Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., referee comment RC2
https://doi.org/10.5194/gmd-2021-319-RC2, 2021
© Author(s) 2021. This work is distributed under
the Creative Commons Attribution 4.0 License.

Comment on gmd-2021-319

Ufuk Utku Turuncoglu (Referee)

Referee comment on "Parallel implementation of the SHYFEM (System of HydrodYnamic Finite Element Modules) model" by Giorgio Micaletto et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-319-RC2, 2021

The paper demostrates the design, implementation and validation of the parallelized version of the SHYFEM ocean model that uses unstructured mesh. For this purpose, the authors use a set of third-party libraries (PETSc and ParMETIS) to develop distributed memory versions of the model. The paper is valuable because the ocean models that use unstructured mesh are very important for the applications like interconnected basins with narrow straits etc. and having efficently paralelized versions of those models will help to design more realsitic applications. The paper is well presented and written but having issues with reproducibility is my main concern and I think it needs to be addressed since it is the one of the most important aspects of scientific applications and tools.

General comments:

- please define graph partitioning with one sentence in section 3.3 for the readers that are not familiar with the concept.

- In figure 5, it is demonstrated that the sequential code uses SPARSEKIT. Please add a short description of it to the text. Is this the tool mentioned in the following link, https://www-users.cse.umn.edu/~saad/software/SPARSKIT/. If so, please also include it in the references. If it is not relevant to include it, then the sequential part can be removed from the figure.

- the authors are discussing the convergence criteria In Section 4.3, specifically at line 326. Is there any other criteria used for the convergence such as maximum number of iteration etc.? In some cases/applications, it would be hard to meet the defined tolerance criteria.

- It seems that the solution vector needs to be gathered and redistributed again due to the mismatch between the domain composition defined in the model and the PETSc. As it indicated in the manuscript, the domain decomposition is defined in the pre-processing stage at the beginning of the simulation through the use of the Zoltan package. Is it possible to reorder decomposition in the PETSc to get rid of collecting and redistributing the solution in every iteration? It could create a bottleneck in the overall performance of the model especially for the high resolution applications. The performance benchmark results are also supporting low efficiency of the model when the number of processors is increased. Is there any available tool that creates a link between Zoltan and PETSc to solve similar issues? If the authors could clarify it in the text that would be great. It would be also nice to add references that mention about the similar issue.

- The section 4.4 mainly aims to give information about the I/O management but it does not give details of the parallel I/O library used to read initial and forcing data. This section needs to be extended to include the details of the used library for the parallel I/O and also its scalability (may be in the validation section). Additionally, since each process reads its own restart file, it seems that it is not possible to restart the model using a different number of processors than used in the initial run. This might add extra limitations to the usability of the model and need to be more emphasized. Since the model restarted in a parallel fashion, is there any issue related with the bit-to-bit reproducibility of the simulation results. What are the restrictions in the current implementation in terms of reproducibility?

- Which tool is used to interpolate the forcing data from the regular grid to the unstructured mesh? Is it custom code? What kind of interpolation algorithm is used here (conservative, bilinear etc.). Since the forcing might include flux components, it would be nice to use conservative type interpolation for those variables. It would be also nice to include information about the interpolation of vector fields and how they are handled. The tool might need to preserve the properties of the vector fields after interpolation.

- To assess the decoupled effect of the I/O and computational performance of the model, it would be better to run the model with the configuration that does not write the output. By this way, the pure performance of the MPI implementation can be clearly seen. Since, I/O can be affected by its way of implementation in the model and the architecture of the underlying system (number of I/O nodes, their interconnection, number MDS and OSS servers, RAID configuration and used parallel file system) it would be hard to assess the effect of the I/O in the benchmark results and could add extra uncertainty of the performance measurements and results. Personally, I prefer to see the pure MPI communication overhead in the benchmark results.

- memory scaling is mentioned about one of the key advantages of the distributed memory approach but there is no any information about the scaling of the memory in the current implementation. It would be nice to add a plot or table (total memory per node vs. number of processors etc.) that includes more information about memory scaling of SHYFEM-MPI model. The memory usage can be monitored by some open source and public tools such as Valgrind etc.

- In the validation section (5.1), author's point that the SHYFEM-MPI model is not bit-to-bit reproducible due to the order of the floating point operations. Although, it is not clear what kind of operations are mentioned in here (it would be nice to extend this part little bit), intel compiler provides way of ensuring order of the floating point operations such as in reduction operations via using special -fp-model precise flag. The special compile flag could slow the model around %10 but it could also help to achieve bit-to-bit results. Is this flag used when building model as well as PETSc library? Personally, two different execution of same configuration with same domain decomposition and same number of cores must produce identical results but it seems that SHYFEM-MPI model does not provide this capability, which is the main drawback of the current implementation. It also makes hard to couple the SHYFEM-MPI model with the other earth system model components (i.e. atmosphere model) because the non-linear interactions among the model components could lead totally different answer, which is very dangerous. The authors also discuss about a version that could create the reproducible results but I am not sure why it is just for debugging. I think that version needs to be publicly available. I think providing a model code to the ocean user community with a lack of reproducibility is very dangerous. I strongly suggest that the authors need to publish their reproducible version of the code in the code availability section. It would have some performance problems but at least it can be used safely in the scientific applications.

Typos:

- Page 4 / Line 96: Referencing to the tables and figures must be consistent across the document. In the same cases the table and figure numbers are given inside the parenthesis and others not. Please review them carefully. For example, table (1) needs to be Table 1.

- Page 11 / Line 225: Please cite PETSc correctly using information in the following link, https://petsc.org/release/#citing-petsc

- Page 17 / Figure 6: I think there is something wrong with Figure 6. There is no top, bottom-left etc. So, the figure caption and figure needs to be consistent.