Developing a common, flexible and efficient framework for weakly coupled ensemble data assimilation based on C-Coupler2.0

Chao Sun[1], Li Liu[1,3], Ruizhe Li[1], Xinzhu Yu[1], Hao Yu[1], Biao Zhao[1,2,4], Guansuo Wang[2,4], Juanjuan Liu[5], Fangli Qiao[2,4], Bin Wang[1,5]

[1] Ministry of Education Key Laboratory for Earth System Modeling, Department of Earth System Science, Tsinghua University, Beijing, China
[2] First Institute of Oceanography, Ministry of Natural Resources, Qingdao, China
[3] Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), China
[4] Key Lab of Marine Science and Numerical Modeling, Ministry of Natural Resources, Qingdao, China
[5] State Key Laboratory of Numerical Modeling for Atmospheric Sciences and Geophysical Fluid Dynamics (LASG), Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, China

Correspondence to: Li Liu (liuli-cess@tsinghua.edu.cn), Bin Wang (wab@tsinghua.edu.cn)

Abstract. Data assimilation (DA) provides initial states of model runs by combining observational information and models. Ensemble-based DA methods that depend on the ensemble run of a model have been widely used. In response to the development of seamless prediction based on coupled models or even Earth system models, coupled DA is now in the mainstream of DA development. In this paper, we focus on the technical challenges in developing a coupled ensemble DA system, especially how to conveniently achieve efficient interaction between the ensemble of the coupled model and the DA methods. We first propose a new DA framework DAFCC1 (**D**ata **A**ssimilation **F**ramework based on **C**-**C**oupler2.0, version 1) for weakly coupled ensemble DA, which enables users to conveniently integrate a DA method into a model as a procedure that can be directly called by the model ensemble. DAFCC1 automatically and efficiently handles data exchanges between the model ensemble members and the DA method without global communications, and does not require users to develop extra codes for implementing the data exchange functionality. Based on DAFCC1, we then develop an example weakly coupled ensemble DA system by combining an ensemble DA system and a regional atmosphere-ocean-wave coupled model. This example DA system and our evaluations demonstrate the correctness of DAFCC1 in developing a weakly coupled ensemble DA system and the effectiveness in accelerating an offline DA system that uses disk files as the interfaces for the data exchange functionality.

1 Introduction

Data assimilation (DA) methods, which provide initial states of model runs by combining observational information and models, have been widely used in weather forecasting and climate prediction. The **en**semble **K**alman **f**ilter (EnKF; Houtekamer and Mitchell, 1998; Evensen, 2003; Lorenc, 2003a; Anderson and Collins, 2007; Whitaker, 2012) is a widely used DA method that depends on an ensemble run of members. Other DA methods such as the Nudging method (Hoke and Anthes, 1976; Vidard

et al., 2003), Optimal Interpolation (OI; Gandin, 1965), Ensemble OI (EnOI; Oke et al., 2002; Evensen, 2003), three-dimensional variational analysis (3D-Var; Anderson et al., 1998; Courtier et al.1998; Gauthier et al., 1999; Lorenc, 2000) and four-dimensional variational analysis (4D-Var; Courtier et al., 1994; Kalnay, 2003; Lorenc, 2003b; Rabier et al., 2007) can be

35   technically viewed as a special case of ensemble-based methods with only one member in the ensemble when we attempt to design and develop a software framework for data assimilation. Moreover, hybrid DA methods, such as hybrid Ensemble/3D-Var (Hamill, 2000; Etherton and Bishop, 2004; Wang et al., 2008, 2013; Ma et al., 2014) and ensemble-based 4D-Var schemes (Fisher, 2003; Bishop and Hodyss, 2011; Bonavita et al., 2012, 2016; Buehner et al., 2015), also depend on the ensemble run of members from the same model.

40   With the rapid development of science and technology, numerical forecasting systems are evolving from only an individual component model (such as an atmospheric model) to coupled models that can achieve better predictability (Brown et al., 2012; Mulholland et al., 2015), and Earth system models are being used to develop seamless prediction that spans timescales from minutes to months or even decades (Palmer et al., 2008; Hoskins, 2013). Along with the use of coupled models in numerical forecasting, common and flexible DA methods for coupled models are urgently needed (Brunet et al., 2015; Penny et al., 2017).

45   Coupled DA technologies have already been investigated widely and DA systems have been constructed (Sugiura et al., 2008; Fujii et al., 2009, 2011; Saha et al., 2010, 2014; Sakov et al., 2012; Yang et al., 2013; Tardif et al., 2014, 2015; Lea et al., 2015; Lu et al., 2015a, b; Mochizuki et al., 2016; Laloyaux et al., 2016, 2018; Browne et al., 2019; Goodliff et al., 2019; Skachko et al. 2019), in which ensemble based DA methods have already been applied (e.g., Zhang et al., 2005, 2007; Sluka et al., 2016).

     To develop a coupled ensemble DA system, besides the scientific challenges regarding DA methods, there are also technical
50   challenges to be addressed, such as how to achieve an ensemble run of a coupled model, how to conveniently integrate the software of a coupled model and the software of ensemble DA methods into a robust system, and how to conveniently achieve efficient interaction between the ensemble of the coupled model and the DA methods. The existing ensemble DA frameworks supporting coupled DA such as the Data Assimilation Research Testbed (DART; Anderson et al., 2009) and the Grid point Statistical Interpolation (GSI; Shao et al., 2016) combined with EnKF (Liu et al., 2018a), employ disk files as the interfaces
55   of data exchange between the model ensemble members and the DA methods, and iteratively switch between the run of the model ensemble and DA using software-based restart functionality that also relies on disk files. Such an implementation (called offline implementation hereafter) can guarantee software independence between the models and the DA methods, so as to achieve flexibility and convenience in software integration; however, the extra I/O accesses of disk files as well as the extra initialization of software modules introduced by the data exchange and the restarts are time-consuming and can be a severe
60   performance bottleneck under finer model resolution (Heinzeller et al., 2016; Craig et al., 2017). The Parallel Data Assimilation Framework (PDAF; Nerger et al., 2005; Nerger and Hiller, 2013; Nerger et al., 2020) and the Employing Message Passing Interface for Researching Ensembles (EMPIRE; Browne and Wilson, 2015) framework have shown that MPI (Message Passing Interfaces)-based data exchanges between the model ensemble members and DA procedures can produce better performance for DA systems, because they do not require disk files or the restart operations.

65   Noting that most existing couplers for Earth system modeling have already achieved flexible MPI-based data exchanges between component models in a coupled system, we design and develop a common, flexible and efficient framework for coupled ensemble data assimilation, based on the latest version of the Community Coupler (C-Coupler2.0; Liu et al., 2018b). Considering that existing observation processing systems can introduce different observation frequencies corresponding to different component models, we take consideration of weakly coupled ensemble DA where the data from different component

70   models are assimilated independently by separate DA methods (Zhang et al., 2005, 2007; Fujii et al., 2009, 2011; Saha et al., 2010, 2014) in this work, and in further work will then target strongly coupled ensemble DA, which generally uses a cross-domain error covariance matrix to account for the impact of the same observational information on different component models cooperatively (Tardif et al., 2014, 2015; Lu et al., 2015a, b; Sluka et al., 2016).

    The remainder of this paper is organized as follows. Section 2 introduces the overall design of the new DA framework

75   named DAFCC1 (**D**ata **A**ssimilation **F**ramework based on **C**-**C**oupler2.0, version 1). The implementation of DAFCC1 is described in Section 3. Section 4 introduces the development of an example weakly coupled ensemble DA system based on DAFCC1. Section 5 evaluates DAFCC1. Finally, Section 6 contains a discussion and conclusions.

2 Overall design of the new framework

    The experiences gained from PDAF and EMPIRE show that, a framework with an online implementation that handles the data

80   exchanges via MPI functionalities is essential for improving the interaction between the model and the DA software. There can be different strategies for the online implementation. In EMPIRE, a DA method is compiled into a standalone executable running on the processes distinct from the model ensemble, and global communications with *MPI_gatherv* and *MPI_scatterv* are used for exchanging data between the model ensemble and the DA method. Such an implementation can maintain the independence between the DA software and the model, but is inefficient because of inefficient global communications and

85   idle processes due to sequential running of the model and DA modules in the sequential DA systems. In PDAF, a DA method is transformed into a native procedure that is called by the corresponding models via the PDAF application programing interfaces (APIs). Thus, a model and a DA method can be compiled into the same executable, and the DA method can share the processes of the model ensemble. The code releases of PDAF (http://pdaf.awi.de/trac/wiki) provide template implementations of data exchanges for a default case where a DA method is running on the processes of the first ensemble

90   member (for example, given that there are 10 ensemble members and each member uses 100 processes, the DA method is running on 100 processes) and uses the same parallel decomposition (grid domain decomposition for parallelization) with the corresponding model. When users want a case different from the default (e.g., a DA method does not use the same processes with the first ensemble member or uses a parallel decomposition different from the model), users should develop new code implementations for the corresponding data exchange functionality following the rules of PDAF.

95   Most DA software consists of parallel programs that generally can be accelerated by using more processor cores. When running an ensemble DA algorithm for a component model in an ensemble run, all ensemble members of the component model

are synchronously waiting for the results of the DA algorithm. Therefore, all the processor cores corresponding to all ensemble members of the component model can be used to accelerate the DA algorithm. To develop a new framework for weakly coupled ensemble data assimilation, we should improve the implementation of the data exchange functionality in at least three aspects: 1) the new implementation does not use global communications with *MPI_gatherv* and *MPI_scatterv*; 2) the new implementation enables a DA method to run on the processes of all ensemble members (for example, given that there are 10 ensemble members and each member uses 100 processes, the DA method can run on 1000 processes); 3) the new implementation does not require users to develop extra codes no matter whether a DA method and the corresponding model use the same of different parallel decompositions. A DA method requires to exchange data with each model ensemble member. When a DA algorithm uses a parallel decomposition that differs from the model, the data exchange between the DA algorithm and an ensemble member will introduce a challenge of transferring fields between different process sets with different parallel decompositions.

Fortunately, such a challenge has already been overcome by most existing couplers (Craig et al., 2012; Valcke, 2012; Liu et al., 2014; Craig et al., 2017; Liu et al., 2018b). Each of these couplers can transfer data between different process sets with different parallel decompositions without the global communications. We therefore use the C-Coupler2.0 (Liu et al., 2018b), the latest version of the Community Coupler (C-Coupler), as the foundation for developing DAFCC1. Moreover, C-Coupler2.0 has more functionalities that DAFCC1 can benefit from. For example, C-Coupler2.0 can handle data exchange of 3-D or even 4-D fields where the source and destination fields can have different dimension orders (e.g., vertical+horizontal at the source field, and horizontal+vertical at the destination field). It will be convenient to combine ensemble members of a coupled model into a single MPI program based on C-Coupler2.0 because each ensemble member can be registered as a component model of C-Coupler2.0. As shown in Fig. 1a, most operations for achieving data exchanges can be generated automatically because C-Coupler2.0 can generate coupling procedures between two process sets even when the two sets are overlapping.

A significant challenge here is that C-Coupler2.0 can only handle coupling exchanges between two component models or within one component model, but cannot handle coupling exchanges between a DA algorithm and each model ensemble member. To address this challenge, DAFCC1 automatically generates a special C-Coupler2.0 component model (called ensemble-set component model hereafter) that covers all ensemble members for running a DA algorithm. Thus, coupling exchanges between a DA algorithm and each model ensemble member can be transformed into the coupling exchanges between the ensemble-set component and each ensemble member. Specifically, DAFCC1 introduces three new steps, i.e., initialization, running and finalization of DA instances (instances of DA algorithms), to the model flowchart with C-Coupler2.0 (Fig. 1b). These steps enable a DA instance to run on the processes of all ensemble members, and achieve automatic coupling exchanges between a DA algorithm and each model ensemble member. The software architecture of DAFCC1 based on C-Coupler2.0 is shown in Fig. 2. It includes a set of new managers (i.e., DA algorithm integration manager, ensemble component manager, online DA procedure manager, and ensemble DA configuration manager) and the new APIs corresponding to these managers. The DA algorithm integration manager enables the user to conveniently develop driving interfaces for a DA algorithm based on a set of new APIs that enables the DA algorithm to declare its input and output fields and to obtain various

information from the model. When a DA algorithm includes multiple independent modules (such as observation operators and analysis modules), each module can be called separately by the model. The dynamic link library (DLL) technique is used to connect a DA algorithm program to a model program. The DA algorithm program is compiled into a DLL that is dynamically linked to a model when an instance of the DA algorithm is initialized. Using the DLL technique allows us to couple a DA algorithm and a model without modifying and recompiling the model codes, and it provides greater independence and convenience because the original configuration and compilation systems of the DA algorithm can generally be preserved. The ensemble component manager governs the communicators of ensemble members. The online DA procedure manager provides several APIs that enable the ensemble members of a component model to initialize, run and finalize a DA algorithm instance cooperatively. The data exchanges between the ensemble members and the DA algorithm are also handled automatically in this manager. The ensemble DA configuration manager enables the user to flexibly choose DA algorithms and set parameters for a DA simulation via a configuration file.

With the software architecture in Fig.2 and the detailed implementations in Section 3, DAFCC1 enables a coupled ensemble DA system to achieve the following features:

1) Each component model can use different instances of DA algorithms online independently, and the execution of a DA algorithm in the MPI processes of a component model does not force other MPI processes to be idled. For example, components 1, 2, and 4 in Fig. 3 perform DA with different time periods (e.g., component 4 performs DA more frequently than component 1 and 2), while component 3 does not use DA algorithms.

2) Given a common DA algorithm, it can be used by different component models under different instances with different configurations (e.g., the fields assimilated, the observational information used, and the frequency). In Fig. 3 for example, components 2 and 4 use different instances of the same DA algorithm 2 independently.

3) An instance of a DA algorithm can either use the processes of all ensemble members of the same component model cooperatively or use the processes of each ensemble member separately. For example, each DA algorithm instance in Fig. 3 uses the processes of all ensemble members of the corresponding component model cooperatively, except procedure 1 of DA algorithm 1 that uses the processes of each ensemble member of component 1 separately.

4) Besides employing the DLL technique for integrating DA algorithm programs, a configuration file is designed for increasing the flexibility and convenience in using a DA algorithm (see Section 3.4 for detailed implementation).

3 Implementation of DAFCC1

In this section, we will detail the implementation of DAFCC1 in terms of the ensemble component manager, DA algorithm integration manager, online DA procedure manager, and ensemble DA configuration manager. Moreover, we will provide an example of how to use DAFCC1 to develop a DA system.

