Supplement of Geosci. Model Dev., 11, 3557–3586, 2018
https://doi.org/10.5194/gmd-11-3557-2018-supplement

Geoscientific Model Development
Open Access
EGU

Supplement of

C-Coupler2: a flexible and user-friendly community coupler for model coupling and nesting

Li Liu et al.

Correspondence to: Li Liu (liuli-cess@tsinghua.edu.cn) and Cheng Zhang (zhangc-cess@tsinghua.edu.cn)

This supplement includes two parts. The first part shows examples of implementing a coupled model with C-Coupler2 APIs, and the second part shows examples for the adaptive restart capability.

1 Examples of implementing a coupled model with C-Coupler2 APIs

Figure S1 shows an example of the use of C-Coupler2 APIs to achieve hybrid coupling configuration and model coupling during the initialization stage of a coupled model with four component models (*comp1*–*comp4*). We assume that *comp1* and *comp2* are coupled together, *comp3* and *comp4* are coupled together, and that *comp3* and *comp4* are the children of *comp1* and depend on some boundary conditions from *comp1*. First, *comp1* and *comp2*, which cover all MPI processes (processes 0–34) and do not share any MPI process, simultaneously call the API *CCP_register_component* to register themselves as the root component models. The child component models *comp3* and *comp4* partially share a subset of MPI processes (processes 9–12). All MPI processes of *comp3* first register *comp3* as a child of *comp1*, and next set the unique time step, register several model grids, register a parallel decomposition, register several coupling field instances, specify a coupling field instance as the dynamic surface field of a 3-D grid, define several timers, and register several coupling interfaces. After calling the API *CCPL_do_individual_coupling_generation* for coupling procedure generation within *comp3* itself, *comp3* executes some coupling interfaces, and then finalizes its coupling configuration stage through calling the API *CCPL_end_coupling_configuration*. *Comp4* follows a C-Coupler2 flowchart similar to *comp3*. As *comp3* and *comp4* share some processes, they cannot conduct coupling configuration and model coupling at the same time in most cases (in this example, we specify *comp3* to run before *comp4*), except for the simultaneous calling of the API *CCPL_do_external_coupling_generation* that can generate coupling procedures for the coupling connections between the two child models. After both child models have finished their coupling initialization stage, their parent conducts its coupling configuration, following a similar flowchart. As *comp1* shares processes with its children, *comp1* cannot conduct coupling registration simultaneously with *comp3* and *comp4*, and thus *comp1* runs after its children here. As *comp2* does not share any process with the other component models, it can conduct coupling registration simultaneously with *comp1*, *comp3*, and *comp4*. Finally, *comp1* and *comp2*, the root component models, simultaneously call the API *CCPL_end_coupling_configuration* to finalize the coupling configuration stage of themselves and the whole coupled model and to invoke global coupling procedure generation. At the end of the initialization stage, each component model can read in the restart data files when necessary.

Figure S2 shows an example of model coupling in the kernel (time integration) stage of the coupled model in Fig. S1. In addition to the assumptions in Fig. S1, *comp1* and *comp2* are further assumed to have the same time step, which is double that of *comp3* and *comp4*. All coupling interfaces are executed here without bypassing the timers. At a time step of *comp1* and *comp2*, they can simultaneously execute coupling interfaces, call the API *CCPL_do_restart_write_IO* to generate restart data files when the restart timer is bypassed or is on, and finally call the API *CCPL_advance_time* to advance the model time managed by C-Coupler2. We strongly recommend checking the consistency of model time between a component model and

C-Coupler through calling the API *CCPL_check_current_time*. *Comp3* and *comp4* alternately use a C-Coupler2 flowchart similar to that for *comp1* and *comp2*, but they will advance their model time twice when *comp1* and *comp2* advance their model time once.

