**A sample DA system based on DAFCC1**

**User’s Guide**

**Chao Sun, Li Liu, Ruizhe Li**

sunchaoo@tsinghua.edu.cn

[liuli-cess@tsinghua.edu.cn](mailto:liuli-cess@tsinghua.edu.cn)

Ministry of Education Key Laboratory for Earth System Modeling,

Department for Earth System Science (DESS),

Tsinghua University, Beijing, China

14th, October, 2022

## Get the source code and prepare the running environment

## Download source code and unzip

Users can download the sample ensemble DA system code from the github address mentioned in the manuscript. After downloading, users can get the compressed package “*DEMO\_DA\_System\_DAFCC1.zip*” and run command “*unzip DEMO\_DA\_System\_DAFCC1.zip*”.

The unzipped directory includes the directory of the C-Coupler platform (*model\_platform*), the working directory of the sample ensemble DA system (*atm\_ocn\_da\_demo*), the source code directory of the toy models (*models*) and the toy DA algorithms (*da\_algorithm\_demo*), and the input data directory (*inputdata*).

The *model\_platform* contains the configuration information of the platform that is not required to modify for users. The *inputdata* contains several sub directories to respectively store the input data for component models and DA algorithms. As the data is too large to upload, please contact the author to obtain it, or use the provided script (*ocn\_create\_netcdf\_file\_eff\_nvars.ncl*, *create\_netcdf\_file\_utils.ncl*) to generate it yourself. The *atm\_ocn\_da\_demo* is the working directories, where *configure.rb*, *compile.rb* and *submit.rb* are supported for users to configure, compile and run the sample ensemble DA system, and there are also some scripts for data preparation and result analysis.

## Setup of the sample DA system

The setup of the sample DA system includes the setup of the sample coupled model and the setup of the sample DA algorithms.

## Set up of the sample coupled model

## Configure the sample coupled model

Users can modify the file “atm\_ocn\_da\_demo/config/case.xml” to make model related settings, such as number of ensemble (*ensemble\_size*), parallelism (*num\_total\_procs*), grid resolution *(grid\_info*), etc, as shown in Fig. 1.



Figure 1. atm\_ocn\_da\_demo/config/case.xml.

Before configure the sample coupled model, users need to specify the source code and input data location of the sample coupled model including the two toy component models and the coupler C-Coupler3. Users can modify the file “atm\_ocn\_da\_demo/config/models/atm/atm\_demo/model.xml”, as shown in Fig. 2, and “atm\_ocn\_da\_demo/config/models/ocn/ocn\_demo/model.xml” to specify the source code (*model\_dir*) and input data (*data\_dir*) of the toy component model. Users can modify the file “atm\_ocn\_da\_demo/config/libs/c-coupler/lib.xml” to specify the source code (*library\_dir*) of the coupler C-Coupler3.



Figure 2. atm\_ocn\_da\_demo/config/models/atm/atm\_demo/model.xml.

After finish the model settings, users should run command “*./configure.rb*” to configure the sample coupled model. Two new directories *run* and *logs* are generated , while directory *run* is the working directory to store the compilation files and running results and directory *logs* is used to store the logs of configuration, compilation, and run according to the generation time. If the user modifies parameters in the file “atm\_ocn\_da\_demo/config/case.xml”, the command “*./configure.rb*” needs to be executed before run the sample DA system.

## Compile the sample coupled model

To compile the sample coupled model, users need to first prepare the common compile environment by modifying the file *config/machines/{machine\_name}/common\_compiler.cfg* and *config/machines/{machine\_name}/submit*, where the *{machine\_name}* is defined in the file “atm\_ocn\_da\_demo/config/case.xml” by the key words *machine.* Fig.3 shows an example of “*common\_compiler.cfg*”, where the compiler, compilation options, and some libraries such as NETCDF and MPI of current machine, are specified here. File “*submit*” specifies how the experiment is submitted to current machine. The user needs to modify the path information (*CASE\_ROOT*) and submit script in this file.

Then, users should run the command “*./compile.rb*” to compile the sample coupled model. The executable file generated by the user is stored in a directory *run/atm/atm\_demo/{ensemble\_index}/exe/*, *run/ocn/ocn\_demo/{ensemble\_index}/exe/*, and *run/libs/c-coupler/cd3c1c1d44f9340d6b8e9f6b6e1fd6d3c2dce75a/exe*/, where *{ensemble\_index}* is the working directory of each ensemble member. The log files of the compilation are stored under the directory *logs*. The operation of compiling the sample coupled model only needs to be performed at the first time or when the source code or the compiler are modified.