3.1 Implementation of the ensemble component manager

To achieve coupling exchanges between a DA algorithm and each ensemble member, each ensemble member should be used as a separate component model registered to C-Coupler2.0 via the API *CCPL_register_component* (Please refer to Liu et al. 2018b for more details). In C-Coupler2.0, model names are used as the keywords to distinguish different component models.

165 To distinguish different ensemble members of a model that generally share the same code or executable, we update the API *CCPL_register_component* to implicitly generate different names of ensemble members by appending the ID of each ensemble member to the model name (the parameter list of the API *CCPL_register_component* is unchanged). The ID of an ensemble member is given as the last argument (formatted as "CCPL_ensemble_*{ensemble numbers}_{member ID}*") of the corresponding executable when submitting an MPI run (see Fig. 4 as an example), where "*ensemble numbers*" marks the

170 number of ensemble members and "*member ID*" marks the ensemble member ID of the current component.

C-Coupler2.0 can manage hierarchical relationship among models, where a model can have a set of children models. Given an ensemble run of a coupled model, although all ensemble members of all component models of the coupled model can be organized into a single level (see Fig. 5a), we recommend constructing two hierarchical levels (see Fig. 5b), where each ensemble member of the coupled model is at the first level while the component models of each ensemble member are at the

175 second level. This hierarchical organization retains the original architecture of the coupled model, and only requires users to simply register the coupled model to C-Coupler2.0.

In order to enable a DA algorithm to run on the MPI processes of all ensemble members of a component model (Fig. 3), an ensemble-set component model that covers all ensemble members of the component model is required for calling the DA algorithm (for example, the green box in Fig. 5b corresponds to an ensemble-set component model). As the ensemble-set

180 component model does not exist in the original hierarchical levels in Fig. 5b, it should be generated with extra efforts. The ensemble component manager provides the capability of automatically generating the corresponding ensemble-set component model when initializing a DA instance. The MPI communicator of the ensemble-set component model is generated through unifying the communicators of the ensemble members of the corresponding component model that runs the DA instance.

3.2 Implementation of the DA algorithm integration manager

185 When a DA algorithm runs on the processes of a component model, the model and the DA algorithm can be viewed as a caller and a callee respectively in a program. A callee generally declares a list of arguments that includes a set of input and output variables, and a caller should match the argument list of the callee when calling the callee (a model that calls a DA algorithm is called the host model of the DA algorithm hereafter). When a caller and a callee are statically linked together, a compiler generally can guarantee the consistency of the argument list between them. However, it is a challenge that compilers cannot

190 guarantee such consistency between a host model and a DA algorithm that is enclosed in a DLL and dynamically linked to the host model.

To address the above challenge, we designed and developed a new solution in DAFCC1 for passing arguments between a host model and a DA algorithm. There are three driving subroutines for initializing, running, and finalizing a DA algorithm. These subroutines are enclosed in the same DLL with the DA algorithm. Specifically, names of these subroutines share the name of the DA algorithm as the prefix and are distinguished by different suffixes. We tried to make the explicit argument list of each driving subroutine as simple as possible (e.g., the explicit argument list only includes a few integer arrays), and developed a set of APIs for flexibly passing implicit arguments. Based on these APIs, the DA algorithm can obtain the required information of the host model, and can also declare a set of implicit input or output arguments of fields. Figure 6 shows an example of the driving subroutines where the running and finalization driving subroutines are quite simple. The initialization driving subroutine includes the original functionalities of the DA algorithm such as determining parallel decompositions, allocating memory space for variables and other operations for initialization. Moreover, it includes additional operations for obtaining the information of the host model via C-Coupler2.0, registering the parallel decompositions, grids, and field instances to C-Coupler2.0, and declaring a set of field instances as implicit input or output arguments. Data exchanges between the host model and the DA algorithm are conducted automatically and implicitly by DAFCC1 (Section 3.3.2), so the running driving subroutine *DA_CCPL_RUN* does not include explicit calls for data exchanges.

The use of DAFCC1 requires some further changes to the codes of a DA algorithm. For example, the original communicator of the DA algorithm needs to be replaced with the communicator of the host model that can be obtained via the corresponding C-Coupler API, and the original I/O accesses for the model data in the DA algorithm can be turned off.

3.3 Implementation of the online DA procedure manager

To make the same DA algorithm used by different component models, DAFCC1 enables a component model to use a separate instance of the same DA algorithm with the corresponding configuration information. Corresponding to the three driving subroutines of a DA algorithm, there are three APIs (*CCPL_ensemble_procedures_inst_init*, *CCPL_ensemble_procedures_inst_run*, and *CCPL_ensemble_procedures_inst_finalize*) that are directly called by the code of a host model. These APIs initialize, run and finalize a DA algorithm instance, and handle the data exchanges between the host model and the DA algorithm instance automatically. In a general case in Fig. 1b, the API *CCPL_ensemble_procedures_inst_init* is called when initializing the ensemble DA system before starting the time loop, the API *CCPL_ensemble_procedures_inst_finalize* is called after finishing the time loop, and the API *CCPL_ensemble_procedures_inst_run* is called in the time loop, which enables different assimilation windows to share the same DA instance without restarting the model and the DA algorithm. When a component model initializes, runs, or finalizes a DA algorithm instance, all ensemble members of this component model should call the corresponding API at the same time.

3.3.1 API for initializing a DA algorithm instance

The API *CCPL_ensemble_procedures_inst_init* includes the following steps for initializing a DA algorithm instance:

1) Determining the host model of the DA algorithm instance according to the corresponding information in the configuration file. If the DA algorithm instance is an individual algorithm that operates on the data of each ensemble member separately (e.g., Procedure 1 of DA algorithm 1 in Fig. 3), each ensemble member will be a host model. Otherwise (i.e., the DA algorithm instance is an ensemble algorithm that operates on the data of the ensemble set; e.g., Procedure 2 of DA algorithm 1 in Fig. 3), the host model will be the ensemble-set component model that will be generated automatically by the ensemble component manager.

2) Preparing information from the host model, such as model grids, parallel decompositions, and field instances, which can be obtained by the initialization driving subroutine of the DA algorithm via the corresponding APIs.

3) Initializing the corresponding DA algorithm instance according to the corresponding algorithm name and DLL name specified in the corresponding configuration file (Section 3.4), where the corresponding DLL will be linked to the host model and the corresponding initialization driving subroutine in the DLL will be called. This implementation enables the user to conveniently change the DA algorithms used in different simulations via the configuration file without modifying the code of the model.

4) Setting up data exchange operations according to the input or output fields of the DA algorithm instance declared in the initialization driving subroutine via the corresponding APIs. If the DA algorithm instance is specified as an individual algorithm via the ensemble DA configuration (Section 3.4), the data exchange is within each ensemble member. Otherwise, the ensemble-set component model is involved in the data exchange. The data exchange is divided into two levels: the data exchange between the ensemble members and DAFCC1, and the data exchange between DAFCC1 and the DA algorithm. The data exchange between DAFCC1 and the DA algorithm instance is simply achieved by the import/export interfaces of C-Coupler2.0, which flexibly rearrange the fields in the same component model between different parallel decompositions. If the DA algorithm instance is an ensemble algorithm, the data exchange between the ensemble members and DAFCC1 is also handled by the import/export interfaces of C-Coupler2.0, which flexibly transfer the same fields between different component models (each ensemble member and the ensemble set are different component models). Otherwise, the data exchange between the ensemble members and DAFCC1 is simplified to a data copy. DAFCC1 will hold a separate memory space for each model field relevant to the DA algorithm, which enables a DA algorithm instance to use instantaneous model results or statistical results (i.e., mean, maximum, cumulative, and minimum) in a time window, and enables an ensemble DA algorithm instance to use aggregated results or statistical results (ensemble-mean, ensemble-anomaly, ensemble-maximum, or ensemble-minimum) from ensemble members.

Consistent with the functionalities in the above steps, the API *CCPL_ensemble_procedures_inst_init* includes the following arguments:

- The *ID* of the current ensemble member that calls this API, and the common full name of the ensemble members. When registering a component model to C-Coupler2.0, its *ID* is allocated and its unique full name formatted as "*parent_full_name@model_name*" is generated, where "*model_name*" is the name of the component model, and "*parent_full_name*" is the full name of the parent component model (if any). Given that the name of the component model

1 in Fig. 5 is "*comp1*", in the one-level model hierarchy in Fig. 5a, the full names of ensemble members of the component model 1 are "*comp1_1*" to "*comp1_N*", and the common full name of the ensemble members is "*comp1_**" where "***" is a wildcard; while in the two-level model hierarchy in Fig. 5b, the full names of ensemble members of the component model 1 are "*coupled_1@comp1*" to "*coupled_N@comp1*" and the common full name is "*coupled_*@comp1*" (given that the name of the coupled model is "*coupled*"). Such a common full name can be used for generating the ensemble-set component model when the DA algorithm instance is an ensemble algorithm.

- The name of the DA algorithm instance, which is the keyword of the DA algorithm instance and also specifies the corresponding configuration information. Different DA algorithm instances can correspond to different DA algorithms or the same DA algorithm. For example, the component models 2 and 4 use different instances of the same DA algorithm in Fig. 3.

- An optional list of model grids and parallel decompositions, which enable the DA algorithm instance to obtain the grid data and use the same parallel decompositions as the host model.

- A list of field instances that can be used for DA. This list should cover all input or output fields of the DA algorithm.

- An optional integer array of control variables that can be obtained by the DA algorithm instance via the corresponding APIs.

- An annotation, which is a string giving a hint for locating the model code of the API call corresponding to an error or warning, is recommended but not mandatory.

3.3.2 API for running a DA algorithm instance

The API *CCPL_ensemble_procedures_inst_run* includes the following steps for running a DA algorithm instance:

1) Executing the data exchange operations for the input fields of the DA algorithm instance. This step automatically transfers the input fields from each ensemble member of the corresponding component model to DAFCC1 and then from DAFCC1 to the DA algorithm instance. The statistical processing regarding the time window or the ensemble is done at the same time.

2) Executing the DA algorithm instance through calling the running driving subroutine of the DA algorithm.

3) Executing the data exchange operations for the output fields of the DA algorithm instance. This step automatically transfers the output fields from the DA algorithm instance to DAFCC1 and then from DAFCC1 to each ensemble member of the corresponding component model.

Each DA algorithm instance has a timer specified via the configuration information, which determines when the DA algorithm instance is run. The API *CCPL_ensemble_procedures_inst_run* for a DA algorithm instance can be called at each time step, while the above three steps will be executed only when the specified timer is on. To store the input data such as the observational information, a DA algorithm instance can either share the working directory of its host model or use its own working directory specified via the configuration information. The API *CCPL_ensemble_procedures_inst_run* will change and then recover the current directory for calling the running driving subroutine of the DA algorithm, if necessary.

3.3.3 API for finalizing a DA algorithm instance

The API *CCPL_ensemble_procedures_inst_finalize* is responsible for finalizing a DA algorithm instance through calling the finalization driving subroutine of the DA algorithm.

3.4 Implementation of the ensemble DA configuration manager

The configuration information of all DA algorithm instances used in a coupled DA simulation is enclosed in an XML configuration file (e.g., Fig. 7). Each DA algorithm instance has a distinct XML node (e.g., the XML node "da_instance" in Fig. 7, where the attribute "name" is the name of the DA algorithm instance associated with the API "*CCPL_ensemble_procedures_inst_init*"), which enables the user to specify the following configurations:

1) The DA algorithm specified in the XML node "external_procedures" in Fig. 7. The attribute "dll_name" in the XML node specifies the dynamic link library, and the attribute "procedures_name" specifies the DA algorithm's name associated with the corresponding driving subroutines. When the user seeks to change the DA algorithm used by a component model, it is only necessary to modify the XML node "external_procedures" in most cases.

2) The periodic timer specified in the XML node "periodic_timer" in Fig. 7, which enables the user to flexibly set the periodic model time of running the corresponding DA algorithm. Besides the attribute "period_unit" and "period_count" for specifying the period of the timer, the user can specify a lag via the attribute "local_lag_count". For example, given a periodic timer <"period_unit"="hours", "period_count"=6, "local_lag_count"=3>, its period is 6 hours, and it will not be on at the 0th, 6th, and 12th hours, but instead on at the 3rd, 9th, and 15th hours due to the "local_lag_count" of 3.

3) Statistical processing of input fields specified in the XML node "field_instances" in Fig. 7. The attribute "time_processing" specifies the statistical processing in each time window determined by the periodic timer. The attribute "ensemble_operation" specifies the statistical processing among ensemble members. For an individual DA algorithm, the attribute "ensemble_operation" should be set to "*none*". All fields can share the default specification of statistical processing, while a field can have its own statistical processing specified in a sub node of the XML node "field_instances".

4) The working directory and the scripts for pre- and post-assimilation analysis (e.g., for processing the data files of observational information) optionally specified in the XML node "processing_control" in Fig. 7. When the working directory is not specified, the DA algorithm instance will use the working directory of its host model. The script specified in the sub XML node "pre_instance_script" will be called by the root process of the host model before the API *CCPL_ensemble_procedures_inst_run* calls the DA algorithm, and the script specified in the sub XML node "post_instance_script" will be called by the root process of the host model after the DA algorithm run finishes.

4 An example weakly coupled ensemble DA system based on DAFCC1

To provide further information on how to use DAFCC1 and for validating and evaluating DAFCC1, we developed an example weakly coupled ensemble DA system by combining the ensemble DA system GSI/EnKF (Shao et al., 2016; Liu et al., 2018b)

and a regional **F**irst **I**nstitute of **O**ceanography **A**tmosphere-**O**cean-**W**ave (FIO-AOW) coupled model (Zhao et al., 2017; Wang et al., 2018). GSI/EnKF mainly focuses on regional numerical weather prediction (NWP) applications coupled with the Weather Research and Forecasting (WRF) model (Wang et al., 2014), while FIO-AOW consists of WRF, the Princeton Ocean Model (POM; Blumberg and Mellor 1987; Wang et al., 2010), the MArine Science and NUmerical Modeling wave model (MASNUM; Yang et al., 2005; Qiao et al, 2016), and all the above three model components are coupled together by using C-Coupler (Liu et al., 2014, 2018b). FIO-AOW has already been used in the research for exploring the sensitivity of typhoon simulation to physical processes and improving typhoon forecasting (Zhao et al, 2017; Wang et al., 2018). There are two main steps in developing the example system.

