SUMMARY OF CHANGES

GIORGIO MICALETTO, IVANO BARLETTA, SILVIA MOCAVERO, IVAN FEDERICO, ITALO EPICOCO, GIORGIA VERRI, GIOVANNI COPPINI, PASQUALE SCHIANO, GIOVANNI ALOISIO, AND NADIA PINARDI

To the Topical Editors of Geoscientific Model Development

Dear Topical Editor,

The revised version of the manuscript GMD-2021-319 entitled "Parallel Implementation of the SHYFEM Model" has been edited to meet the suggestions included in the revision.

1. Review #1

We thank the reviewer for good appreciation of the manuscript.

2. Review #2

I would like to thank to the authors to address most of my concerns about the previous version of the manuscript. Here are my comments about the current version of the manuscript.
What is the portion of the redistribution timing between PETSc and model when all execution time is considered? It might be nice to add this information to the text in Sec. 3.4.

Summary of changes: The time needed for the redistribution between the PETSc and the model depends by the number of cores used for the simulation and its impact increases as the number of cores used increases ranging between 1% and 10% of the overall execution time.

I am still not convinced about the parallel implementation of the model and its reproducibility issue. It does not makes sense to me to have a numerical model with reproducibility issue. This indicates a major problem in the parallelization approach used in here. Please discuss the reproducibility issue in a more detailed way (maybe a dedicated section could be good idea) by comparing with the other models? How this issue is addressed by other unstructured ocean models (i.e. SCHISM and others) used in the literature? If other models have also same issue, please add the relevant studies and papers as reference. If not, is this specific to optimized version of PETSc and/or used solver? If issue is related with the PETSc specifically, please add reference of other applications that indicates the similar issue. It seems that DEBUGON version of the model is able to reproduce the results with very poor performance. This indicates that there is no way to run model in

parallel and to create reproducible results. In same cases, poor performance can be acceptable if the configuration can not be run in single node due to the memory limitations but this is just for very high-resolution cases that requires lots of memory.

Summary of changes: Section 5.1 was meant to include the discussion concerning the model validation and reproducibility, hence, instead of creating a new Section, we have further extended Section 5.1 to better point out the reproducibility issue and its origin. We better clarified that the round-off error comes from the floating point numbers representation and it is one of the main reason why all the numerical models, including GCMs, are affected by uncertainty of the results. We also extended the Section 5.1 citing other works regarding the assessment of models reproducibility when changing the computational environment such as compilers, HPC architectures or MPI decompositions. Namely, Cousins and Xue (2001) developed the parallel version of Princeton Ocean Model (POM) and found that there is a significant difference between the serial and parallel version of the POM concluding that the error from the data communication process via MPI is the main reason for the difference. Wang et al. (2007) studied the results of the atmospheric model SAMIL simulated with different CPUs and pointed out that the difference is chiefly caused by the round-off error. Song et al. (2012) assessed the round-off error impact due to MPI on the parallel implementation of Community Climate System Model (CCSM3). Guarino et al. (2020) presented the evaluation of the reproducibility of HadGEM3-GC3.1 model on different HPC platforms. Geyer et al. (2021) assessed the limit of reproducibility of COSMO-CLM5.0 model comparing the same code executed on different computational architectures.

Their analysis showed that the simulation results are dependent on the computational environments and the magnitude of uncertainty due to the parallel computational error could be in the same order as that of natural variations in the climate system.

Finally we underlined that although we can force the MPI version to execute the floating point operations in the same order of the sequential version and then the parallel model results are bit-to-bit identical with those of the serial model, we cannot guarantee the results of the serial model have no uncertainty, because the serial model also contains the round-off error. For this reason the "reproducible" version of MPI should be considered only for debugging purpose and it should not be meant as a more reliable version of the code w.r.t. the efficient MPI version.

It would be nice to add information and figures that compare the optimized version of the code with the debug version. Please also include speedup (based on sequential code) information to the benchmark results. How long does it take the sequential run of same model configuration? This will give more information to the potential users about the overall performance of the parallel version of the model.

Summary of changes: We enriched Fig. 11 reporting the performance comparison among the efficient version of the model, the execution without I/O and the debugging

version. We also added the detailed execution time in Table 3.

The benchmark results indicates also issue related with the parallel I/O implementation that could be addressed in the future version of the model. Especially for high-resolution cases with high-frequency output, this could be a bottleneck and might affect overall performance of the modeling system. Using third-party libraries like PIO, SCORPIO or ADIOS might be a better approach rather than using in-house parallel I/O implementation since they are well tested and proved their scalability for different applications.

Summary of changes: We thank the reviewer for these suggestions. In Section 4.4 we added the following sentence:
`we are evaluating the adoption of efficient external libraries to enhance the I/O performance in the next version of the model code. Among the suitable I/O libraries, we mention here XIOS, PIO, SCORPIO or ADIOS`

Typos/Figures:
- The multiple citations needs to be formatted correctly. For example, page 2, line 25 will be in following form (Casulli and Walters, 2000; Chen et al., 2003) and page 2, line 26 will be in (Danilov et al., 2004; Zhang et al., 2016; Umgiesser et al., 2004).
- Please combine Fig. 11 and 12 and present like that (red vs. blue etc.). It is hard to compare the benchmarking results (with and w/o I/O).

Summary of changes: We thank the reviewer for pointing this out: we modified the citations format in order to be compliant with the reviewer suggestion. We also merged Fig. 11 and 12