., 2013) are used. This collection of in situ data contains about 1000 to 2000 profiles per day at depths between the surface and 5000 m depth. Figure 8 shows the RMSE of the experiment DA-SST relative to the RMSE for the free run. Hence values below one indicate improvements. For the temperature, a gradual improvement is visible during the first 100 d. The error reduction reaches about 40 % during the year. On average, the RMSE is reduced by 24 % from 1.85 to 1.40 °C. The variations in the RMSE, e.g., the elevated values around day 250,

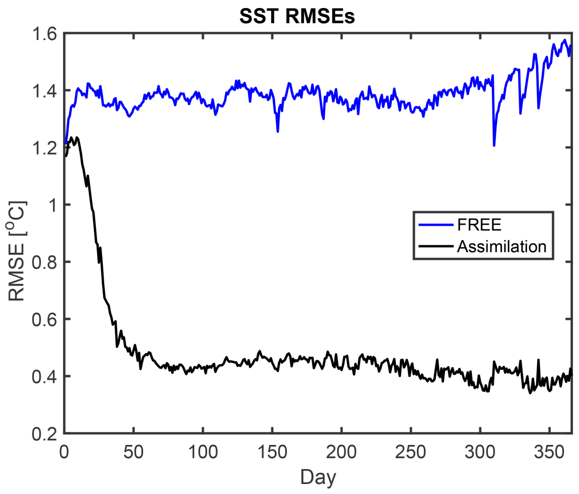

Figure 7. Root-mean-square errors of the estimated SST with regard to the assimilated SST observations. Shown are the free run (blue) and the SST assimilation experiment (black). During the spin-up period of the DA, the RMSEs are strongly reduced.

are due to the varying coverage and location of the profiles in the EN4 data set. For the salinity, the effect of the DA is lower. While the RMSE of the salinity first increases during the first month, it is reduced from day 60, but until day 140 it is sometimes larger than at the initial time. The RMSE is partly reduced by up to 23 % at day 144. On average, over the full year of the experiment, the RMSE of salinity is reduced by 5.6 %. This smaller effect on the salinity is expected, because there are no strong correlations between the SST and the salinity at different depths. The improvements of the model fields by the DA of SST are mainly located in the upper 200 m of the ocean. For the temperature, the RMSE is reduced by 15.2 % in the upper 200 m but only 3.0 % below 200 m. This is also an expected effect, because the correlations between SST and subsurface temperature are largest in the mixed layer of the ocean.

6 Discussion

The good scalability of the assimilation system allows us to perform the assimilation experiment of Sect. 5 over one full year with daily assimilation in slightly less than 4 h, corresponding to about 53 000 core hours. As such, the system is significantly faster than the coupled ensemble DA application by Karspeck et al. (2018), who reported an experiment to complete 1 year in 3 to 6 weeks with an ensemble of 30 states and about 1×10^6 core hours per simulation year. However, both systems are not directly comparable. Karspeck et al. (2018) used atmospheric and ocean models with 1° resolution. Thus, their atmosphere had a higher resolution than used here, while the ocean resolution was comparable to the coarse FESOM resolution in the open ocean, which

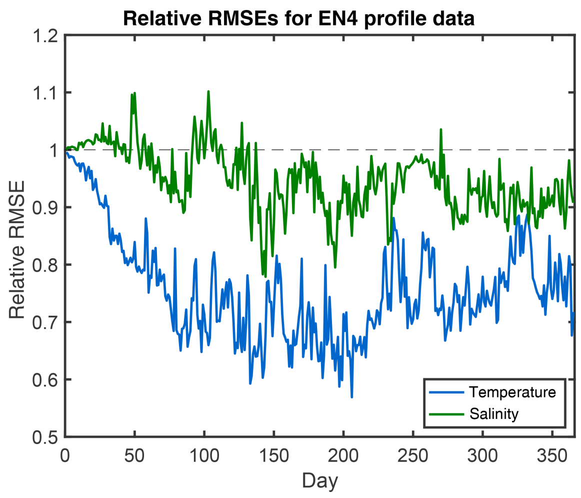

Figure 8. Root mean square errors (RMSEs) of the assimilation experiment relative to the free run computed with regard to the in situ EN4 profile observations. Shown are the relative RMSEs for temperature (blue) and salinity (green). Temperature is more strongly improved than salinity.