1) We developed an ensemble DA sub-system of WRF by adapting GSI/EnKF to DAFCC1. This sub-system helps validate DAFCC1 and evaluate the improvement in performance obtained by DAFCC1 (Section 5).

2) We merged the above sub-system and FIO-AOW to produce the example DA system that only computes atmospheric analyses corresponding to WRF currently. This system demonstrates the correctness of DAFCC1 in developing a weakly coupled ensemble DA system.

4.1 An ensemble DA sub-system of WRF

4.1.1 Brief introduction to GSI/EnKF

GSI/EnKF combines a variational DA sub-system (GSI; Shao et al., 2016) and an ensemble DA sub-system (EnKF; Liu et al., 2018a), which can be used as a variational, a pure ensemble or a hybrid DA system sharing the same observation operator in the GSI codes. It provides two options for calculating analysis increments for ensemble DA; i.e., a serial Ensemble Square Root Filter (EnSRF) algorithm (Whitaker et al., 2012) and a Local Ensemble Kalman Filter (LETKF) algorithm (Hunt et al., 2007). In this paper, we use the pure ensemble DA system without using variational DA, where GSI is used as the observation operator that calculates the difference between model variables and observations on the observation space and EnSRF is chosen for calculating atmospheric analyses and updating atmosphere model variables.

Figure 8a shows the flowchart for running the pure ensemble DA system of the WRF model in a DA window. It consists of the following main steps that are driven by scripts, while the data exchanges between these main steps are achieved via data files.

1) Ensemble model forecast. An ensemble run of WRF is initiated or restarted from a set of input data files, and then is stopped after producing a set of output files (called model background files hereafter) for DA and for restarting the ensemble run in the next DA window.

2) Calculating the ensemble mean of model DA variables. A separate executable is initiated for calculating the ensemble mean of each DA variable based on the model background files, and then outputs the ensemble mean to a new background file.

3) Observation operator for the ensemble mean. GSI is initiated as the observation operator for the ensemble mean. It takes the ensemble mean file, files of various observational data (e.g., conventional data, satellite radiance observations, GPS radio occultations, and radar data) and multiple fixed files (e.g., statistic files, configuration files, bias correction files, and CRTM coefficient files) as input, and produces an observation prior (observation innovation) file for the ensemble mean and files containing observational intermediate information (e.g., bias correction and thinning).

4) Observation operator for each ensemble member. GSI is initiated as the observation operator for each ensemble member. It takes the background file of the corresponding ensemble member, the fixed files and the observational intermediate information files as input, and produces an observation prior file for the corresponding ensemble member.

5) EnKF for calculating analysis increments. EnKF is initiated for calculating analysis increments of the whole ensemble. It takes the model background files, the observation prior files and the fixed files as input, and finally updates model background files with the analysis increments. The updated model background files are used for restarting the ensemble model forecast in the next DA window.

4.1.2 Adapting GSI/EnKF to DAFCC1

When adapting GSI/EnKF to DAFCC1, an ensemble-set component model derived from the ensemble forecast of WRF (corresponding to the first main step in Section 4.1.1) is generated as the host model that drives the DA algorithm instances corresponding to the remaining main steps. As shown in Fig. 9, three DA instances corresponding to the last three main steps in Section 4.1.1 (i.e., observation operator for the ensemble mean, observation operator for each ensemble member, and EnKF for calculating analysis increments) are enclosed in DLLs, without a DA algorithm instance corresponding to the second main step in Section 4.1.1. This is because the online DA procedure manager of DAFCC1 enables a DA algorithm instance to automatically obtain the ensemble mean of model DA variables (Section 3.3). Although both the third and fourth main steps correspond to the same GSI, they are transformed into two different DA algorithm instances, because the third is an ensemble algorithm (i.e., it operates on the data of the ensemble set) and the fourth is an individual algorithm (i.e., it operates on the data of each ensemble member). Moreover, we compiled the same GSI code into two separate DLLs, each of which corresponds to one of these two instances, to enable these two instances to use different memory space.

For each DA algorithm instance, three driving subroutines and the corresponding configuration were developed (Fig. 9). In fact, the two instances corresponding to GSI share the same driving subroutines but use different configurations (especially regarding the specification of "ensemble_operation"). To enable the GSI code and EnKF code to be used as DLL, we made the following slight modifications to the code.

1) We turned off the MPI initialization and finalized and replaced the original MPI communicator with the MPI communicator of the host model that can be obtained via DAFCC1.

2) We obtained the required model information and the declared input/output fields via DAFCC1, and turned off the corresponding I/O accesses.

To drive the DA algorithm instances, the WRF code was updated with the new subroutines for initializing, running, and finalizing all DA algorithm instances. Moreover, the functionality of outputting model background files can be turned off, because the data exchanges between WRF and the DA algorithm instances are automatically handled by DAFCC1 and the WRF ensemble can be run continuously throughout DA windows without stopping and restarting. As a result, DAFCC1 saves sets of data files and the corresponding I/O access operations, while only the observation files, fixed files, and the files for the data exchanges among the DA algorithm instances are reserved (compare Fig. 8b and Fig. 8a).

4.2 Example ensemble DA system of FIO-AOW

FIO-AOW, which previously used C-Coupler1 (Liu et al., 2014) for model coupling, has already been upgraded to C-Coupler2.0 by us (Fig. 10a). As GSI/EnKF and FIO-AOW share WRF, the development of the example ensemble DA system of FIO-AOW in Fig. 10b can significantly benefit from the DA system of WRF. In this ensemble DA system, the ensemble of WRF computes atmospheric analyses based on the ensemble DA sub-system in Section 4.1, while each ensemble member of other component models is impacted by the atmospheric analyses via model coupling. It only took the following steps to construct the example ensemble DA system.

1) Using the ensemble component manager, set up the two hierarchical levels of models shown in Fig. 11; i.e., the first level corresponds to all ensemble members of FIO-AOW while each member includes its three component models at the second level.

2) Merge the model code modifications, the DA algorithm instances, and configurations in the DA system of WRF into the example ensemble DA system FIO-AOW.

As well as being described by the flowchart involving the WRF and the DA algorithm instances in Fig. 8b, the example ensemble DA system of FIO-AOW follows the process layout in Fig. 12, which is essentially a real case of the process layout in Fig. 3.

5 Validation and evaluation of DAFCC1

In this section, we evaluate the correctness of DAFCC1 in developing a weakly coupled ensemble DA system based on the example ensemble DA system (referred to as the full-example-DA-system hereafter) described in Section 4, and will also validate DAFCC1 and evaluate the impact of DAFCC1 in accelerating DA based on the sub-system with WRF and GSI/EnKF (WRF-GSI/EnKF hereafter).

5.1 Experimental setup

The example ensemble DA system used in this validation and evaluation consists of WRF Version 4.0 (Wang et al., 2014), GSI version 3.6 and EnKF version 1.2, and the corresponding versions of POM and MASNUM used in FIO-AOW (Zhao et al., 2017; Wang et al., 2018). In EnKF version 1.2 the default settings are used; i.e., the EnSRF algorithm is used to calculate

analysis increments for ensemble DA, the inflation factor is 0.9 without smoothing, and the covariance is localized by distance
correlation function with horizontal localization radius of 400 km and vertical localization scale coefficient of 0.4. The example
ensemble DA system is run on a supercomputer of the Beijing Super Cloud Computing Center (BSCC) with the lustre file
system. Each computing node on the supercomputer includes two Intel Xeon E5-2678 v3 CPUs (Intel(R) Xeon(R) CPU), with
24 processor cores in total, and all computing nodes were connected with an InfiniBand network. The codes were compiled by
an Intel Fortran and C++ compiler at the optimization level O2, using an Intel MPI library. A maximum 3200 cores are used
for running the example ensemble DA system.

The WRF-GSI/EnKF integrates over an approximate geographical area generated from a Lambertian projection of the area
0°–50°N, 99°–160°E with center point at 35°N, 115°E. Initial fields and lateral boundary conditions (at 6 hour intervals) for
the ensemble run of WRF are taken from the NCEP Global Ensemble Forecast System (GEFS) (at 1° × 1° resolution)
(https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-ensemble-forecast-system-gefs). To configure
WRF, an existing physics suite 'CONUS' (https://www2.mmm.ucar.edu/wrf/users/ncar_convection_suite.php) and 32 vertical
sigma layers with the model top at 50 hPa are used. One-day integration on June 1st, 2016 is used for running the WRF-
GSI/EnKF. NCEP global GDAS Binary Universal Form for the Representation of meteorological data (BUFR;
https://www.emc.ncep.noaa.gov/mmb/data_processing/NCEP_BUFR_File_Structure.htm) and Prepared BUFR
(https://www.emc.ncep.noaa.gov/mmb/data_processing/prepbufr.doc/document.htm), including conventional observation
data and satellite radiation data, are assimilated every 6 hours (i.e., at 0000, 0600, 1200, and 1800 UTC). The air temperature
(T), specific humidity (QVAPOR), longitude and latitude wind (UV), and column disturbance dry air quality (MU) are the
variables analyzed in the data assimilation. The WRF-GSI/EnKF experiments are classified into four sets, where variations of
horizontal resolution (and the corresponding time step), number of ensemble members and process number (each process runs
on a distinct processor core) are considered (Tables 1 and 2).

All component models of the full-example-DA-system integrate over the same geographical area (0°–50°N, 99°–150°E)
with the same horizontal resolution of 0.5° × 0.5° but different time steps (100 s for WRF and 300 s for POM and MASNUM,
coupled by C-Coupler2.0 at 300 s intervals). More details of the model configurations can be found in Zhao et al. (2017). The
configuration of initial fields, lateral boundary conditions, and observations of WRF for the ensemble run of the full-example-
DA-system are the same as for WRF-GSI/EnKF. The full-example-DA-system integrates over 3 days (June 1st to 3rd, 2016),
while the first model day is considered as spin-up, and DA is performed every 6 hours in the last two model days with T, UV
and MU as DA variables.

5.2 Validation of DAFCC1

To validate DAFCC1, we compare the outputs of the two versions of WRF-GSI/EnKF: the original WRF-GSI/EnKF (hereafter
offline WRF-GSI/EnKF; https://dtcenter.org/community-code/gridpoint-statistical-interpolation-gsi/community-gsi-version-
3-6-enkf-version-1-2) and the new version of WRF-GSI/EnKF with DAFCC1 (hereafter online WRF-GSI/EnKF) introduced
in Section 4.1. As DAFCC1 improves only the data exchanges between a model and the DA algorithms, the simulation results

of an existing DA system should not change when it is adapted to use DAFCC1. We therefore employ a validation standard that the WRF-GSI/EnKF with DAFCC1 keeps bit-identical result with the original offline WRF-GSI/EnKF. DAFCC1 passes the validation test with all experimental setups in Table 2, where the binary data files output by WRF at the end of the 1-day integration are used for the comparison.

5.3 Impact in accelerating an offline DA

WRF-GSI/EnKF is further used to evaluate the impact of DAFCC1 in accelerating an offline DA, by comparing the execution time of the offline and online WRF-GSI/EnKF under each experimental setup in Table 2. Considering that all ensemble members of the online WRF-GSI/EnKF are integrated simultaneously, we run all ensemble members of the offline WRF-GSI/EnKF concurrently through a slight modification to the corresponding script, in order to make a fair comparison.

The impact of varying the number of ensemble members is evaluated based on Set 1 in Table 2. DAFCC1 obviously accelerates WRF-GSI/EnKF, and can achieve higher performance speedup with more ensemble members (Fig. 13a). This is because DAFCC1 significantly accelerates the DA for both GSI and EnKF (Fig. 13b-d). Similarly, DAFCC1 significantly accelerates the DA as well as WRF-GSI/EnKF under different process numbers (Fig. 14, corresponding to Set 2 in Table 2) and resolution (Fig. 15, corresponding to Set 3 in Table 2). Considering that more processor cores are generally required to accelerate the model run under higher resolution, we also make an evaluation based on Set 4 in Table 2, where concurrent changes in resolution and process number are made to achieve similar numbers of grid points per process throughout the experimental setups. This evaluation also demonstrates the correctness of DAFCC1 in accelerating the DA as well as WRF-GSI/EnKF (Fig. 16).

The performance speedups observed from Figs. 11-14 result mainly from the significant decrease in I/O accesses. Although the online WRF-GSI/EnKF still has to access the observation prior files (Section 4.1.1 and Fig. 8b), most I/O accesses correspond to the model ensemble background files and model ensemble analysis files, and these I/O accesses have been eliminated by DAFCC1 (Table 3). Moreover, more I/O accesses can be saved under higher resolution or more ensemble members.

We note that, the execution time of the offline GSI in Fig. 13c increases when using more ensemble members. This is reasonable, because more ensemble members introduce more I/O accesses, as shown in Table 3. We also note that, the execution time of the offline and online EnKF in Fig. 13d and Fig. 14 increases when using more ensemble members. This is because the current parallel version of EnKF does not achieve good scaling performance, and thus longer execution time can be observed when EnKF uses more processor cores.

5.4 Correctness in developing a weakly coupled ensemble DA system

We have successfully run the full-example-DA-system with ten ensemble members, which enables us to investigate the model fields before and after DA. We find that changes to the atmospheric fields resulting from DA can be observed; e.g., the bias regarding T is slightly decreased and the bias regarding UV is more obviously decreased after using DA, as shown in Fig. 17.

Changes to the atmospheric fields predicted based on the initial fields updated with the atmospheric analyses can also be
480   observed (e.g., the fields U and V in Fig. 18). Although only atmospheric analyses are computed currently, the model coupling
in the weakly coupled DA system makes ocean and wave fields impacted by the atmospheric analyses, and therefore changes
to the ocean and wave fields can be observed in a prediction (e.g., the fields SST and HS in Fig. 18).

6 Conclusions and discussion

In this paper, we propose a new common, flexible and efficient framework for weakly coupled ensemble data assimilation
485   based on C-Coupler2.0, DAFCC1. It provides simple APIs and a configuration file format to enable users to conveniently
integrate a DA method into a model as a procedure that can be directly called by the model, while still guaranteeing the
independence of configuration and compilation systems between the model and the DA method. The example weakly coupled
ensemble DA system in Section 4 and the evaluations in Section 5 demonstrate the correctness of DAFCC1 in both developing
a weakly coupled ensemble DA system and accelerating an offline DA system. The development of a DA system that only
490   employs a single model run but not an ensemble run can also benefit from the advantages of DAFCC1, while the functionality
of data exchanges will be automatically simplified without generating ensemble-set component models for saving extra
overhead.

