Dear editor and reviewer,

First of all, we would like to express our sincere appreciation to your valuable feedbacks. Your comments are highly insightful and enable us to substantially improve the quality of our manuscript. Below are our point-by-point responses to all the comments. The red fonts describe how the manuscript has been modified.

Responses to the comments of referee #1

Major revision comments

1. Major revision comments 1 and 2. “A new paragraph in the introduction explaining why this framework is necessary given other frameworks that already exist in an open-sourced format. I think this would help avoid any assumptions of redundancy that could be confusing to readers familiar with other efforts in this space. A new paragraph in the conclusion comparing and contrasting this new framework with other existing frameworks. This could also just be a table or whatever format is easy for the authors.”

[Response]:

To address these concerns and to maintain the overall structure of this manuscript, we have added a discussion section. It now clearly explains why this framework is necessary to us, and its broad implication to the community. For the comparison with existing frameworks, while GRIST implements some techniques and choices that have been already used in the community, its scientific models are different. Developing these models is closely connected with the parallel infrastructure, requiring tailored software engineering efforts. We provided Table 2 to give some details of GRIST. This table follows the conventions of Ullrich et al. (2017) and briefly summarizes some key features. We make a comparison with three models in Ullrich et al. (2017) to describe the unique aspects of GRIST. All this information is given in Section 5.

These two paragraphs in Section 5 explain why this framework is necessary for us: "First, the authors understand…", "Second, our intention for global atmospheric modelling…".

This paragraph in Section 5 discusses the broad implication: "While GRIST is still under active development…"

This paragraph in Section 5 compares GRIST with three existing counterparts that have been used for weather and climate modelling (MPAS, ICON, FV3): "As a response to one reviewer, it is also worthwhile to pinpoint how GRIST differs from existing counterparts…". Table 2 can be compared to the tables in Ullrich et al. (2017).

2. Major revision comment 3. "I think it would be beneficial for the authors to try and add some serial (and lower processor count) data to figure 3. It looks like the authors stop exploring their strong scaling space at around 250 processes, and it's hard to judge what the loss in parallel efficiency is without these numbers."

[Response]:

We have added the test results with lower processor counts in Section 2.3, including two subfigures in Figure 3 (Figures 3(c) and 3(d), also illustrated below) and a new paragraph discussing the results. Due to the memory limitation, we are not able to run the tests with processor counts lower than 300 for the G10 grid. Therefore, in Figures 3(c) and 3(d), we have shown the results for the G8 grid with processor counts ranging from 2 to 4200, from which we can observe the super-linear speedup phenomenon for low processor counts, and the index reordering strategies can accelerate the calculations. The parallelized code requires at least two processors to run. The one-to-many consistency is guaranteed by maintaining a separate serial version that only contains a minimum code segment for examinations, but this serial version is not very suitable for a reference of computational performance.

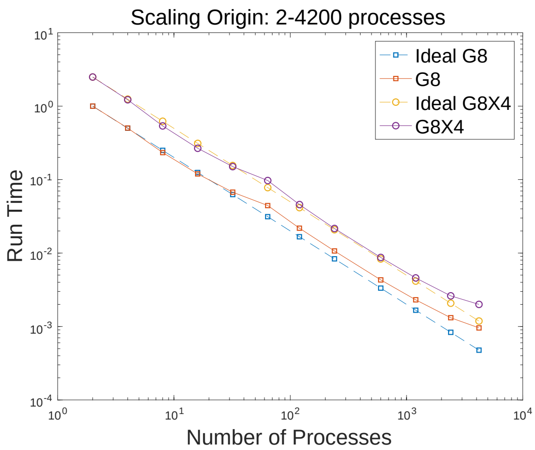

Figure 3 (c)