was then regionally refined. Given that both model compartments in AWI-CM scale to larger processor numbers than we used for the DA experiment, we expect that the DA in AWI-CM with ECHAM at a resolution of T127 (i.e., about 1°) could be run at a similar execution time as for T63 given that a higher number of processors would be used. Further, Karspeck et al. (2018) also applied the DA in the atmosphere, while here only oceanic data were assimilated. Given that the atmospheric analysis step would typically be applied after each sixth hour, the time for the DA coupling and the analysis steps would increase. However, we do not expect that a single atmospheric analysis step would require significantly more time than the ocean DA, so due to the parallelization, the overall run time should not increase by more than 10 %–20 %. Further, we expect a similar scalability in the case of strongly coupled DA. The major change for strongly coupled DA is to communicate the observations in between the compartments as mentioned above. This communication will only be a small part of the analysis time.

Important for the online-coupled assimilation system is that there is obviously no significant time required for redistributing the model field (i.e., the time for the DA coupling discussed in Sect. 4.1). Furthermore, there is no transpose of the ensemble array to be performed, which was reported to be costly by Karspeck et al. (2018). Here, the implementation of the analysis step uses the same domain decomposition as the models; hence, only the full ensemble for each process subdomain has to be collected by the DA coupling. Thus, only groups of up to 46 processes communicate with each other in this step.

The online-coupled assimilation system avoids any need for frequent model restarts. Actually, the initial model startup of AWI-CM took about 95 s and the finalization of the model with writing restart files took another 15 s. Thus, these operations take about 3.3 times longer than integrating the coupled model for 1 d. If the DA would be performed in a separate program coupled to AWI-CM through files, these operations would be required each model day. In addition, the assimilation program would also need to read these restart files and write new restart files after the analysis step. Assuming that these observations take about 15 s, such as the finalization of the coupled model, the execution time would increase by a factor of 4 for offline-coupled DA compared to online-coupled DA.

The code structure using interface routines inserted into the model code and case-specific call-back routines makes the assimilation framework highly flexible. Further, the abstraction in the analysis step, which uses only state and observation vectors without accounting for the physical fields, allows one to separate the development of advanced DA algorithms from the development of the model. Thus, a separation of concerns is ensured, which is mandated for efficient development of complex model codes and their adaptions to modern computers (Lawrence et al., 2018). The separation allows all users (as soon as a new DA method is implemented) with their variety of models to use this method by updating the PDAF library. To ensure compatibility of different versions of the library, the interfaces to the PDAF routines are kept unchanged. However, for a new filter, additional call-back routines might be required, e.g., a routine to compute the likelihood of an ensemble according to the available observations in the case of the nonlinear ensemble transform filter (NETF; Tödter and Ahrens, 2015) or a particle filter. The abstraction in the analysis step and the model-agnostic code structure also allow users to apply the assimilation framework independently of the specific research domain. For example, applications of PDAF with a geodynamo model (Fournier et al., 2013), hydrological applications (Kurtz et al., 2016), ice shield modeling (Gillet-Chaulet, 2020), and volcanic ash clouds (Pardini et al., 2020) have been published.

The example here uses a parallelization in which the analysis step is computed using the first model task and the same domain decomposition as the model. Other parallel configurations are possible. For example, one could compute the analysis step not only by using the processes of model task 1 but also by using processes of several or all model tasks. This could be done by either using a finer domain decomposition than in the model integrations or by distributing different model fields onto the processes. These alternative parallelization strategies are, however, more complex to implement and hence not the default in PDAF. A further alternative, which is already supported by PDAF, is to dedicate a set of processes for the analysis step. In this case, the DA coupling would communicate all ensemble members to these separate processes. However, these processes would idle during the forecast phase. To this end, separating the processes for the analysis step would mainly be a choice if the available memory on the first model task is not sufficient to execute the analysis step. Also in this case, the distribution of the analysis step over several processors would reduce the required memory. For the parallel configuration of AWI-CM-PDAF in Fig. 3, a particular order of the processes is assumed. This order originates from the startup procedure of MPI and is determined by the command line which starts the program. Thus, for other models one might need a different setup, which can usually be obtained by only modifying the routine *init_parallel_pdaf*. Further, the default version of this routine splits the communicator MPI_COMM_WORLD. However, for other models a different suitable communicator might be split if not all processes participate in the time stepping. This can be the case when, for example, an input/output (OI) server is used that reserves processes exclusively for the file operations. To provide flexibility to adapt to such requirements, the routine *init_parallel_pdaf* is compiled with the model and is not part of the core routines of the PDAF library.