DAFCC1 is able to automatically handle data exchanges between a model ensemble and a DA algorithm because its design
and implementation significantly benefit from C-Coupler2.0, which already has the functionalities of automatic coupling
495   generation and automatic data exchanges between different component models or within the same component model. DAFCC1
will therefore be an important functionality of the next generation of C-Coupler (C-Coupler3) that is planned to be released no
later than 2022. Although the example ensemble DA system of FIO-AOW developed in this work only computes atmospheric
analyses currently, the future work similar to adapting GSI/EnKF to DAFCC1 can be conducted to further enable the
computation of ocean or wave analyses. Moreover, we have considered software extendibility when designing and
500   implementing DAFCC1, which will enable us to conveniently achieve upgrades either for strongly coupled ensemble DA
systems or for more types of data exchange operations in the future. As shown in Fig. 8, the I/O accesses to the observation
prior files for the data exchanges between DA algorithms are still retained after using DAFCC1. Although they are not currently
a performance bottleneck (Table 3), we will investigate how to avoid these types of I/O accesses when further upgrading
DAFCC1.
505   Regarding the evaluations in Section 5, we can only use at most 3200 processor cores, which limits the maximum number
of cores per ensemble member. Consequently, we use relatively coarse resolutions of WRF and FIO-AOW. However, the
results in Fig. 16 from the experiment Set 4 in Table 2 indicate that DAFCC1 will also obviously accelerate the DA system
when using a finer resolution and more processor cores, because it will also significantly decrease I/O accesses. DAFCC1 can
tackle the technical challenges in developing or accelerating a DA system, but cannot contribute to improvements in simulation
510   results that generally depend on scientific settings which must be determined in the research environment (e.g., the DA

algorithm configuration, the inflation factor, localization settings, initial states of the model ensemble run). Consequently, we did not examine the improvements in simulation results resulting from the full-example-DA-system based on various variables in Section 5.4, but only made a simple comparison of simulation results demonstrating that the full-example-DA-system can successfully run and produce simulation results.

515   The offline implementation of a DA system that relies on disk files and restart functionalities of models and DA algorithms can be a robust strategy when it comes to massively parallel computing where the risk of random task failures generally increases with more processor cores used by a task, because a failed task that corresponds to an ensemble member can be resumed from the corresponding restart files. The online implementation that unifies all ensemble members into a task enables to significantly increase the number of cores used by a task. At the same time of enlarging the risk of random task failures, the

520   online implementation can decrease such risk because it can significantly reduce disk file accesses that are generally an important source of task failures. The robustness of an online implementation can be further improved through developing the restart capability of the DA system based on the restart capabilities of the model and C-Coupler2, while users are enabled to flexibly set the restart-file-writing frequency for the online implementation that can be lower than the corresponding frequency for an online implementation generally determined by observation data frequencies. Moreover, the impact of the overhead of

525   writing restart files in an online implementation can be further decreased via asynchronous I/O support.

Code availability. The source code of DAFCC1 can be viewed via https://doi.org/10.5281/zenodo.3739729 (please contact us for authorization before using DAFCC1 for developing a system). The original source code and scripts corresponding to WRF and GSI/EnKF can be download from https://www2.mmm.ucar.edu/wrf/users/downloads.html and https://dtcenter.org/com-GSI/users/downloads/index.php respectively. For the source code of FIO-AOW, please contact the authors of (Zhao et al.,

530   2017; Wang et al., 2018). The additional codes, configurations, scripts and guidelines for developing and running the example weakly coupled ensemble DA system can also be download from https://doi.org/10.5281/zenodo.3774710.

Author contributions. CS was responsible for code development, software testing and experimental evaluation of DAFCC1 with the example DA system, contributed to the motivation and design of DAFCC1 and co-led paper writing. LL initiated this research, was responsible for the motivation and design of DAFCC1, co-supervised CS, and co-led paper writing. RL, XY and

535   HY contributed to code development and software testing. BZ, GW, JL and FQ contributed to the development of the example DA system. BW contributed to scientific requirements and the motivation, and co-supervised CS. All authors contributed to improvement of ideas and paper writing.

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

Acknowledgements. This work was jointly supported in part by the National Key Research Project of China (grant no. 2017YFC1501903), the National Natural Science Foundation of China(grant no. 41875127) and the National Key Research Project of China (grant no. 2019YFA0606604).

References

Andersson, E., Haseler, J., Unden, P., Courtier, P., Kelly, G., Vasiljevic, D., and Thepaut, J.: The ECMWF implementation of three-dimensional variational assimilation (3D-Var). III: Experimental results, Q. J. Roy. Meteor. Soc., 124, 1831-1860, 1998.

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

Anderson, J. and Collins, N.: Scalable implementations of ensemble filter algorithms for data assimilation, J. Atmos. Ocean Technol., 24, 1452-1463, 2007.

Bishop, C. and Hodyss, D.: Adaptive ensemble covariance localization in ensemble 4D-VAR state estimation, Mon. Weather Rev., 139, 1241–1255, 2011.

Blumberg, A. and Mellor, G.: A description of a three-dimensional coastal ocean circulation model, in Three-Dimensional Coastal Ocean Models, edited by N. S. Heaps, pp. 1–16, AGU, Washington, D. C, 1987.

Bonavita, M., Isaksen, L., and Holm, E.: On the use of EDA background-error variances in the ECMWF 4D-Var, Q. J. R. Meteorol. Soc., 138, 1540-1559, 2012.

Bonavita, M., Holm, E., Isaksen, L., and Fisher, M. A.: The evolution of the ECMWF hybrid data assimilation system, Q. J. Roy. Meteor. Soc., 142, 287-303, 2016.

Brown, A., Milton, S., Cullen, M., Golding, B., Mitchell, J., and Shelly, A.: Unified modeling and prediction of weather and climate: a 25 year journey, B. Am. Meteorol. Soc., 93, 1865–1877, https://doi.org/10.1175/BAMS-D-12-00018.1, 2012.

Browne, P., de Rosnay, P., Zuo, H., Bennett, A., and Dawson, A.: Weakly coupled ocean-atmosphere data assimilation in the ECMWF NWP system, Remote Sens., 11, 1-24, 2005.

Browne, P., 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.

Brunet, G., Jones, S., and Ruti, P.: Seamless prediction of the Earth System: from minutes to months, Tech. Rep. WWOSC-2014, World Meteorological Organization, 2015.

Buehner, M., McTaggart-Cowan, R., Beaulne, A., Charette, C., Garand, L., Heilliette, S., Lapalme, E., Laroche, S., Macpherson, S.R., Morneau, J. and Zadra, A.: Implementation of deterministic weather forecasting systems based on ensemble-variational data assimilation at Environment Canada. Part I: the global system, Mon. Weather Rev., 143, 2532–2559, 2015.

570 Courtier, P., Andersson, E., Heckley, W., Vasiljevic, D., Hamrud, M., Hollingsworth, A., and Pailleux, J.: The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation, Q. J. Roy. Meteor. Soc., 124, 1783-1807, 1998.

Courtier, P., Thepaut, J., and Hollingsworth, A.: A strategy for operational implementation of 4D-Var, using an incremental approach, Q. J. Roy. Meteor. Soc., 120, 1367-1387, 1994.

575 Craig, A., Vertenstein, M., and Jacob, R.: A new flexible coupler for earth system modeling developed for CCSM4 and CESM1, Int. J. High Perform. Comput. Appl., 26, 31-42, 2012.

Craig, A., Valcke, S., and Coquart, L.: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0, Geosci. Model Dev., 10, 3297-3308, https://doi.org/10.5194/gmd-10-3297-2017, 2017.

Etherton, B. and Bishop, C.: Resilience of hybrid ensemble/3DVAR analysis schemes to model error and ensemble covariance
580 error, Mon. Weather Rev., 132, 1065-1080, 2004.

Evensen, G.: The ensemble kalman filter: theoretical formulation and practical implementation, Ocean Dyn., 53, 343-367, 2003.

Fisher, M.: Background error covariance modelling, in: Proceedings of Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean, Reading, UK, 8-12 September 2003, 45–63, 2003.

585 Fujii, Y., Kamachi, M., Nakaegawa, T., Yasuda, T., Yamanaka, G., Toyoda, T., Ando, K. and Matsumoto, S.: Assimilating ocean observation data for ENSO monitoring and forecasting, in: Climate Variability - Some Aspects, Challenges and Prospects, edited by: Hannachi, A., InTechOpen, Rijeka, Croatia, 75-98, 2011.

Fujii, Y., Nakaegawa, T., Matsumoto, S., Yasuda, T., Yamanaka, G., and Kamachi, M.: Coupled climate simulation by constraining ocean fields in a coupled model with ocean data, J. Clim., 22, 5541-5557, 2009.

590 Gandin, L.: Objective analysis of meteorological fields. Jerusalem: Israel program for scientific translations, 1965

Gauthier, P., Charette, C., Fillion, L., Koclas, P., and Laroche, S.: Implementation of a 3D variational data assimilation system at the Canadian Meteorological Center. Part I: The global analysis, Atmos. Ocean, 37, 103-156, 1999.

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 Dyn., 69,
595 1217-1237, 2019.

Hamill, T.: A hybrid ensemble kalman filter-3D variational analysis scheme, Mon. Weather Rev., 128, 2905, 2000.

Heinzeller, D., Duda, M. G., and Kunstmann, H.: Towards convection-resolving, global atmospheric simulations with the Model for Prediction Across Scales (MPAS): an extreme scaling experiment, Geosci. Model Dev., 8, 6987-7061, 2016.

Hoke, J. and Anthes, R.: The initialization of numerical models by a dynamic initialization technique, Mon. Weather Rev.,
600 104, 1551–1556, 1976.

Hoskins, B.: The potential for skill across the range of the seamless weather-climate prediction problem: a stimulus for our science, Q. J. Roy. Meteor. Soc., 139, 573-584, 2013.

Houtekamer, P., and Mitchell, H.: Data assimilation using an ensemble kalman filter technique. Mon. Weather Rev., 126, 796-811, 1998.

605 Hunt, B., Kostelich, E., and Szunyogh, I.: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter, International Symposium on Physical Design, 230, 112-126, 2007.

Kalnay, E.: Atmospheric modeling, data assimilation and predictability, Cambridge University Press, Cambridge, UK, pp. 364, 2002.

Laloyaux, P., Thepaut, J., and Dee, D.: Impact of scatterometer surface wind data in the ECMWF coupled assimilation system, 610 Mon. Weather Rev., 144, 1203-1217, 2016.

Laloyaux, P., Frolov, S., Benjamin Ménétrier, and Bonavita, M.: Implicit and explicit cross-correlations in coupled data assimilation, Q. J. R. Meteorol. Soc., 144, 2018.

Lea, D., Mirouze, I., Martin, M., King, 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.

615 Liu, H., Hu, M., Ge, G., Stark, D., Shao, H., Newman, K., and Whitaker, J.: Ensemble Kalman Filter (EnKF) User's Guide Version 1.3, Developmental Testbed Center, available at: https://dtcenter.org/community-code/ensemble-kalman-filter-system-enkf/documentation, 80 pp, 2018a.

Liu, L., Yang, G., Wang, B., Zhang, C., Li, R., Zhang, Z., Ji, Y., and Wang, L.: C-Coupler1: a Chinese community coupler for Earth System Modeling, Geosci. Model Dev., 7, 2281-2302, doi: 10.5194/gmd-7-2281-2014, 2014.

620 Liu, L., Zhang, C., Li, R., Wang, B., and Yang, G.: C-Coupler2: a flexible and user-friendly community coupler for model coupling and nesting, Geosci. Model Dev., 11, 3557-3586, https://doi.org/10.5194/gmd-11-3557-2018, 2018b.

Lorenc, A., Ballard, S., Bell, R., Ingleby, N., Andrews, P., Barker, D., Bray, J., Clayton, A., Dalby, T., Li, D., Payne, T., and Saunders, F.: The Met. Office global three-dimensional variational data assimilation scheme, Q. J. Roy. Meteor. Soc., 126, 2991-3012, 2000.

625 Lorenc, A.: The potential of the ensemble Kalman filter for NWP-A comparison with 4D-VAR, Q. J. Roy. Meteor. Soc., 129: 3183-3203, 2003a.

Lorenc, A.: Modelling of error covariances by 4D-Var data assimilation, Q. J. Roy. Meteor. Soc., 129, 3167-3182, 2003b.

Lu, F., Liu, Z., Zhang S., and Liu Y.: Strongly coupled data assimilation using leading averaged coupled covariance (LACC). Part I: Simple model study, Mon. Weather Rev., 143, 3823–3837, doi: 10.1175/MWR-D-14-00322.1, 2015a.

630 Lu, F., Liu, Z., Zhang, S., Liu, Y., and Jacob, R.: Strongly coupled data assimilation using leading averaged coupled covariance (LACC). Part II: GCM Experiments, Mon. Weather Rev., 143, 4645-4659, doi: 10.1175/MWR-D-14-00322.1, 2015b.

Ma, X., Lu, X., Yu, M., Zhu, H., and Chen, J.: Progress on hybrid ensemble-variational data assimilation in numerical weather prediction, J. Trop. Meteorol., 30, 1188-1195, 2014.

Mochizuki, T., Masuda, S., Ishikawa, Y., and Awaji, T.: Multiyear climate prediction with initialization based on 4D-Var data 635 assimilation, Geophys. Res. Lett., 43, 3903-3910, 2016.

Mulholland, D., Laloyaux, P., Haines, K., and Balmaseda, M.: Origin and impact of initialization shocks in coupled atmosphere–ocean forecasts, Mon. Weather Rev., 143, 4631–4644, doi: 10.1175/MWR-D-15-0076.1, 2015.

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

640 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., pp. 63–83, World Scientific, 2005.

Nerger, L., Tang, Q., and Mu, L.: Efficient ensemble data assimilation for coupled models with the Parallel Data Assimilation Framework: example of AWI-CM (AWI-CM-PDAF 1.0), Geosci. Model Dev., 13, 4305–4321,
645 https://doi.org/10.5194/gmd-13-4305-2020, 2020.

Oke, P., Allen, J., Miller, R., Egbert, G., and Kosro, P.: Assimilation of surface velocity data into a primitive equation coastal ocean model. J. Geophys. Res, 107, 2002.

Palmer, T., Doblas-Reyes, F., Weisheimer, A. and Rodwell, M.: Toward seamless prediction: Calibration of climate change projections using seasonal forecasts, Bull. Am. Meteorol. Soc., 89, 459-470, 2008.