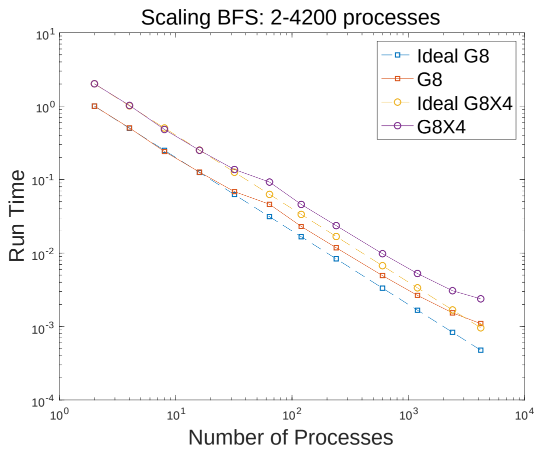

Figure 3 (d)

Minor points

1. Minor point 1: While the paper as a whole recognizes prior contributions by frameworks such as MPAS, the abstract does not. It would be useful to add something to that effect into the abstract.

[Response]:

We have improved the abstract to address this concern.

"GRIST is a new framework in the icosahedral-/Voronoi-mesh modelling community for both research and application purposes. It adopts some well-established techniques and choices that have been used, but supports different scientific models. Developing these models is closely connected with the parallel infrastructure, requiring tailored software engineering efforts. In this paper, we focus on three major aspects that facilitate rapid iterative development".

2. Minor point 2: Based on the usage of METIS, I'm assuming that the authors use the offline capability (specifically in METS) and not the online capability in ParMETIS, but this could be clarified.

[Response]:

Yes, we are using the offline capability of METIS. This has been clarified in Section 2.1 in the revised manuscript. For GRIST, this METIS-offline partition can be done either online (in the initialization) or offline (via a separate driver), and the offline setup is more suitable for very high-resolution runs, since the mesh partition will consume more time than simply reading the partitioned data for high-resolution simulations.

3. Minor point 3: Line 40: "which reduces the" -> "which can reduce the". Line 80: "establishment of numerical modelling" -> "establishment of a numerical modelling"

[Response]:

Thanks. We have corrected these statements.

4. Minor point 4: Section 2.3: You never mention that the G8 performance actually is degraded by using the BFS reordering strategy. This is also mentioned again on lines 370-372. This is important to notice, because the reordering strategies become unnecessary once the problems are strong scaled out to a certain point. It could also be useful to give a description of what this point is as a function of number of cells / process.

[Response]:

We have added the statement of the degradation phenomenon of BFS strategy for the G8 performance in the last paragraph of Section 2.3, and referred to Section 3.4 for the discussions. In the last paragraph of Section 3.4, we have added the sentence "Based on our tests, we find out that the index reordering will become unnecessary when there are less than 1000 cells/process". Thanks for your comments.

5. Minor point 5: Section 4.2: This essentially describes NCAR's PIO library, which is available on github. Though no reference to it appears in this section, and no discussion of why this was rewritten instead of used wholesale appears. This should definitely be added. It could be as simple as describing why it is not the same as NCAR's PIO library, and moving on from there though.

[Response]:

We have added the discussion with the PIO library in the second paragraph of Section 4.2. PIO is a high-level parallel I/O library for the structured grid applications, which also allows to designate some subset of processors to perform I/O like the group I/O method introduced in this manuscript. We note that MPAS uses PIO as its tool. When compared with PIO, the communications of data between the non-I/O processes and the I/O processes for the group I/O method are much easier. This is because the communications for the group I/O method are only carried out between processes in the same group, which can be accomplished by using the 'MPI_Scatterv/MPI_Gatherv' interfaces, and the indices in the I/O processes are not required to be continuous. While for PIO, the indices in the I/O processes should be continuous, therefore the communications between the non-I/O processes and the I/O processes will be more complicated, and the 'MPI_Alltoallv' interface may be used. Anyway, the group I/O method is more tailored for the data structure and distribution of GRIST, and thus can be implemented more easily.

6. Minor point 6: Figure 4: The title on the top left sub-plot is mirrored.

[Response]:

Thanks. We have corrected this.

We really appreciate your highly constructive comments. If there are any other questions, please do not hesitate to contact us.

Best wishes,

Xiaomeng Huang