While the fully parallel execution of the assimilation program is very efficient, it is limited by the overall job size allowed on the computer. The maximum ensemble size was here limited by the batch job size of the used computer. The model used in the example here can scale even further than the 192 processes used for FESOM and the 72 processes for ECHAM. Thus, using the same computer, one could run a larger ensemble, with less processes per model and accordingly a larger run time, or a smaller ensemble with less run time. The number of processes should be set so that the requirements on the ensemble size for a successful assimilation can be fulfilled. Nonetheless, the ensemble DA is computationally demanding; for larger applications, one might need to obtain a compute allocation at larger computing sites, such as national compute centers.

7 Conclusions

This study discussed the Parallel Data Assimilation Framework (PDAF) and its use to create a coupled data assimilation program by augmenting the code of a coupled model and using in-memory data transfers between the model and the data assimilation software. The implementation strategy was exemplified for the coupled ocean–atmosphere model AWI-CM for which two separate programs for the ocean and atmosphere were augmented. However, the strategy can be easily used for other model systems consisting of a single or multiple executables.

The implementation of a DA system based on PDAF consists of augmenting the model codes with calls to routines of the assimilation framework. These routines modify the parallelization of the model system so that it becomes an ensemble model. Further, the ensemble is initialized and the analysis step of the data assimilation can be executed at any time without restarting the model. Operations to transfer between model fields and the abstract state vector of the assimilation (and the observation handling) are performed in the case-specific routines. These routines are executed as call-back routines and can be implemented like routines of the numerical model, which should simplify their implementation.

Numerical experiments with daily assimilation of sea surface temperature observations into the AWI-CM showed an excellent scalability when the ensemble size was increased. This resulted in an overhead which was, depending on the ensemble size, only up to 15 % in computing time compared to the model without assimilation functionality. The execution time of the coupled ensemble data assimilation program was dominated by the time to compute the ensemble integrations in between the time instances at which the observations are assimilated. This excellent scalability resulted from avoiding disk operations by keeping the ensemble information in memory and exchanging it through parallel communication during the run time of the program. Care has to be taken that in the coupled model the pairs of atmosphere and ocean model compartments are placed close to each other in the computer, which can be achieved by specifying these pairs in the command starting the parallel program. The time to collect this ensemble information before the analysis step and to distribute it afterwards showed significant variations from run to run. These variations are due to the fact that the large compute application is widely spread over processors of the computer. In any case, no systematic time increase was observed when the ensemble size was increased, and the time was only up to about 6 % of the time required for the forecasting. Distributing the different models over separate directories improved the scalability, because it avoided possible conflicts for the file handling, which can be serialized by the operating system of the computer.

PDAF provides a model-agnostic framework for the efficient data assimilation system as well as filter and smoother algorithms. As such, it provides the capacity to ensure a separation of concerns between the developments in the model, observations, and the assimilation algorithms. Functionality to interface between the model, which operates on physical fields, and the assimilation code, which only works on abstract state vectors, has to be provided in a case-specific manner by the users based on code templates. This also holds for the observation handling. While there are typical observational data sets for the different Earth system compartments, the observation operator links the observations with the model fields on the model grid. Thus, the observation operator has to be implemented by taking into account the specific character of the model grid such as the unstructured structure of FESOM's grid.

The current implementation of AWI-CM-PDAF only contains assimilation for the ocean component, while assimilation for the atmosphere is technically prepared. First studies (Mu et al., 2020; Tang et al., 2020) based on this implementation have been published. In future work, we plan to add the assimilation of atmospheric observations and to complete the implementation of strongly coupled data assimilation, which requires the exchange of observations in between the ocean and atmosphere.