650 Penny, S., 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.

Qiao, F., Zhao, W., Yin, X., Huang, X., Liu, X., Shu, Q., Wang, G., Song, Z., Liu, H., Yang, G., and Yuan, Y.: A highly
655 effective global surface wave numerical simulation with ultra-high resolution, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '16), IEEE Press, Piscataway, NJ, USA, doi: 10.1109/SC.2016.4, 2016.

Rabier, F., Jarvinen, H., Klinker, E., Mahfouf, J., and Simmons, A.: The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics, Q. J. Roy. Meteor. Soc., 126, 1143-
660 1170, 2007.

Saha, S., Moorthi, S., Pan, H., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., Liu, H., Stokes, D., Grumbine, R., Gayno, G., Wang, J., Hou, Y., Chuang, H., Juang, H. H., Sela, J., Iredell, M. T. R., Kleist, D., van Delst, P., Keyser, D., Derber, J., Ek, M., Meng, J., Wei, H., Yang, R., Lord, St., Van Den Dool, H., Kumar, A., Wang, W., Long, C., Chelliah, M., Xue, Y., Huang, B., Schemm, J., Ebisuzaki, W., Lin, R., Xie, P., Chen, M., Zhou, S., Higgins, W., Zou,
665 C., Liu, Q., Chen, Y., Han, Y., Cucurull, L., Reynolds, R. W., Rutledge, G., and Goldberg, M.: The NCEP Climate Forecast System Reanalysis, Bull. Am. Meteorol. Soc., 91, 1015-1057, 2010.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y., Chuang, H., Iredell, M., Ek, M., Meng, J., Yang, R., Mendez, M., Van Den Dool, H., Zhang, Q., Wang, W., Chen, M., and Becker, E.: The NCEP Climate Forecast System Version 2, J. Clim., 27, 2185–2208, http://dx.doi.org/10.1175/JCLI-D-12-00823.1, 2014.

670 Sakov, P., Counillon, F., Bertino, L., Lisæter, K. A., Oke, P. R., and Korablev, A.: TOPAZ4: an ocean-sea ice data assimilation system for the North Atlantic and Arctic, Ocean Sci., 8, 633-656, 2012.

Shao, H., Derber, J., Huang, X. Y., Hu, M., Newman, K., Stark, D., Lueken, M., Zhou, C., Nance, L., Kuo, Y. H., and Brown, B.: Bridging Research to Operations Transitions: Status and Plans of Community GSI. Bull. Amer. Meteor. Soc., 97, 1427-1440, doi: 10.1175/BAMS-D-13-00245.1, 2016.

675 Skachko, S., Buehner, M., Laroche, S., Lapalme, E., Smith, G., Roy, F., Surcel-Colan, D., Bélanger, J., and Garand, L.: Weakly coupled atmosphere-ocean data assimilation in the Canadian global prediction system (v1), Geosci. Model Dev., 12, 5097-5112, 2019.

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

680 Sugiura, N., Awaji, T., Masuda, S., Mochizuki, T., Toyoda, T., Miyama, T., Igarashi, H., and Ishikawa, Y.: Development of a 4-dimensional variational coupled data assimilation system for enhanced analysis and prediction of seasonal to interannual climate variations, J. Geophys. Res., 113, C10017, doi: 10.1029/2008JC004741, 2008.

Tardif, R., Hakim, G., and Snyder, C.: Coupled atmosphere–ocean data assimilation experiments with a low-order climate model, Clim. Dyn., 43, 1631-1643, doi: 10.1007/s00382-013-1989-0, 2014.

685 Tardif, R., Hakim, G., and Snyder, C.: Coupled atmosphere-ocean data assimilation experiments with a low-order model and CMIP5 model data, Clim. Dyn., 45, 1415-1427, doi: 10.1007/s00382-014-2390-3, 2015.

Valcke, S., Balaji, V., Craig, A., Deluca, C., Dunlap, R., Ford, R. W., Jacob, R., Larson, J., Okuinghttons, R., Riley, G., and Vertenstein, M: Coupling technologies for Earth System Modelling, Geosci. Model Dev., 5, 1589-1596, 2012.

Vidard, P. A., Le Dimet, F. X., and Piacentini, A.: Determination of optimal nudging coefficients, Tellus, 55A, 1–15, 2003.

690 Wang, G., Qiao F., and Xia C.: Parallelization of a coupled wave-circulation model and its application, Ocean Dyn., 60(2), 331–339, doi:10.1007/s10236-010-0274-6, 2010.

Wang, G., Zhao, B., Qiao, F., and Zhao, C.: Rapid intensification of Super Typhoon Haiyan: the important role of a warm-core ocean eddy, Ocean Dyn, 68, 1649-1661, 2018.

Wang, W., Barker, D., Bray J., Bruye`re C., Duda M., Dudhia J., Gill D., and Michalakes J.: WRF Version 3 Modeling
695 System User's Guide, available at http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/contents.html, 2014.

Wang, X., Barker, D., Snyder, C., and Hamill, T.: A hybrid ETKF-3DVAR data assimilation scheme for the WRF model. Part I: Observing system simulation experiment, Mon. Weather Rev., 136, 5116–5131, 2008.

Wang, X., Parrish, D., Kleist, D., and Whitaker, J.: GSI 3DVar-based ensemble-variational hybrid data assimilation for NCEP global forecast system: Single-resolution experiments, Mon. Weather Rev., 141, 4098 – 4117, 2013.

700 Whitaker, J.and Hamill, T. M. Ensemble Data Assimilation without Perturbed Observations, Mon. Weather Rev., 130, 1913-1924, 2012.

Yang, X., Rosati, A., Zhang, S., Delworth, T., Gudgel, R., Zhang, R., Vecchi, G., Anderson, W., Chang, Y., DelSole, T., Dixon, K., Msadek, R., Stern, W., Wittenberg, A., and Zeng, F.: A predictable AMO-like pattern in the GFDL fully coupled ensemble initialization and decadal forecasting system, J. Clim., 26, 650-661, 2013.

705 Yang, Y., Qiao, F., Zhao, W., Teng, Y., and Yuan, Y.: MASNUM ocean wave numerical model in spherical coordinates and its application, Acta Oceanol. Sin., 27, 1–7, 2005.

Zhao, B., Qiao, F., Cavaleri, L., Wang, G., Bertotti, L., and Liu, L: Sensitivity of typhoon modeling to surface waves and rainfall, J. Geophys. Res. Oceans, 122, 1702–1723, doi:10.1002/2016JC012262, 2017.

Zhang, S., Harrison, M., Wittenberg, A., Rosati, A., Anderson, J., and Balaji, V.: Initialization of an ENSO forecast system

710 using a parallelized ensemble filter, Mon. Weather Rev., 133, 3176–3201, 2005.

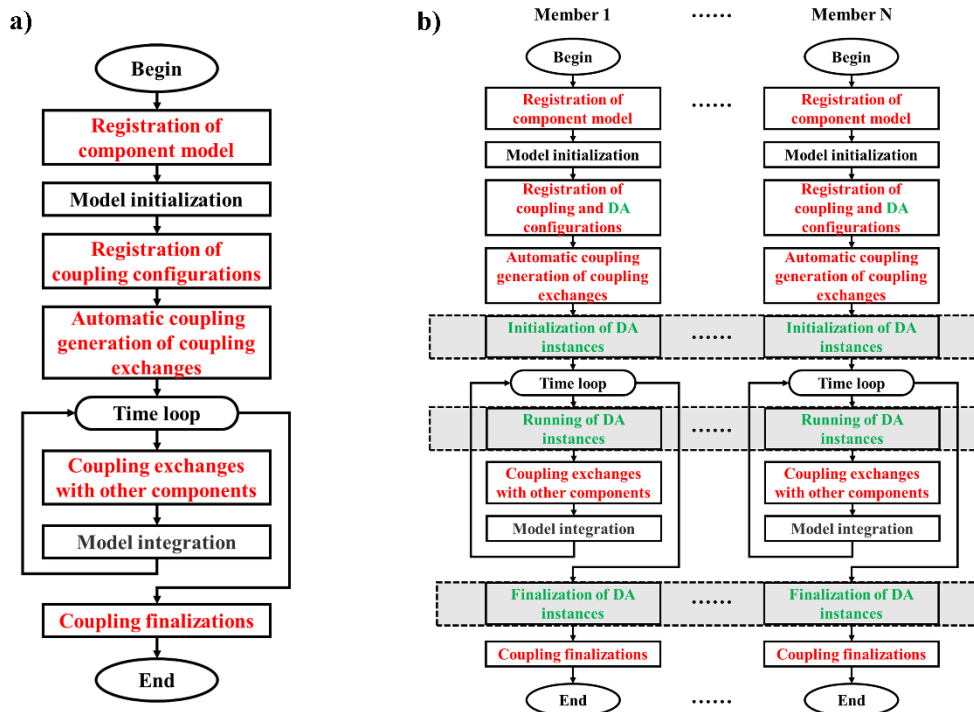
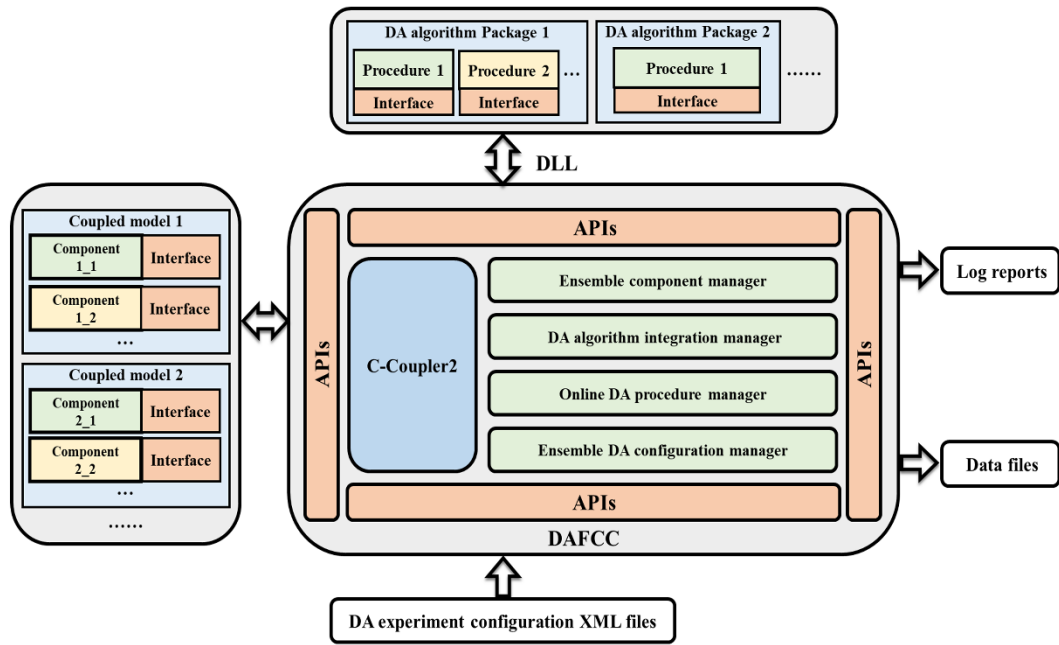
Zhang, S., Harrison, M., Rosati, A., and Wittenberg, A.: System design and evaluation of coupled ensemble data assimilation for global oceanic climate studies, Mon. Weather Rev., 135, 3541–3564, doi: 10.1175/MWR3466.1, 2007.

715 **Figure 1. Model flowchart with C-Coupler2.0 (a) and new model flowchart with ensemble DA based on C-Coupler2.0 (b). Black font indicates the major steps in the original flowchart of a component model without coupling, red font indicates the major steps for achieving coupling exchanges among component models with C-Coupler2.0, and green font indicates the new steps for achieving coupling exchanges between a DA algorithm and a model ensemble. The gray shadow in a dashed rectangle indicates that all members in a model ensemble cooperatively work together for the corresponding step.**

720

Figure 2. Architecture of DAFCC1.

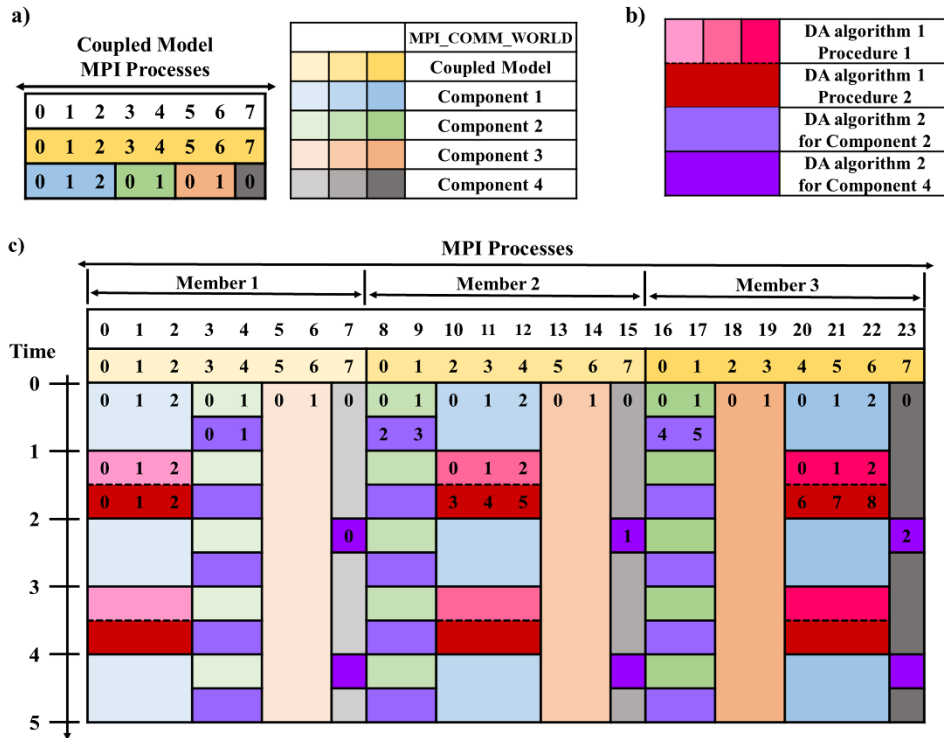
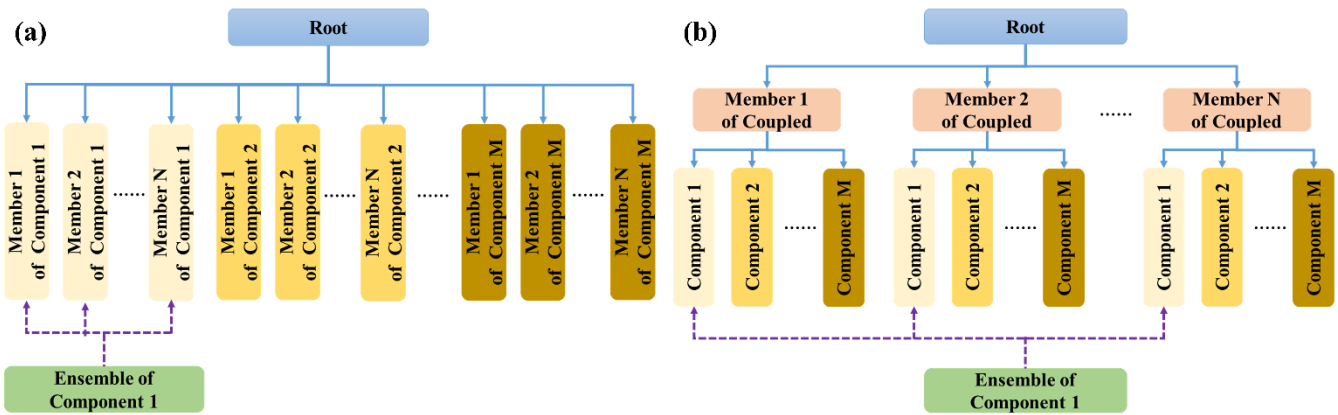