2 Examples for adaptive restart capability

- **Example 1**

We'd like to set the first example based on the coupled model setting in Fig. S3(a), where the coupling connection from the component model *comp1* to *comp2* has a lag of 600 s. Given that the whole coupled model has already produced restart data files corresponding to the model time of 600 s in a previous run (corresponding to the red words of *"do restart write"* in the second iteration in Fig. S3(a)), in a restart run of the coupled model restarted from the model time of 600 s (Fig. S3(b)), after the both component models read in the corresponding restart data (corresponding to the red words of *"do restart read"* in the second iteration in Fig. S3(b)) and next advance the model time, *comp2* will enter the third iteration with the model time of 1200 s and the coupling interface *imp2_3* will import the coupling field instance values exported by the coupling interface *exp1_2* of *comp1* at its model time of 600 s. However, *comp1* will also enter the third iteration with the model time of 1200 s and will never execute *exp1_2* again. Therefore, besides the values imported by *imp2_2* (these values may be used by *comp2* before executing *imp2_3*), the values imported by *imp2_3* should also be included in the restart data files corresponding to the model time of 600 s. This example indicates that the restart data files corresponding to a restart write model time should include the coupling field instance values at different model time corresponding to a positive lag on a coupling connection.

- **Example 2**

We'd like to set the second example based on the coupled model setting in Fig. S3(c). There are no coupling lags on the coupling connections between the two component models, and the only difference between the two component models is that, they have different orders for writing restart data files and advancing model time in each iteration of the main loop. Given that the whole coupled model should prepare restart data corresponding to the model time of 600 s, the component model *comp1* should produce restart data files at the second iteration, while *comp2* should produce restart data files at the first iteration because its model time has already been advanced to 600 s. In a restart run of the coupled model restarted from the model time of 600 s (Fig. S3(d)), after the both component models read in the corresponding restart data (corresponding to the red words of *"do restart read"* in the second iteration of *comp1* and in the first iteration of *comp2* in Fig. S3(d)), *comp2* will enter the second iteration with the model time of 600 s and it coupling interface *imp2_2* will import the coupling field instance values exported by the coupling interface *exp1_2* of *comp1* at its model time of 600 s. However, *comp1* will enter the third iteration with the model time of 1200 s and will never execute *exp1_2* again. Therefore, besides the values imported by *imp2_1* (these values may be used by *comp2* before executing *imp2_2*), the values imported by *imp2_2* should also be included in the restart data files corresponding to the model time of 600 s. This example indicates that the restart data

files corresponding to a restart write model time may need to include the coupling field instance values at different model time even when there is no lag on any coupling connection.

- **Example 3**

As shown in Fig. S3(d), *comp2* will execute the coupling interface *exp2_2* after the coupled model run is restarted. *exp2_2* will try to export the coupling field instance values to the coupling interface *imp1_2* of *comp1*, however, *imp1_2* will never be executed again in the restart run. Therefore, the data export by *exp2_2* should be bypassed, to avoid deadlocks. Similarly, regarding the third example corresponding to Fig. S3(e) with a coupling lag of -600 s from *comp1* to *comp2*, in the restart run corresponding to Fig. S3(f), the data export by the coupling interface *exp1_3* of *comp1* should also be bypassed. These examples indicate that it may need to bypass the data export of some export interfaces at some model time after restarting the coupled model run.

- **Example 4**

Fig. S3(g) shows an example about the implementation of the adaptive restart capability based on the coupled model setting in Fig. S3(a), where the red operations are restart write related. Given that the unique restart timer will be on at the model time of 600 s, for each of *comp1* and *comp2*, the API *CCPL_do_restart_write_IO* will write the coupling field instance values imported at the model time of 600 s (corresponding to *imp1_2* and *imp2_2*) into the corresponding NetCDF restart data file and mark 600 s as the restart writing model time. At the third iteration with the model time of 1200 s, *imp2_3* of comp2 will obtain the coupling field instance values that are exported by *exp1_2* of *comp1* at its model time of 600 s. As 600 s is the same with but not later than the restart writing model time, according to the step 4 of the implementation, *imp2_3* will write its obtained coupling field instance values into the corresponding NetCDF restart data file. According to the above step 5 of the implementation, *comp1* will produce the corresponding binary restart data file when advancing the model time at the third iteration, while *comp2* will produce the corresponding binary restart data file when advancing the model time at the fourth iteration. Fig. S3(h) shows a restart run of the coupled model restarted from the model time of 600 s, where red operations are restart read related. Both *comp1* and *comp2* should call the API *CCPL_start_restart_read_IO*, which will read in the restart management information and set the restarted model time to 600 s. After entering the main loop, *comp2* will first execute the third iteration (with the model time of 1200 s), where the coupling field instance values obtained at the second iteration (with the model time of 600 s) may be used and can be recovered through calling the API *CCPL_restart_read_fields_all* or *CCPL_restart_read_fields_interface* in advance. As the coupling field instance values that will be obtained by *imp2_3* of *comp2* at its model time of 1200 s are exported by *comp1* at its model time of 600 s, and 600 s is the same with but not later than the restarted model time, according to the step 8 of the implementation, C-Coupler2 will automatically read in the corresponding coupling field instance values from the corresponding NetCDF restart data file automatically. Similar examples can be derived from Fig. S3(c) ~ S3(f).