Code availability. The model binding for AWI-CM-PDAF 1.0 used in this study is archived at Zenodo (https://doi.org/10.5281/zenodo.3822030; Nerger et al., 2019a) The PDAF code (version 1.14 was used here), as well as a full code documentation and a usage tutorial, is available at http://pdaf.awi.de (last access: 14 September 2020). The source code of the coupled AWI-CM is available from the model developers via https://fesom.de (last access: 14 September 2020). The ECHAM6 source code is maintained by the Max Planck Institute for Meteorology and is freely available to the public (http://www.mpimet.mpg.de/en/science/models/mpi-esm/echam/; Max Planck Institute for Meteorology, 2019a). External access to the ECHAM6 model is provided through their licensing procedure (https://mpimet.mpg.de/en/science/models/availability-licenses, last access: 14 September 2020). Only after registering for using ECHAM6 can access to AWI-CM be granted. The OASIS3-MCT coupler is available for download at https://portal.enes.org/oasis (ENES Portal, 2011).

Data availability. The experiments have been performed using the "LR" mesh of FESOM. For the availability of the configuration, mesh, and input files, see Rackow et al. (2019). The output files containing the timing information, the outputs from the 1-year experiments, and plotting scripts are available at Zenodo (https://doi.org/10.5281/zenodo.3823816, Nerger et al., 2019b).

Author contributions. The main body of the article was written by LN, with inputs from the co-authors. LN also leads the development of PDAF and developed the assimilation coupling strategy. QT implemented PDAF with AWI-CM and performed the timing experiments. Both QT and LM worked on optimizing the compute performance of the implementation of PDAF with AWI-CM. LM further developed the restarting functionality based on separate run directories.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank the North-German Supercomputing Alliance (HLRN) for providing compute resources (project hbk00064). We are grateful to Dmitry Sidorenko for support in the setup of the AWI-CM. We would like to thank the two anonymous reviewers and the editor for their comments that helped to improve the article.

Financial support. This research has been supported by the Helmholtz Association (project ESM – Advanced Earth System Modeling Capacity).

The article processing charges for this open-access publication were covered by a Research Centre of the Helmholtz Association.

Review statement. This paper was edited by James Annan and reviewed by two anonymous referees.

References

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., and Arellano, A.: The Data Assimilation Research Testbed: A Community Facility, B. Am. Meteorol. Soc., 90, 1283–1296, 2009.

Androsov, A., Nerger, L., Schnur, R., Schröter, J., Albertella, A., Rummel, R., Savcenko, R., Bosch, W., Skachko, S., and Danilov, S.: On the assimilation of absolute geodetic dynamics topography in a global ocean model: Impact on the deep ocean state, J. Geodesy, 93, 141–157, 2019.

Browne, P. A. and Wilson, S.: A simple method for integrating a complex model into an ensemble data assimilation system using MPI, Environ. Modell. Softw., 68, 122–128, 2015.

Browne, P. A., de Rosnay, P., Zuo, H., Bennett, A., and Dawson, A.: Weakly coupled ocean–atmosphere data assimilation in the ECMWF NWP system, Remote Sensing, 11, 234, https://doi.org/10.3390/rs11030234, 2019.

Burgers, G., van Leeuwen, P. J., and Evensen, G.: On the Analysis Scheme in the Ensemble Kalman Filter, Mon. Weather Rev., 126, 1719–1724, 1998.

Chang, Y.-S., Zhang, S., Rosati, A., Delworth, T. L., and Stern, W. F.: An assessment of oceanic variability for 1960–2010 from the GFDL ensemble coupled data assimilation, Clim. Dynam., 40, 775–803, 2013.

Danilov, S., Kivman, G., and Schröter, J.: A finite-element ocean model: Principles and evaluation, Ocean Model., 6, 125–150, 2004.

Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res., 99, 10143–10162, 1994.

Fournier, A., Nerger, L., and Aubert, J.: An ensemble Kalman filter for the time-dependent analysis of the geomagnetic field, Geochemistry Geophysics Geosystems, 14, 4035–4043, 2013.

Frolov, S., Bishop, C. H., Holt, T., Cummings, J., and Kuhl, D.: Facilitating strongly coupled ocean-atmosphere data assimilation with an interface solver, Mon. Weather Rev., 144, 3–20, 2016.

Gaspari, G. and Cohn, S. E.: Construction of Correlation Functions in Two and Three Dimensions, Q. J. Roy. Meteor. Soc., 125, 723–757, 1999.