Figure 3. Example of running a DAFCC1-based weakly coupled ensemble DA system with three ensemble members. (a) Each ensemble member of the coupled model (yellow series) uses 8 MPI processes, where component 1 (blue series) uses three MPI processes, component 2 (green series) uses two MPI processes, component 3 (orange series) uses two MPI processes, and component 4 (grey series) uses one MPI process. (b) DA algorithm 1 and two instances of DA algorithm 2 (purple series) are used in this DA system, where DA algorithm 1 includes procedure 1 (pink series) and procedure 2 (red). (c) Execution of the DA system: the process layout of ensemble members of component models, the process layout of DA algorithms, and the alternative execution of a DA algorithm and the corresponding component model. Each number in the colored box in (a) and (c) indicates the process ID in the corresponding local communicator of a member of the coupled model, a member of a component model, or all members of a component model.

```
mpirun –np N1_1 Comp1 namelist CCPL_ensemble_3_1 : –np N1_2 Comp2 namelist
CCPL_ensemble_3_1 : –np N2_2 Comp2 namelist CCPL_ensemble_3_2 : –np N2_1 Comp1
namelist CCPL_ensemble_3_2 : –np N3_1 Comp1 namelist CCPL_ensemble_3_3 : –np N3_2
Comp2 namelist CCPL_ensemble_3_3
```

Figure 4. Example of the command for submitting an MPI run of three ensemble members of a coupled model that consists of Comp1 and Comp2. Comp1 can be before Comp2 at the second ensemble member, and the process numbers N1_1, N2_1, and N3_1 of Comp1 at different ensemble members can be different.

740 **Figure 5. Two examples of the organization of N ensemble members of a coupled model consisting of M component models. (a) Single-level organizational architecture of all ensemble members of the component models in the coupled model. (b) Two-hierarchical-levels organizational architecture. All ensemble members of the coupled model are organized as the first level with all component models from each ensemble member of the coupled model at the second level. An ensemble-set component model that covers all ensemble members of component model 1 is generated as an example for using the DA algorithm in ensemble component**
745 **manager.**

a)

1	**Subroutine** *DA*_**CCPL_INIT**
2	Obtain model information and grids from C-Coupler2
3	Initialize grid and parallel decomposition
4	Allocate memory spaces of variables
5	Initialize other operations
6	Register the parallel decompositions, required grids and field instances to C-Coupler2
7	Declare field instances as implicit input or output arguments
8	**End subroutine** *DA*_**CCPL_INIT**

b)

1	**Subroutine** *DA*_**CCPL_RUN**
2	DA analysis
3	**End subroutine** D*A*_**CCPL_RUN**

c)

1	**Subroutine** *DA*_**CCPL_FIN**
2	DA finalize
3	**End subroutine** *DA*_**CCPL_FIN**

750 **Figure 6. Example of the driving subroutines in a DA algorithm. (a) Initialization driving subroutine. (b) Running driving subroutine. (c) Finalization driving subroutine. The name of the DA algorithm "DA" is used as the prefix of the three driving subroutines; different suffixes are used for distinction. Black font indicates original functionalities of the DA algorithm, while red font indicates additional operations to perform online data exchanges between the model and DA algorithm.**

```
 1   <root>
 2   <da_instance name="DA_algorithm1_procedure1" status="on">
 3       <external_procedures status="on" procedures_name="algorithm1_procedure1" dll_name="lib_da1_p1.so"/>
 4       <periodic_timer status="on" period_unit="seconds" period_count="21600" local_lag_count="0"/>
 5       <field_instances status="on" time_processing="inst" ensemble_operation="none"/>
 6       <processing_control status="on" >
 7           <working_directory status="off" path=""/>
 8           <config_scripts status="on">
 9               <pre_instance_script status="on" name="da1_p1_online_run.sh"/>
10               <post_instance_script status="off" name=""/>
11           </config_scripts>
12       </processing_control>
13   </da_instance>
14   <da_instance name="DA_algorithm1_procedure2" status="on">
15       <external_procedures status="on" procedures_name="algorithm1_procedure2" dll_name="lib_da1_p2.so"/>
16       <periodic_timer status="on" period_unit="seconds" period_count="21600" local_lag_count="0"/>
17       <field_instances status="on" time_processing="inst" ensemble_operation="aver">
18           
19           
20       </field_instances>
21       <processing_control status="on" >
22           <working_directory status="on" path="./experiment/da1"/>
23           <config_scripts status="on">
24               <pre_instance_script status="off" name=""/>
25               <post_instance_script status="on" name="da1_p2_online_run.sh"/>
26           </config_scripts>
27       </processing_control>
28   </da_instance>
29   <da_instance name="DA_algorithm2" status="on">
30       <external_procedures status="on" procedures_name="algorithm2" dll_name="lib_da2.so"/>
31       <periodic_timer status="on" period_unit="seconds" period_count="21600" local_lag_count="0"/>
32       <field_instances status="on" time_processing="inst" ensemble_operation="gather"/>
33       <processing_control status="on" >
34           <working_directory status="on" path="./experiment/da2"/>
35           <config_scripts status="on">
36               <pre_instance_script status="on" name="./scripts/da2_pre_online_run.sh"/>
37               <post_instance_script status="on" name="./scripts/da2_post_online_run.sh"/>
38           </config_scripts>
39       </processing_control>
40   </da_instance>
41   </root>
```

755   **Figure 7. Example of the XML configuration for a DA experiment.**

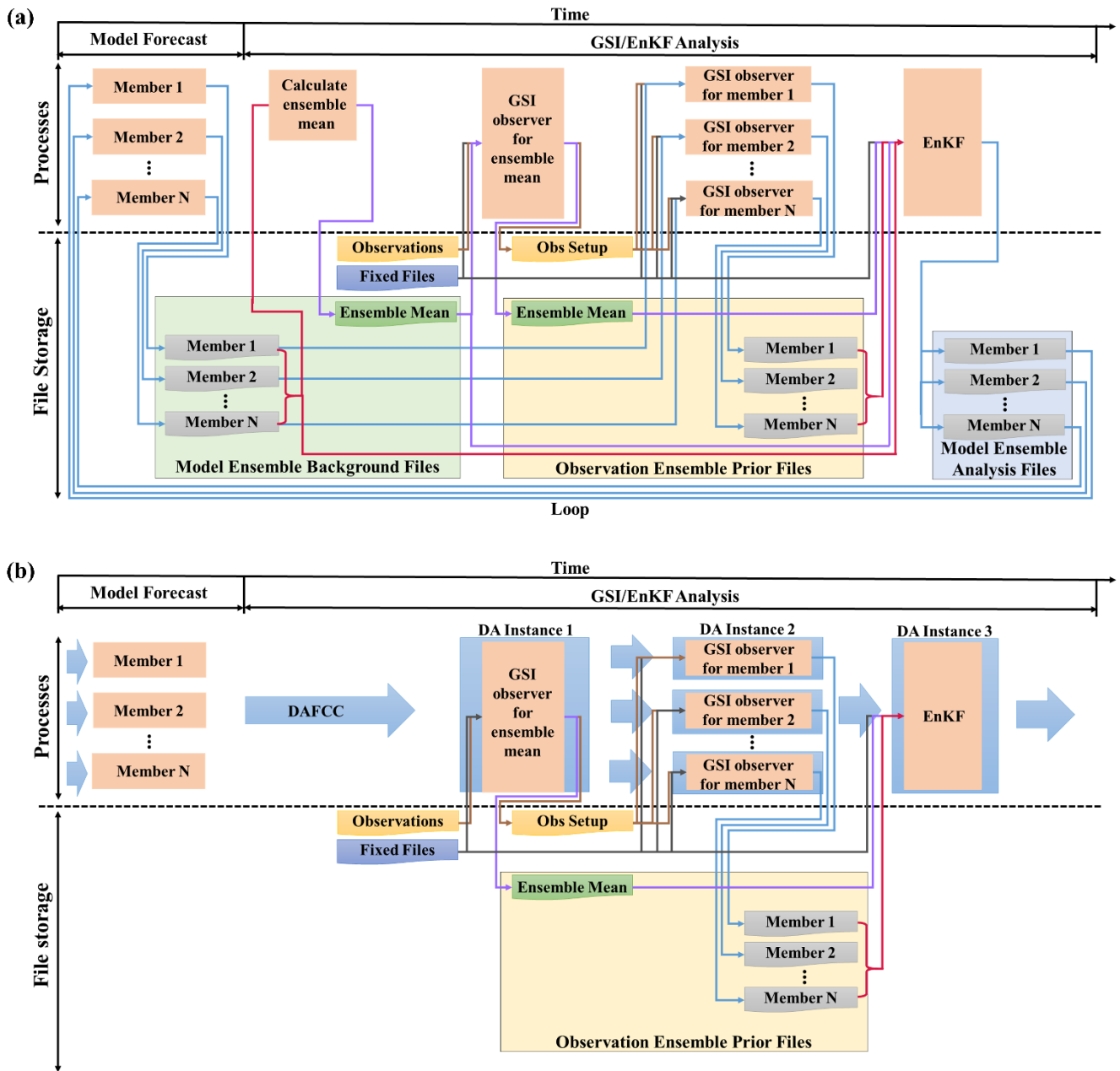
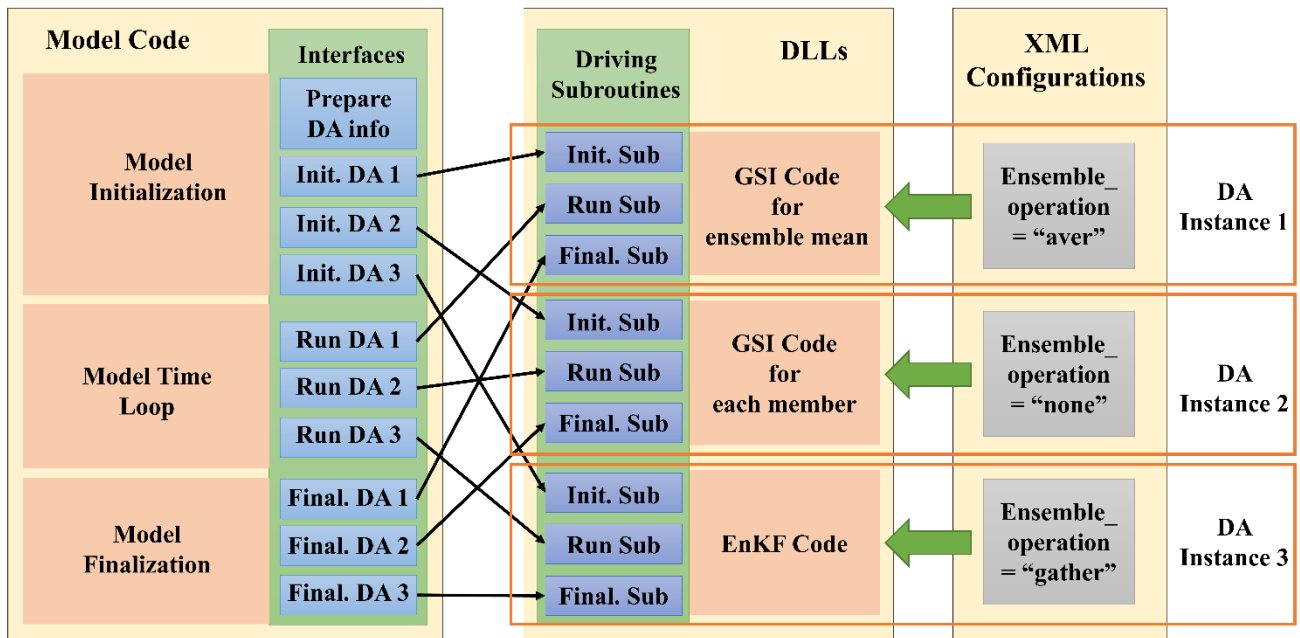



Figure 8. Running processes and data scheduling for (a) original GSI/EnKF used as a pure ensemble DA system, and (b) modified
GSI/EnKF based on DAFCC1. Orange rectangles in the Processes panel indicate different running processes, while thick blue arrows
mark data scheduling based on DAFCC1. Rectangles of various colors with a curved lower edge in the File Storage panel indicate
different files, while arrows of different colors indicate the scheduling of corresponding files.