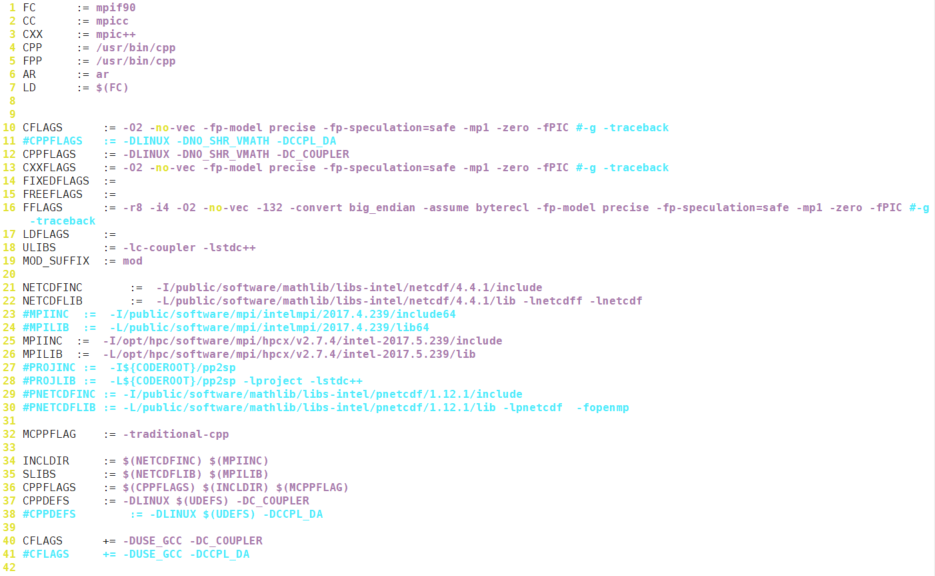


Figure 3. An example of “*common\_compiler.cfg*”

## Set up of the sample DA algorithms

## Configure the sample DA algorithms

The configuration of the sample DA algorithms includes the preparation of the input data and the modification of the namelist. A script (“atm\_ocn\_da\_demo/prepare\_da\_demo\_run.sh”) is supported to users to make these configurations. The user needs to modify the path information (*ROOT*) in the script, and needs to modify the corresponding number of ensemble (*ensemble\_num*) and grid resolution settings (*ocn\_grid\_info/atm\_grid\_info*) according to the settings of the “atm\_ocn\_da\_demo/config/case .xml”. If the user modifies these two parameters, the command “*./prepare\_da\_demo\_run.sh*” needs to be executed before run the sample DA system.

## Compile the sample DA algorithms

To compile the sample DA algorithms, users need to go to the source code directory (da\_algorithm\_demo). A configuration file (*Makefile.conf*) is supported to users to specify the compiler and compilation options. A script “*build.sh*” is supported to users to compile the sample DA algorithms, where some path information about the source codes (*ROOT\_DIR*) and libraries (*NETCDFPATH/MPIPATH*) need to be modified before executing this script. A script “*clean.sh*” is also supported to users to remove the files produced by the compilation.

Users should run command “./*build.sh*” to compile the sample DA algorithms. The dynamic link libraries of the sample DA algorithms will be generated when they are compiled successfully and will be automatically placed in the directory “atm\_ocn\_da\_demo/CCPL\_dir/libs/external\_procedures/”. The operation of compiling the sample DA algorithms only needs to be performed at the first time or when the DA algorithms code or the compiler are modified.

## Run the sample DA system

After finishing the setup of the sample coupled model and the sample DA algorithms, users should run the command “*./submit.rb*” under the working directory to run the sample DA system. Note that, “*./submit.rb*” is common to any kind of simulations on any kind of hardware platforms.

## Evaluation of the sample DA system

The log file of sample DA system run is stored under the directory *logs* according to its configuration and execution time (“logs/{configure\_time}/ execute.{submit\_time}”) by default or stored in the file specified by the user in the submit script. The log file allows the user to tell if the experiment was completed correctly.

A script (“atm\_ocn\_da\_demo/collect\_result.sh”) that collects the resulting data to the directory “atm\_ocn\_da\_demo/results” and counts the running time of each stage to the file “atm\_ocn\_da\_demo/{ocn\_grid\_info}km.results”, is provided to the user. The user needs to modify the path information (*ROOT*) in the script, and needs to modify the corresponding grid resolution settings (*ocn\_grid\_info, atm\_grid\_info*) and the parallelism (*atm\_proc,* *ocn\_proc*) according to the settings of the “atm\_ocn\_da\_demo/config/case .xml”. The user can obtain the running time statistics of each stage by executing commands “*./collect\_result.sh*”. Fig.4 shows an example of “atm\_ocn\_da\_demo/20km.results”, where “*CCPL\_initial*” and “*CCPL\_run*” give the statistical results of initialization and data exchange time of three toy DA algorithms respectively.

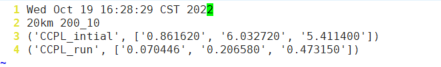


Figure 4. An example of “*20km.results*”