Review 1:

1. For the implementations described in Section 2, the implemented interface is specifically developed for WRF v4.6.0. If so, are there any plans to maintain this interface for higher or lower versions of WRF? Would it be easy to implement this interface for other WRF versions?

As explained in the paper (lines 69-71), this coupling interface is an update of the coupling interface already available in WRF since 2014. All the changes made to the code are available on GitHub. They are listed here https://github.com/massonseb/WRF/commits/GMD_wrf_coupling/?author=massonseb

As the interface is non-intrusive, it should not necessitate work to maintain it in past/future WRF versions. Apart from the changes we have made for wave coupling, our update of the coupling interface concerns module_cpl.F, the replacement of 3 calls to cpl_rcv by a single call to cpl_rcv_sfcdrv in module_surface_driver.F and the use of the variables cosa and sina in module_first_rk_step_part1.F and module_surface_driver.F. Backporting these changes to older versions of WRF will therefore be fairly straightforward. We are currently preparing a "merge request" on GitHub to integrate this update to the coupling interface into the official WRF repository, which would ensure its use in future versions.

We added the following sentence at the end of the paragraph, line 72:

Note that all changes made to the code are available on github at the following address https://github.com/massonseb/WRF/commits/GMD_wrf_coupling/?author=massonseb. They are limited to a few routines, so porting them to older versions of WRF will be fairly straightforward.

2. Line 155 is confusing to me. Do the authors separate the WRF code into three subroutines (init, run, finalize)? Or do the authors put functions in "frame/module_cpl_oasis3.F" used in these processes?

We did not separate the WRF code into 3 subroutines. The coupling interface provides (in "frame/module_cpl_oasis3.F") subroutines that will be used in the native WRF processes how it is illustrated in Fig. 4. We added *"provided in the coupling interface"* in the following sentence line 155:

The coupling sequence (Fig. 2) is structured into 3 steps with functionalities corresponding to specific Fortran subroutines provided in the coupling interface and callable from the original code

3. Are the authors using the existing WRF I/O streams (auxinput or auxhist) when getting the input or output? Why don't use the existing ones?

This coupling interface does not interfere with existing WRF I/O streams (auxinput or auxhist). Auxinput and auxhist files are used the same way with or without activating the coupling interface at the compilation stage (see section 2.5). This interface is also compatible with WRF IO quilting (see lines 258-259) and we used WRF IO quilting in several of our coupled simulations. We also clarified the use of WRF IO quilting in the new section 2.6 (lines 382-393).

Having said that and regardless of the coupling interface, in our forced or coupled simulations, we use WRF auxinput stream, but we don't generally use the auxhist stream. Instead, we use the XIOS I/O server (https://forge.ipsl.jussieu.fr/ioserver/wiki) to produce the output files. Our main reason for using XIOS is that it can calculate the time average/max/average/std of any variable declared in the Registry without changing anything in the code. We find it much more convenient than modifying the WRF subroutine clwrf_output_calc in phys/module_diag_cl.F. We find XIOS is much more flexible than the default WRF output stream. We also think XIOS is more efficient than WRF IO quilting. The use of the XIOS-WRF is not related to the coupling interface described in this paper, and we don't think we should mention it.

4. Line 236, when in the coupled experiment. I feel it is challenging to use WRF IO quilting when using the same processors for both Ocean and Atmosphere models. How should the ocean model set these processors for IO quilting? Have the authors tested this?

In this question, does the term "processor" refer to a CPU made up of several cores or to an MPI process?
In this coupling interface, the ocean and atmosphere models run on different and dedicated MPI processes and not sequentially on the same MPI process.
We added a new section "2.6 Running the coupled model" to detail and clarify how to run and allocate MPI resources to the coupled model (lines 382-393):
"To achieve higher parallelism and keep the different models as independent as possible, each model has its own executable with its dedicated MPI resources. All the executables run in parallel and share the same MPI world using the "multiple programs, multiple data" (MPMD) launch mode. WRF can be coupled to one or several external models. WRF and the external models can include one or several domains (e.g. embedded zooms, see section 4). For example: if X is the number MPI tasks allocated to WRF and Y the number of MPI tasks allocated to the external model to which WRF is coupled (e.g. an ocean model), the number of MPI tasks that must be allocated to run the simulation is X + Y. Then, if using WRF IO quilting, the X WRF MPI tasks will be split among X1 "compute nodes" and X2 "server nodes" with X = X1 + X2. If WRF is configured with one nested zoom, the parent and the child domains, d01 and d02, will run sequentially on X1 MPI tasks. The same applies to the external model, which could include its own zoom domains running on Y MPI tasks. Note that the external models do not use/see WRF I/O quilting. OASIS3-MCT can couple models sharing the same executable, so we could imagine to further integrate WRF I/O quilting in the coupling interface, but this would require specific modifications of the external models and greater entanglement between the different codes, which is not our objective."