Gillet-Chaulet, F.: Assimilation of surface observations in a transient marine ice sheet model using an ensemble Kalman filter, The Cryosphere, 14, 811–832, https://doi.org/10.5194/tc-14-811-2020, 2020.

Good, S. A., Martin, M. J., and Rayner, N. A.: EN4: quality controlled ocean temperature and salinity profiles and monthly objective analyses with uncertainty estimates, J. Geophys. Res.-Oceans, 118, 6704–6716, 2013.

Goodliff, M., Bruening, T., Schwichtenberg, F., Li, X., Lindenthal, A., Lorkowski, I., and Nerger, L.: Temperature assimilation into a coastal ocean-biogeochemical model: Assessment of weakly- and strongly-coupled data assimilation, Ocean Dynam., 69, 1217–1237, 2019.

Gropp, W., Lusk, E., and Skjellum, A.: Using MPI: Portable Parallel Programming with the Message-Passing Interface, The MIT Press, Cambridge, Massachusetts, 1994.

Han, G., Wu, X., Zhang, S., Liu, Z., and Li, W.: Error Covariance Estimation for Coupled Data Assimilation Using a Lorenz Atmosphere and a Simple Pycnocline Ocean Model, J. Climate, 26, 10218–10231, 2013.

Harlim, J. and Hunt, B. R.: Four-dimensional local ensemble transform Kalmn filter: numerical experiments with a global corculation model, Tellus, 59A, 731–748, 2007.

Houtekamer, P. L. and Mitchell, H. L.: Data Assimilation Using an Ensemble Kalman Filter Technique, Mon. Weather Rev., 126, 796–811, 1998.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I.: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter, Physica D, 230, 112–126, 2007.

Karspeck, A. R., Danabasoglu, G., Anderson, J., Karol, S., Collins, N., Vertenstein, M., Raeder, K., Hoar, T., Neale, R., Edwards, J., and Craig, A.: A global coupled ensemble data assimilation system using the Community Earth System Model and the Data Assimilation Research Testbed, Q. J. Roy. Meteor. Soc., 144, 2404–2430, https://doi.org/10.1002/qj.3308, 2018.

Kirchgessner, P., Toedter, J., Ahrens, B., and Nerger, L.: The smoother extension of the nonlinear ensemble transform filter, Tellus A, 69, 1327766, https://doi.org/10.1080/16000870.2017.1327766, 2017.

Kunii, M., Ito, K., and Wada, A.: Preliminary Test of a Data Assimilation System with a Regional High-Resolution Atmosphere-Ocean Coupled Model Based on an Ensemble Kalman Filter, Mon. Weather Rev., 145, 565–581, 2017.

Kurtz, W., He, G., Kollet, S. J., Maxwell, R. M., Vereecken, H., and Hendricks Franssen, H.-J.: TerrSysMP–PDAF (version 1.0): a modular high-performance data assimilation framework for an integrated land surface–subsurface model, Geosci. Model Dev., 9, 1341–1360, https://doi.org/10.5194/gmd-9-1341-2016, 2016.

Laloyaux, P., Balmaseda, M., Dee, D., Mogensen, K., and Janssen, P.: A coupled data assimilation system for climate reanalysis, Q. J. Roy. Meteor. Soc., 142, 65–78, 2016.

Lawrence, B. N., Rezny, M., Budich, R., Bauer, P., Behrens, J., Carter, M., Deconinck, W., Ford, R., Maynard, C., Mullerworth, S., Osuna, C., Porter, A., Serradell, K., Valcke, S., Wedi, N., and Wilson, S.: Crossing the chasm: how to develop weather and climate models for next generation computers?, Geosci. Model Dev., 11, 1799–1821, https://doi.org/10.5194/gmd-11-1799-2018, 2018.

Lea, D. J., Mirouze, I., Martin, M. J., King, R. R., Hines, A., Walters, D., and Thurlow, M.: Assessing a new coupled data assimilation system based on the Met Office coupled atmosphere-land-ocean-sea ice model, Mon. Weather Rev., 143, 4678–4694, 2015.

Liu, Z., Wu, S., Zhang, S., Liu, Y., and Rong, X.: Ensemble data assimilation in a simple coupled climate model: The role of ocean-atmopshere interaction, Adv. Atmos. Sci., 30, 1235–1248, 2013.