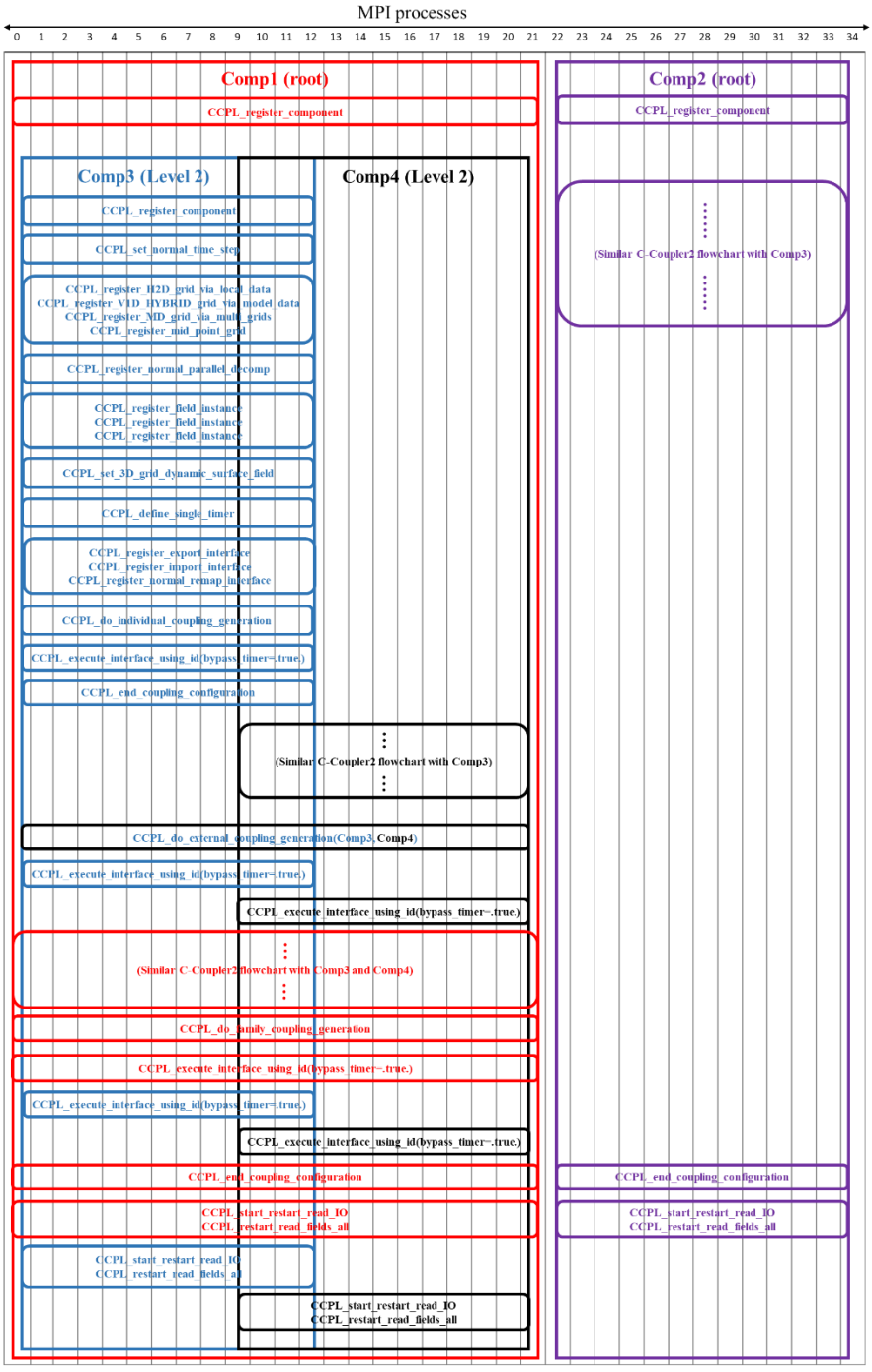


Figure S1 An example of hybrid coupling configuration and model coupling in the initialization stage of a coupled model constructed with C-Coupler2. *Comp1–comp4* are the four