765   **Figure 9. Modifications of model code and the invoking of relationships to the DA algorithm in the example ensemble DA system.**

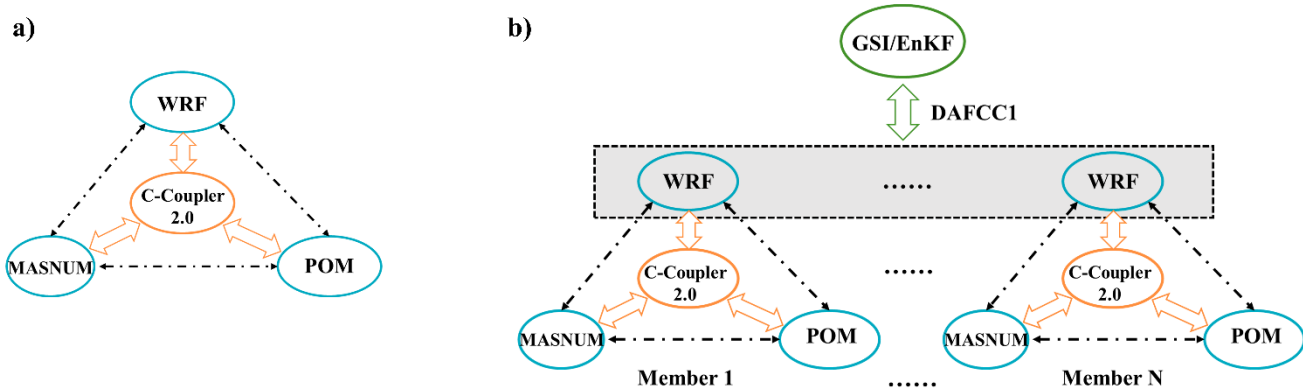



Figure 10. Architecture of FIO-AOW (a) and the corresponding example ensemble DA system (b). The gray shadow in a dashed rectangle in (b) indicates that atmospheric analyses are computed by GSI/EnKF that has been coupled with the ensemble of WRF based on DAFCC1.

770

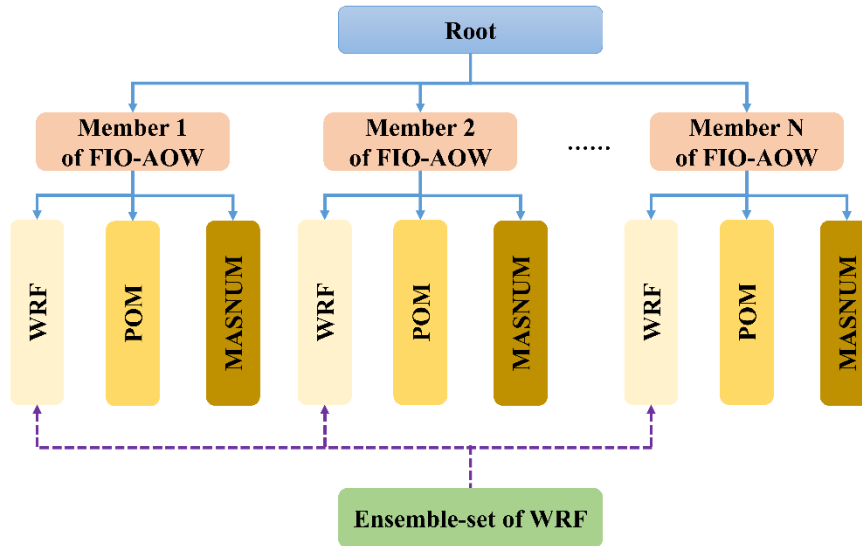



Figure 11. Two-hierarchical-level organizational architecture for N ensemble members of FIO-AOW consisting of WRF, POM, and MASNUM. All ensemble members of FIO-AOW are organized as the first level with all component models in each ensemble member at the second level. An ensemble-set that covers all ensemble members of component model WRF is generated by the ensemble component manager.

775

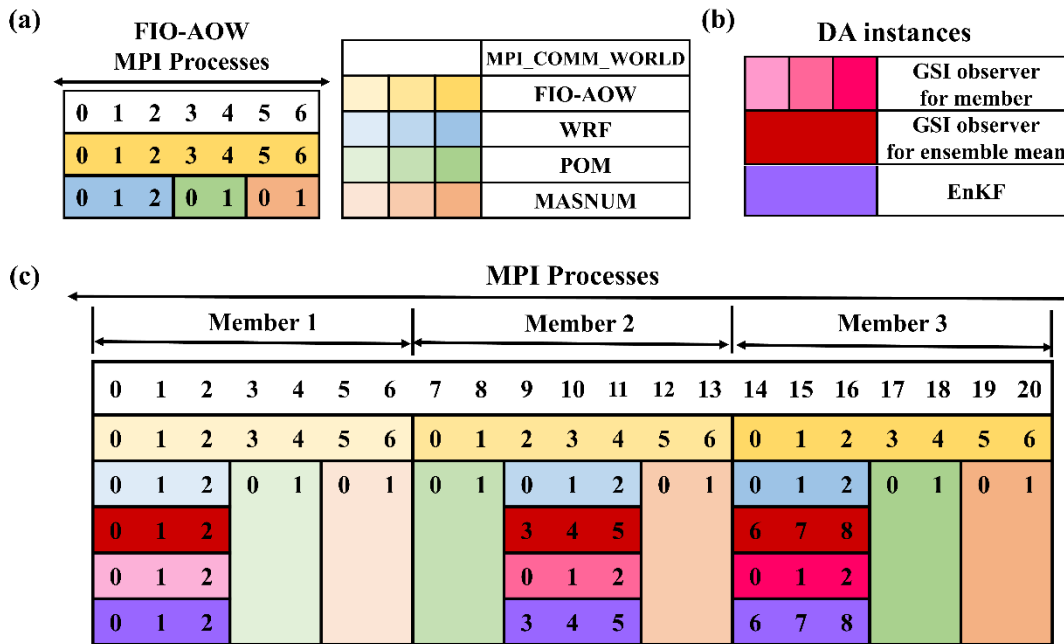



Figure 12. Example of the process layout of the example ensemble DA system FIO-AOW. (a) Each ensemble member of FIO-AOW (yellow series) uses 7 MPI processes, where WRF (blue series) uses 3 MPI processes, POM (green series) uses 2 MPI processes, and MASNUM (orange series) uses 2 MPI processes. (b) Two DA algorithm instances of GSI are adopted for each member (pink series) and ensemble mean (red) respectively following another DA algorithm instance of EnKF in this DA system. (c) Process layout of the DA system: the process layout of ensemble members of component models and the process layout of DA algorithms. Each number in the colored boxes in (a) and (c) indicates the process ID in the corresponding local communicator of a member of the coupled model, a member of a component model, or all members of a component model.