Mu, L., Yang, Q., Losch, M., Losa, S. N., RIcker, R., Nerger, L., and Liang, X.: Improving sea ice thickness estimates by assimilating CryoSat-2 and SMOS sea ice thickness data simultaneously, Q. J. Roy. Meteor. Soc., 144, 529–538, 2018.

Mu, L., Nerger, L., Tang, Q., Losa, S. N., Sidorenko, D., Wang, Q., Semmler, T., Zampieri, L., Losch, M., and Goessling, H. F.: Toward a data assimilation system for seamless sea ice prediction based on the AWI climate model, J. Adv. Model. Earth Sy., 12, 359, https://doi.org/10.1029/2019MS001937 , 2020.

Nerger, L. and Hiller, W.: Software for Ensemble-based Data Assimilation Systems – Implementation Strategies and Scalability, Comput. Geosci., 55, 110–118, 2013.

Nerger, L., Hiller, W., and Schröter, J.: PDAF - The Parallel Data Assimilation Framework: Experiences with Kalman filtering., in: Use of High Performance Computing in Meteorology – Proceedings of the 11. ECMWF Workshop, edited by: Zwieflhofer, W. and Mozdzynski, G., World Scientific, 63–83, 2005.

Nerger, L., Danilov, S., Hiller, W., and Schröter, J.: Using sea level data to constrain a finite-element primitive-equation ocean model with a local SEIK filter, Ocean Dynam., 56, 634–649, 2006.

Nerger, L., Janjić, T., Schröter, J., and Hiller, W.: A regulated localization scheme for ensemble-based Kalman filters, Q. J. Roy. Meteor. Soc., 138, 802–812, 2012a.

Nerger, L., Janjić, T., Schröter, J., and Hiller, W.: A unification of ensemble square root Kalman filters, Mon. Weather Rev., 140, 2335–2345, 2012b.

Nerger, L., Schulte, S., and Bunse-Gerstner, A.: On the influence of model nonlinearity and localization on ensemble Kalman smoothing, Q. J. Roy. Meteor. Soc., 140, 2249–2259, 2014.

Nerger, L., Tang, Q., and Mu, L.: The PDAF model binding for AWI-CM (AWI-CM-PDAF version 1.0), Zenodo, https://doi.org/10.5281/zenodo.3822030, 2019a.

Nerger, L., Tang, Q., and Mu, L.: Efficient ensemble data assimilation for coupled models with the Parallel Data Assimilation Framework: Example of AWI-CM – output files and plot scripts, Zenodo, https://doi.org/10.5281/zenodo.3823816, 2019b.

OpenMP: OpenMP Application Program Interface Version 3.0, available at: http://www.openmp.org/ (last access: 14 September 2020), 2008.

Pardini, F., Corradini, S., Costa, A., Ongari, T. E., Merucci, L., Neri, A., Stelitano, D., and deichieli Vitturi, M.: Ensemble-Based Data Assimilation of Volcanic Ash Clouds from Satellite Observations: Application to the 24 December 2018 Mt. Etna Explosive Eruption, Atmosphere, 11, 359, https://doi.org/10.3390/atmos11040359, 2020.

Penny, S. G., Akella, S., Alves, O., Bishop, C., Buehner, M., Chevalier, M., Counillon, F., Drper, C., Frolov, S., Fujii, Y., Kumar, A., Laloyaux, P., Mahfouf, J.-F., MArtin, M., Pena, M., de Rosnay, P., Subramanian, A., Tardif, R., Wang, Y., and Wu, X.: Coupled data assimilation for integrated Earth system analysis and prediction: Goals, Challenges and Recommendations, Tech. Rep. WWRP 2017-3, World Meteorological Organization, 2017.

Pham, D. T., Verron, J., and Roubaud, M. C.: A singular evolutive extended Kalman filter for data assimilation in oceanography, J. Marine Syst., 16, 323–340, 1998.

Pradhan, H. K., Voelker, C., Losa, S. N., Bracher, A., and Nerger, L.: Assimilation of global total chlorophyll OC-CCI data and its impact on individual phytoplankton fields, J. Geophys. Res.-Oceans, 124, 470–490, 2019.