component models. Labels and boxes of the same color correspond to the same component model.

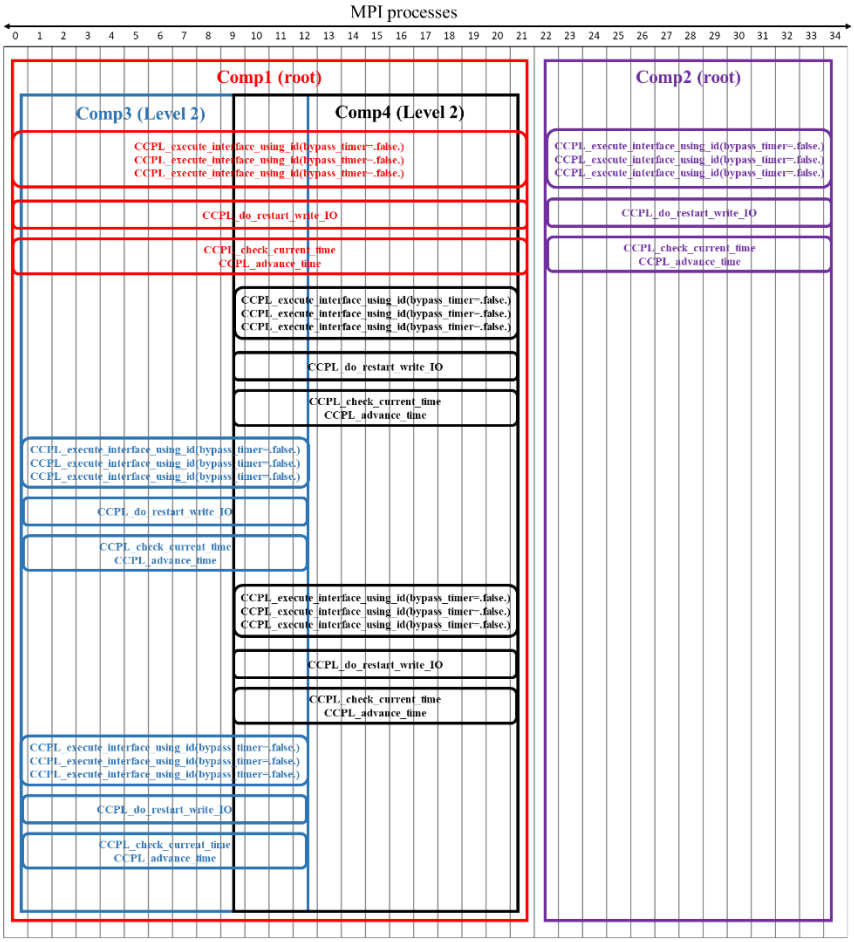
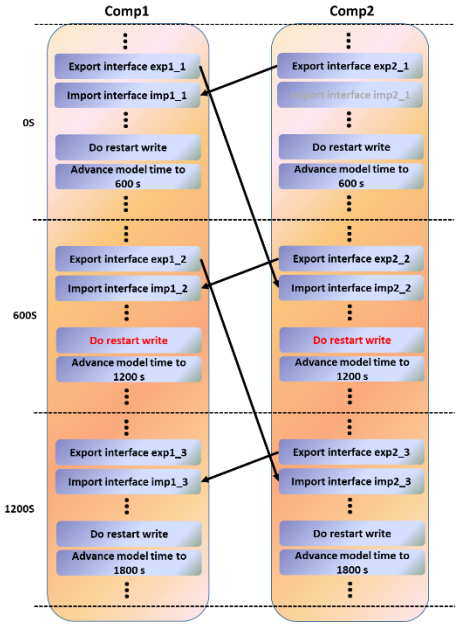
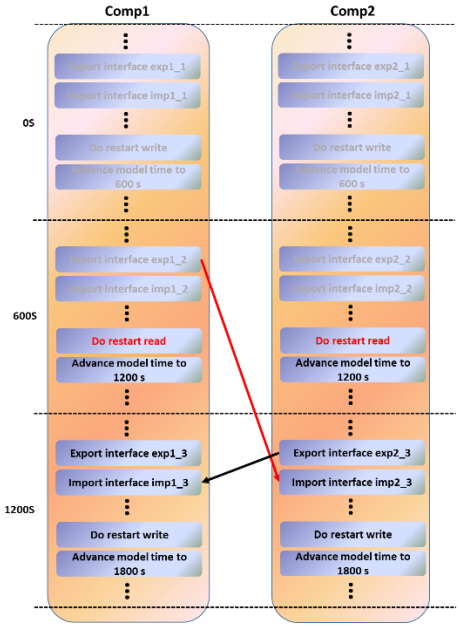