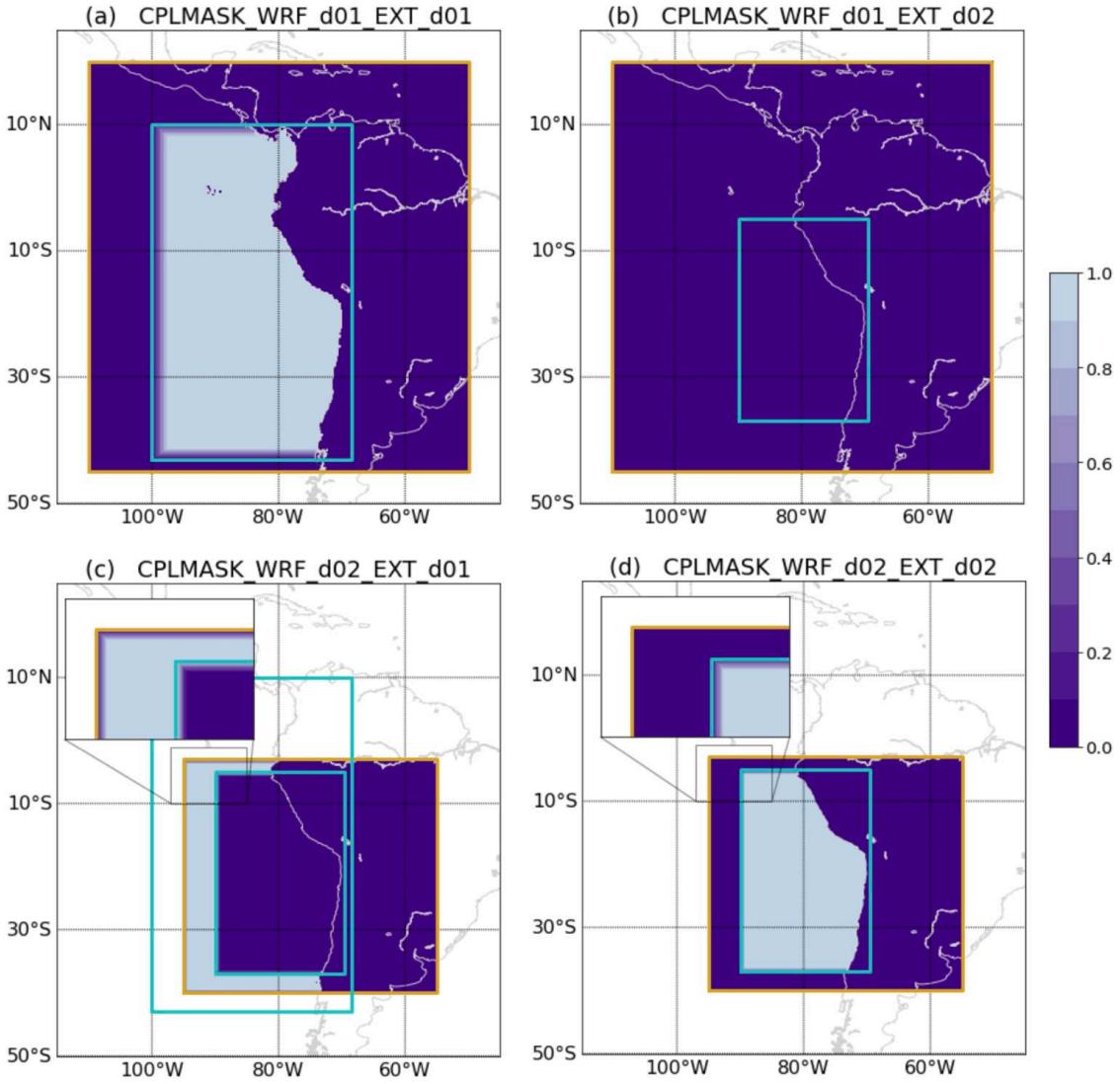
5. Section 3.3. Are the users free to add more variables for the coupling processes?

Yes definitely, this is one of the aims of this interface. We designed this interface (isolating the coupling interface in module_cpl.F and the OASIS-specific routines in module_cpl_oasis3.F) to make changes as easy as possible. Such procedure is detailed in section 3.3.

6. Section 4.1.3 is a very interesting example. Has anyone tested the nested domains in a realistic application? In addition, in Fig. 19 the filled color and the color of the boxes are very close. It would be better if the author could use a different colormap for the masks.

Yes, we have used this coupling configuration in realistic configurations (tropical channel, South-East Pacific, Gulf Stream…) with nested atmospheric and/or oceanic zooms for more than 10 years. See for example this presentation of unpublished work, https://www.clivar.org/sites/default/files/documents/wgomd/ws2014/04__highres_Masson.pdf
It is true that most existing publications using this coupling interface do not use atmospheric zooms, with the exception of:
Li, Y., Jourdain, N. C., Taschetto, A. S., Gupta, A. S., Argüeso, D., Masson, S., and Cai, W.: Resolution dependence of the simulated precipitation and diurnal cycle over the Maritime Continent, Clim Dyn, 48, 4009–4028, https://doi.org/10.1007/s00382-016-3317-y, 2017.

Figure 19 have been changed according to the reviewer's suggestions:



7. In Appendix 2, there is a typo in https://github.com/massonseb/WRF/blob/GMD_wrf_coupling/run/namecouple_example. It should be https://github.com/massonseb/WRF/blob/GMD_wrf_coupling/run/namcouple_example.

Thank you, we corrected it.