Rackow, T., Sein, D. V., Semmler, T., Danilov, S., Koldunov, N. V., Sidorenko, D., Wang, Q., and Jung, T.: Sensitivity of deep ocean biases to horizontal resolution in prototype CMIP6 simulations with AWI-CM1.0, Geosci. Model Dev., 12, 2635–2656, https://doi.org/10.5194/gmd-12-2635-2019, 2019.

Sakov, P. and Oke, P. R.: A deterministic formulation of hte ensemlbe Kalman filter: an alternative to ensemble square root filters, Tellus, 60A, 361–371, 2008.

Sidorenko, D., Rackow, T., Jung, T., Semmler, T., Barbi, D., Danilov, S., Dethloff, K., Dorn, W., Firg, K., Goessling, H. F., d. Handorf, Harig, S., Hiller, W., Juricke, S., Losch, M., Schröter, J., Sein, D. V., and Wang, Q.: Towards multi-resolution global climate moeling with ECHAM6-FESOM. Part I: model formulation and mean climate, Clim. Dynam., 44, 757–780, 2015.

Sluka, T. C., Penny, S. G., Kalnay, E., and Miyoshi, T.: Assimilating atmospheric observations into the ocean using strongly coupled ensemble data assimilation, Geophys. Res. Lett., 43, 752–759, 2016.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J.: Obstacles to high-dimensional particle filtering, Mon. Weather Rev., 136, 4629–4640, 2008.

Stevens, B., Giorgetta, M., Esch, M., Mauritsen, T., Crueger, T., Rast, S., Salzmann, M., Schmift, H., an K. Blovk, J. B., Brokopf, R., Fast, I., Kinne, S., Koernblueh, L., Lohmann, U., Pincus, R., Reichler, T., and Roeckner, E.: Atmospheric component of the MPI-M Earth system model: ECHAM6., J. Adv. Model. Earth Sy., 5, 146–172, 2013.

Tang, Q., Mu, L., Sidorenko, D., Goessling, H., Semmler, T., and Nerger, L.: Improving the ocean and atmosphere in a coupled ocean-atmosphere model by assimilating satellite sea surface temperature and subsurface profile data, Q. J. Roy. Meteor. Soc., https://doi.org/10.1002/qj.3885, in press, 2020.

Tödter, J. and Ahrens, B.: A second-order exact ensemble square root filter for nonlinear data assimilation, Mon. Weather Rev., 143, 1347–1467, 2015.

Valcke, S.: The OASIS3 coupler: a European climate modelling community software, Geosci. Model Dev., 6, 373–388, https://doi.org/10.5194/gmd-6-373-2013, 2013.

van Leeuwen, P. J.: Particle Filtering in Geophysical Systems, Mon. Weather Rev., 137, 4089–4114, 2009.

van Leeuwen, P. J.: Nonlinear data assimilation in geosciences: An extremely efficient particle filter, Q. J. Roy. Meteor. Soc., 136, 1991–1999, 2010.

van Leeuwen, P. J., Künsch, H. R., Nerger, L., Potthast, R., and Reich, S.: Particle filters for high-dimensional geoscience applications: a review, Q. J. Roy. Meteor. Soc., 145, 2335–2365, 2019.

Vetra-Carvalho, S., van Leeuwen, P. J., Nerger, L., Barth, A., Altaf, M. U., Brasseur, P., Kirchgessner, P., and Beckers, J.-M.: State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems, Tellus A, 70, 1445364, https://doi.org/10.1080/16000870.2018.1445364, 2018.

Wang, Q., Danilov, S., and Schröter, J.: Finite element ocean circulation model based on triangular prismatic elements with application in studying the effect of topography representation, J. Geophys. Res., 113, C05015, https://doi.org/10.1029/2007JC004482, 2008.

Yu, L., Fennel, K., Bertino, L., Gharamti, M. E., and Thompson, K. R.: Insights on multivariate updates of physical and biogeochemical ocean variables using an ensemble Kalman filter and an idealized model of upwelling, Ocean Model., 126, 13–28, 2018.

Zhang, S., Harrison, M. J., Rosati, A., and Wittenberg, A.: System design and evaluation of a coupled ensemble data assimilation for global oceanic climate studies, Mon. Weather Rev., 135, 3541–3564, 2007.