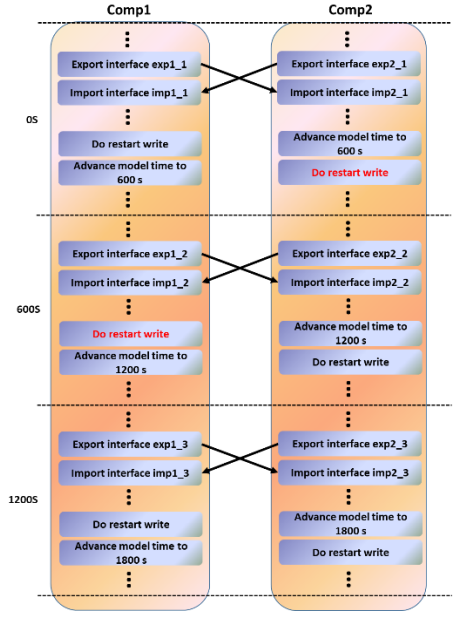
Figure S2 An example of model coupling in the kernel (time integration) stage of a coupled model constructed with C-Coupler2. *Comp1–comp4* are the four component models. Labels and boxes of the same color correspond to the same component model.



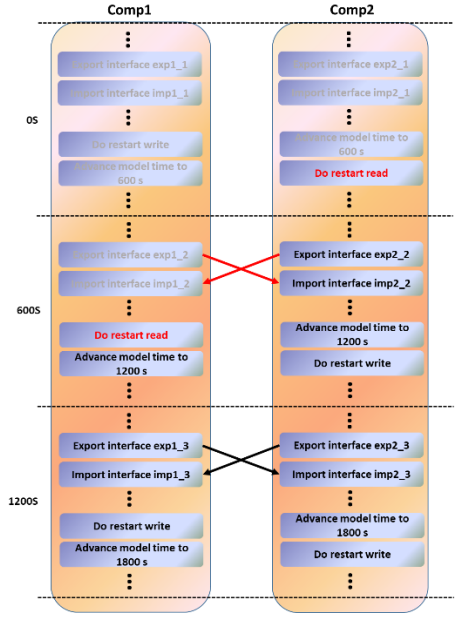
(a) An initial run of a two-way coupled model with a lag of 600 s from *comp1* to *comp2*



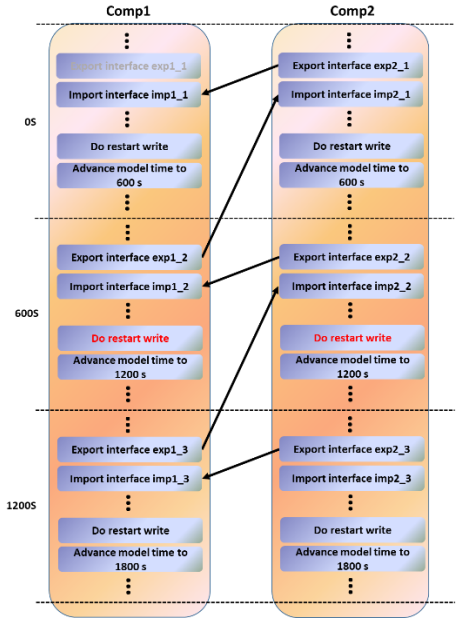
(b) A restart run of the coupled model in Figure (a) restarted from the model time of 600 s