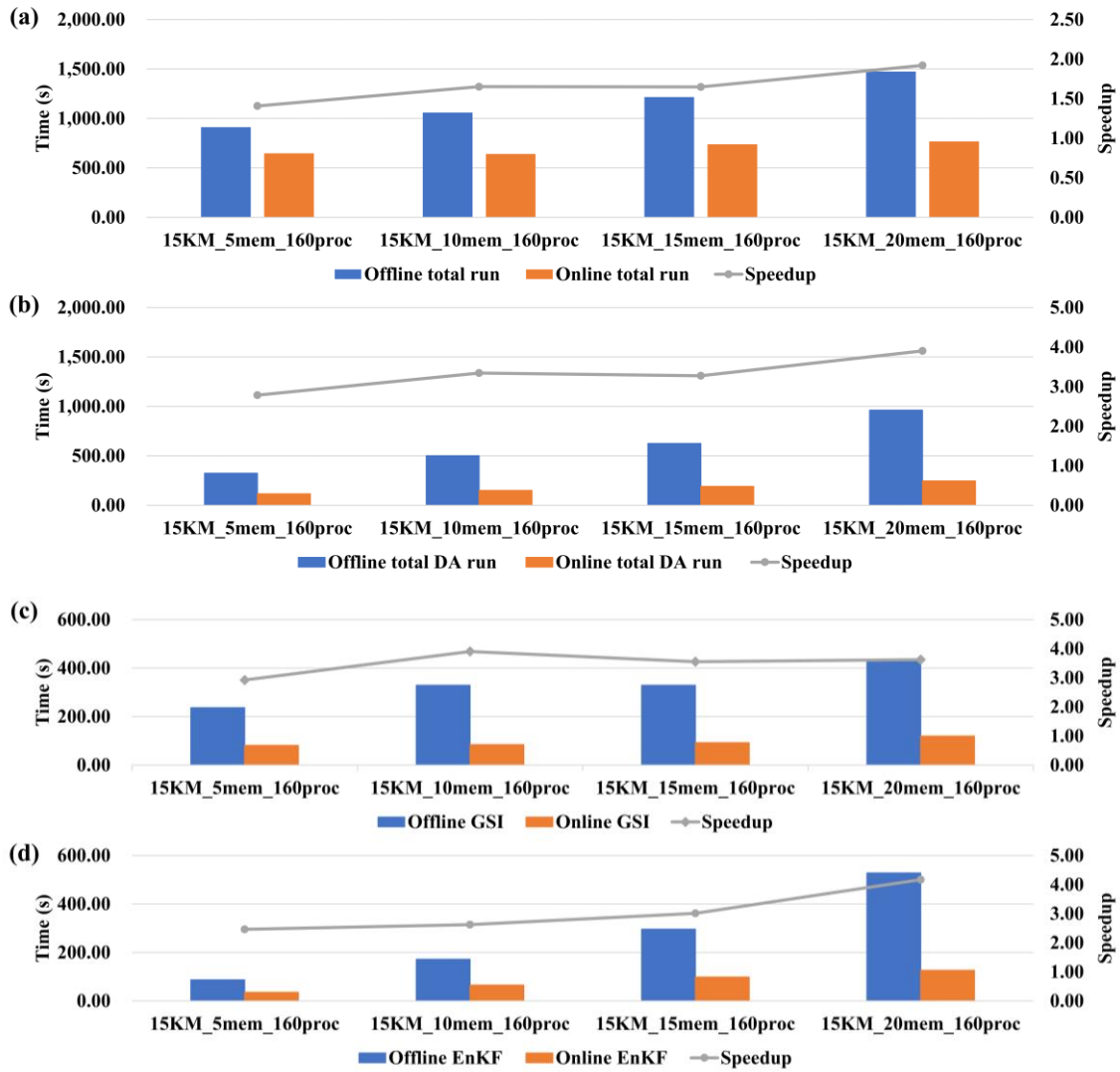



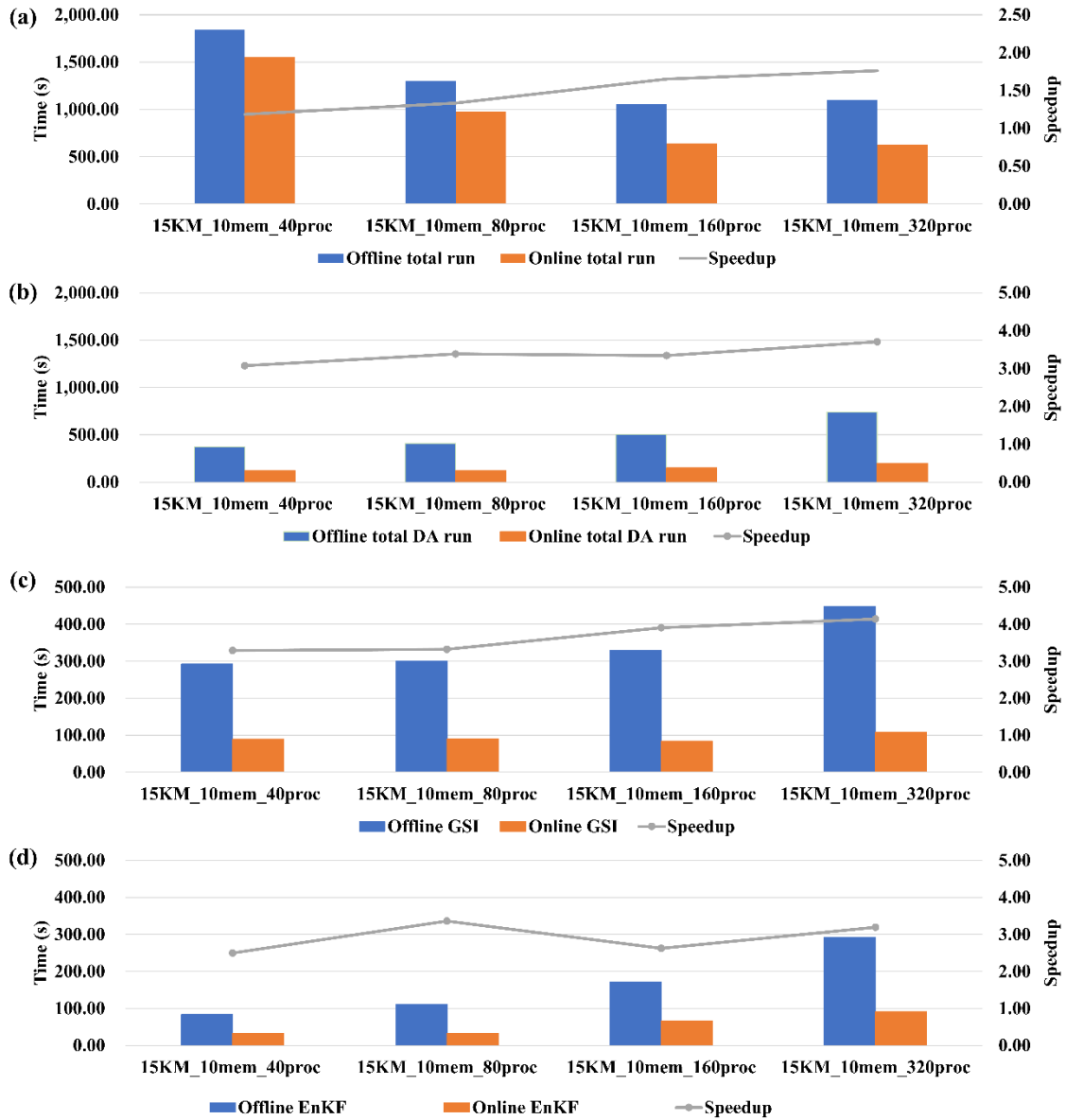
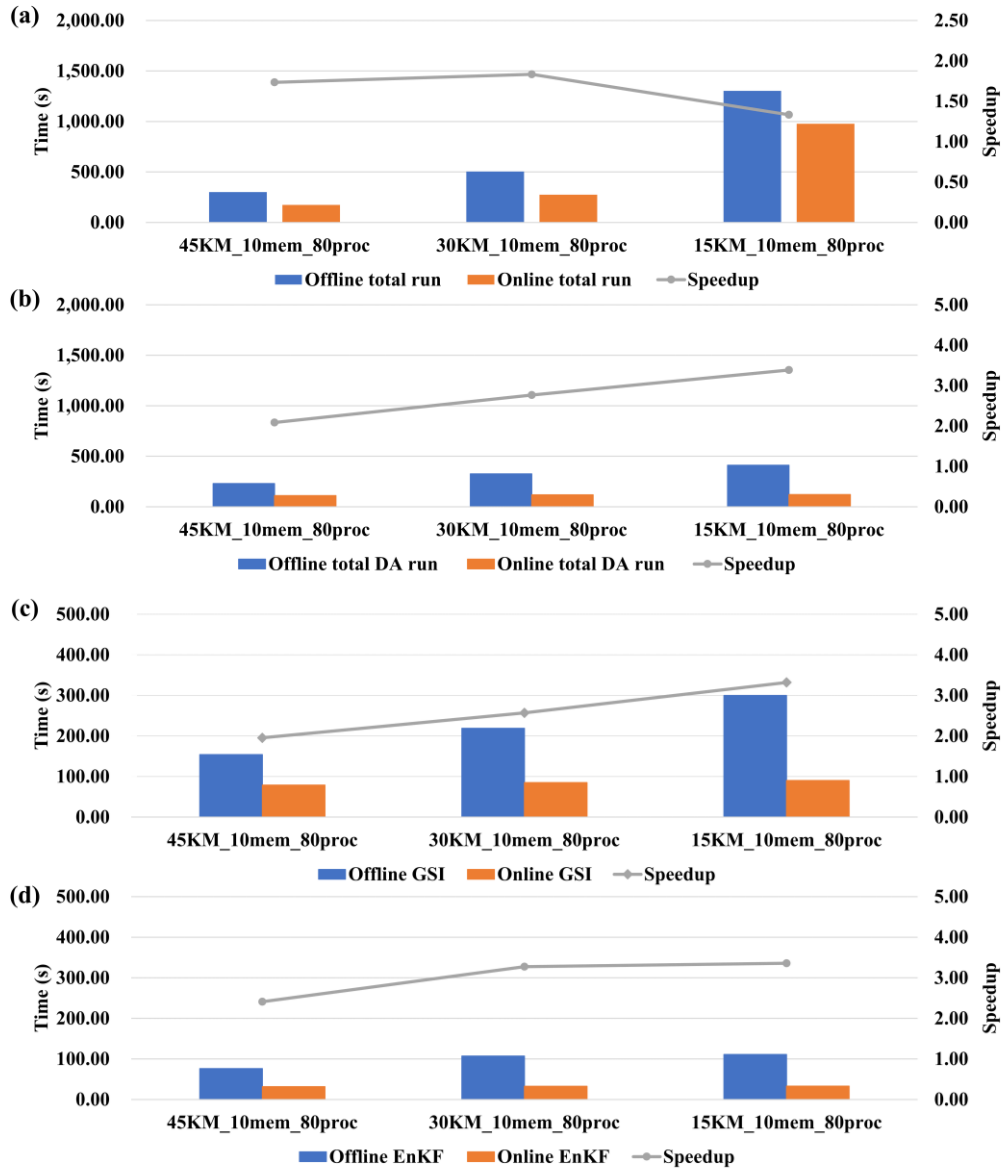
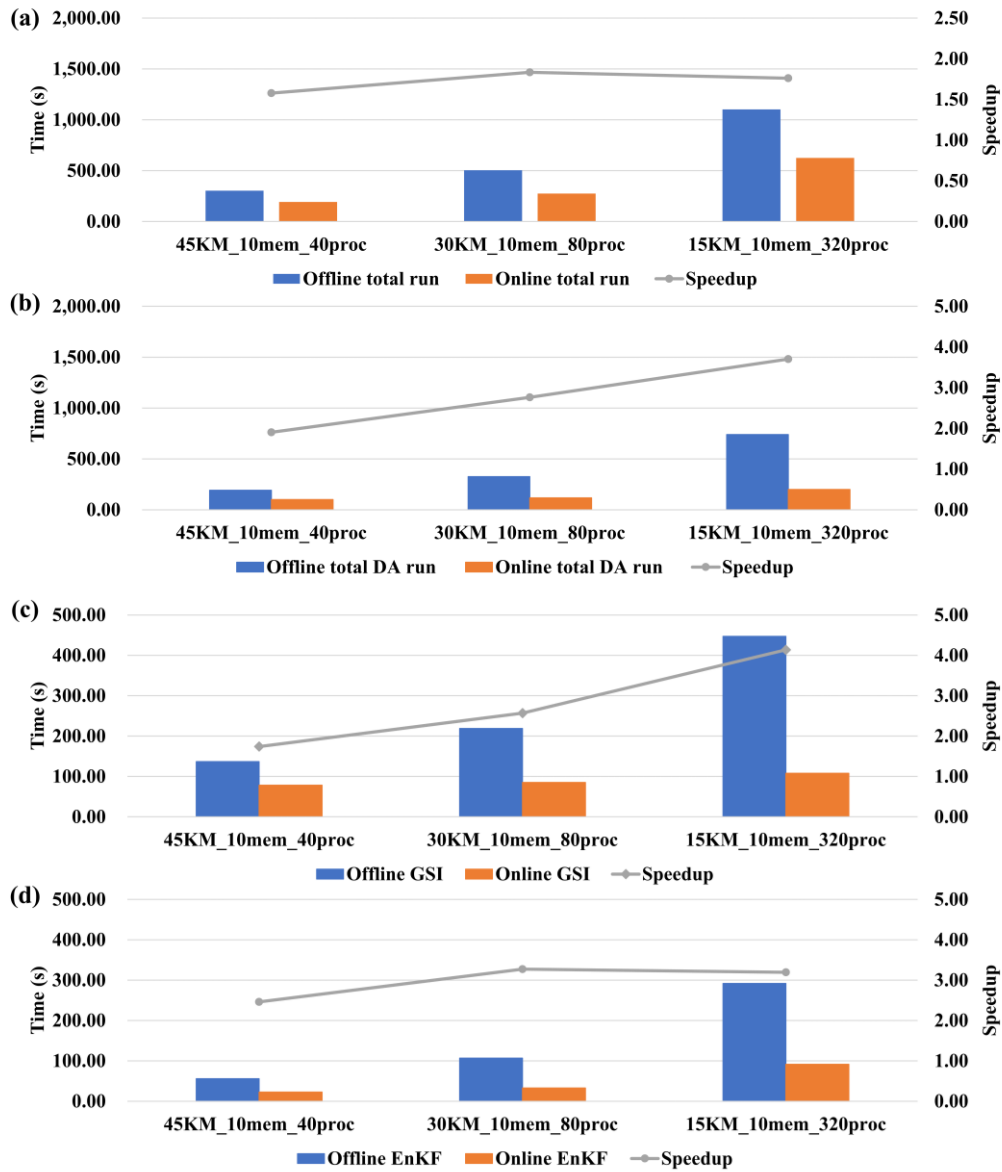
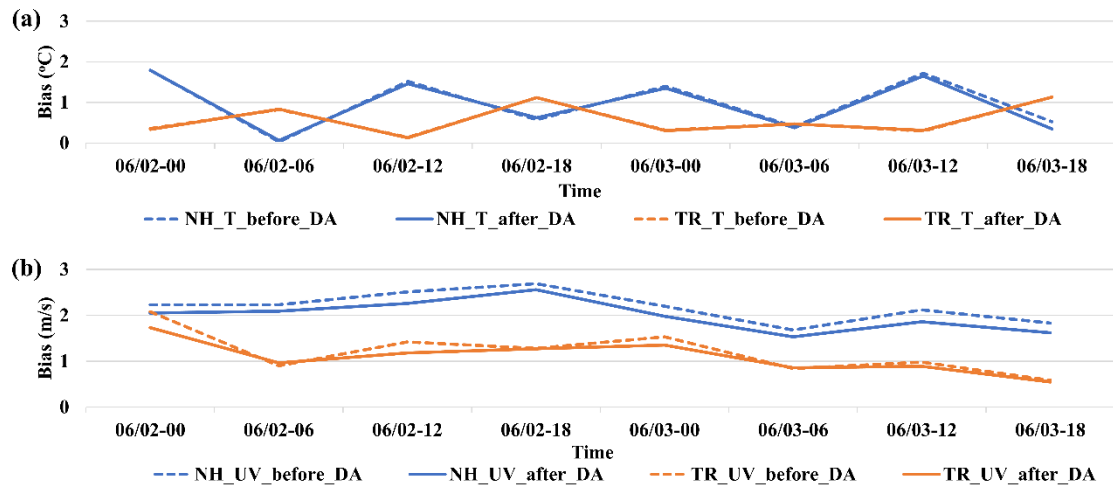
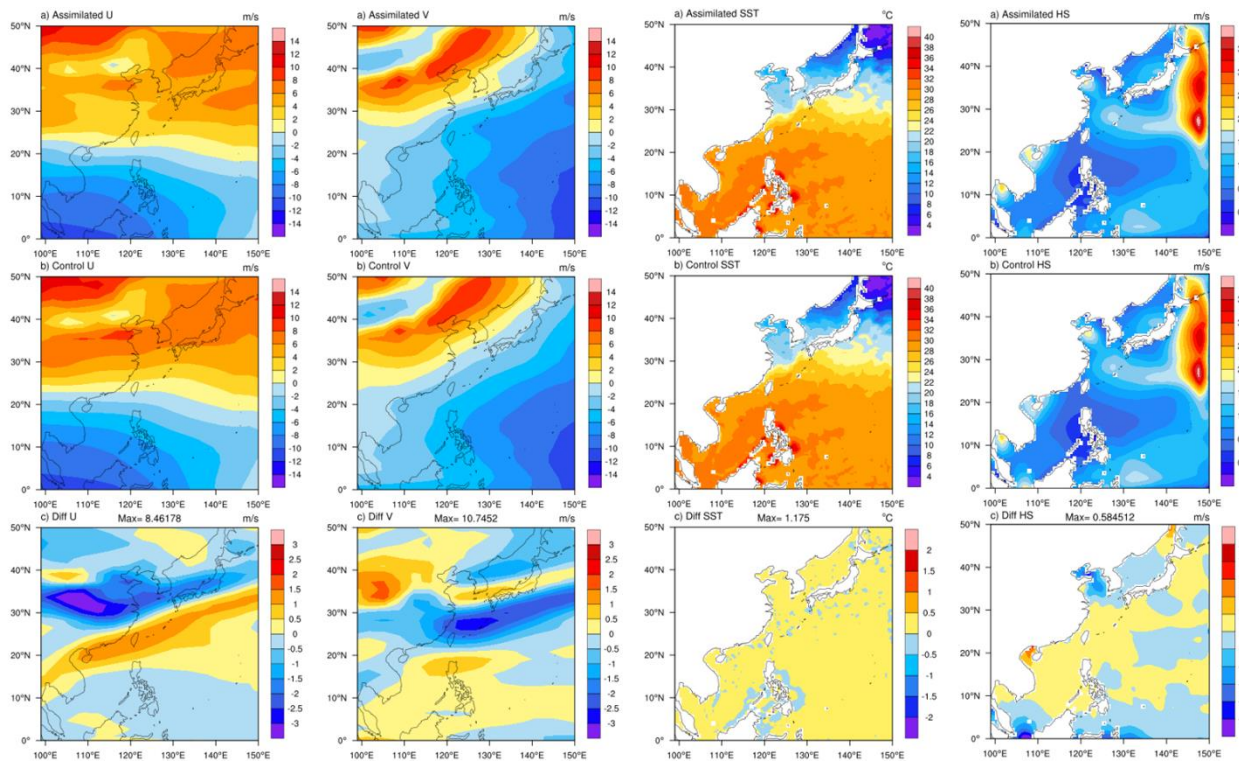
Figure 13. Execution time (colored bars) corresponding to the online and offline WRF-GSI/EnKF and the corresponding speedup (gray line, ratio of offline execution time to online execution time) from experiment set 1 in Table 2. (a) Total run (including model run and DA algorithms run). (b) DA algorithms (including GSI and EnKF) run. (c) GSI run. (d) EnKF run.

795 **Figure 14. As in Fig. 13, but from experiment set 2 in Table 2.**

800   **Figure 15. As in Fig. 13, but from experiment set 3 in Table 2.**

805 **Figure 16. As in Fig. 13, but from experiment set 4 in Table 2.**

810   **Figure 17. Total bias of assimilated variables relative to corresponding observations before and after DA for (a) T and (b) UV at each DA time from the EnKF standard output file. The dotted lines indicate the bias of assimilated variables before DA and the solid lines indicate the bias of assimilated variables after DA. Blue lines are the bias in the area of 0°–25° N, and orange lines are the bias in the area of 25°–50° N.**

815

Figure 18. Simulation results of FIO-AOW (at 06:00 on June 3rd, 2016) about the fields of meridional wind (U; the first column) and zonal wind (V; the second column) produced by WRF, sea surface temperature (SST; the third column) produced by POM, and sea surface significant wave height (HS; the fourth column) produced by MASNUM. The first row shows the results of the full-example-DA-system predicted since 00:00 on June 3rd, based on the DA experimental setup in Section 5.1, the second row shows the results without DA and the third row is the corresponding differences.

820

Table 1. Horizontal resolutions and time steps of WRF.

Horizontal Resolution	Total Horizontal Grid Points	Time Step
45 km	160×120	180 s
30 km	240×180	120 s
15 km	480×360	60 s

830 **Table 2. Setup of four experiment sets in terms of horizontal resolution, number of ensemble members and number of processes**

Experiment set	Horizontal resolution	Number of ensemble members	Processes for each ensemble member	Label marks
Set 1	15 km	5	160	15KM_5mem_160proc
		10		15KM_10mem_160proc
		15		15KM_15mem_160proc
		20		15KM_20mem_160proc
Set 2	15 km	10	40	15KM_10mem_40proc
			80	15KM_10mem_80proc
			160	15KM_10mem_160proc
			320	15KM_10mem_320proc
Set 3	45 km	10	80	45KM_10mem_80proc
	30 km			30KM_10mem_80proc
	15 km			15KM_10mem_80proc
Set 4	45 km	10	40	45KM_10mem_40proc
	30 km		80	30KM_10mem_80proc
	15 km		320	15KM_10mem_320proc

Table 3. I/O access statistics corresponding to WRF-GSI/EnKF

Horizontal resolution	Number of ensemble members	Number of observation prior files	Total I/O accesses to observation priors	Number of model ensemble background & analysis files	Total I/O accesses to model ensemble background & analysis files
15 km	5	12	0.11 GB	324	129.13 GB
15 km	10	22	0.21 GB	624	251.30 GB
15 km	15	32	0.30 GB	924	373.48 GB
15 km	20	42	0.39 GB	1224	495.65 GB
30 km	10	22	0.18 GB	624	62.86 GB
45 km	10	22	0.17 GB	624	27.96 GB