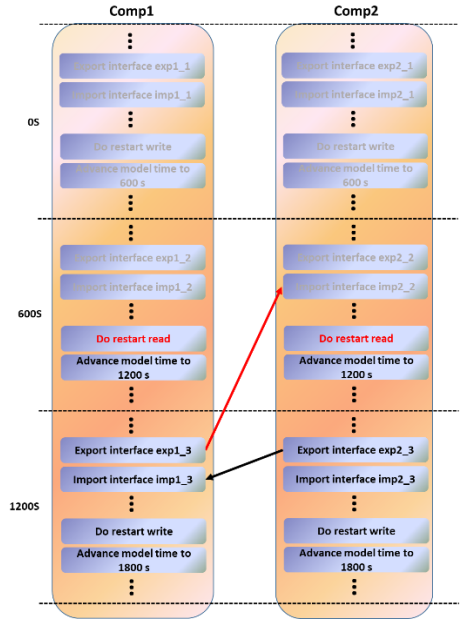
(c) An initial run of a two-way coupled model without coupling lags.



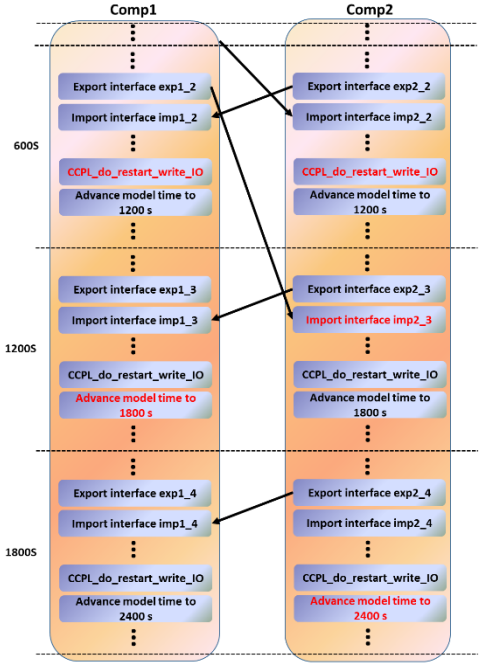
(d) A restart run of the coupled model in Figure (c) restarted from the model time of 600 s



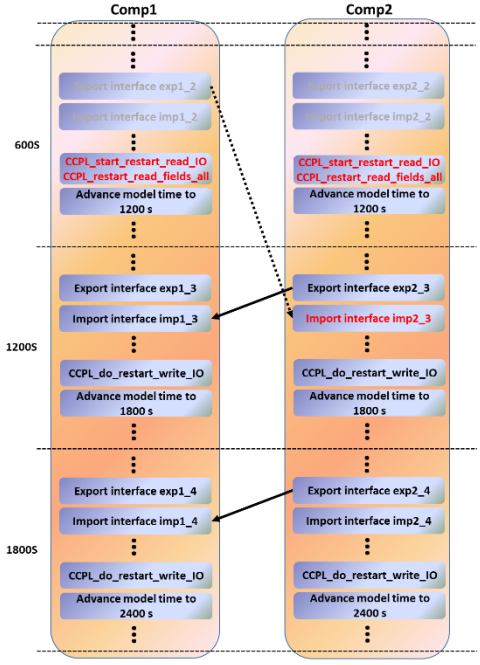
(e) An initial run of a two-way coupled model with a lag of -600 s from *comp1* to *comp2*



(f) A restart run of the coupled model in Figure (e) restarted from the model time of 600 s



(g) An initial run corresponding to Figure (a). The first iteration is not shown. The red operations are restart write related



(h) A restart run of the coupled model in Figure (g) restarted from the model time of 600 s. The red operations are restart read related

Figure S3 Sample restart requirements under different coupling lag settings. The grey words indicate that the corresponding operations will be bypassed or not executed.