Title: **GEOS-Chem High Performance (GCHP): A next-generation implementation of the GEOS-Chem chemical transport model for massively parallel applications**

Authors: **Sebastian D. Eastham et al.**

Journal: **GMD**

Ref.: **gmd-2018-55 (MS Type: Model description paper)**

Summary

I am not sure what to expect from these sort of papers as they do describe complex systems with only couple of pages of words. So if you are an outside reader then it is almost impossible to get a firm idea about the described system. It is also debatable whether these brief descriptions have any usefulness to other researcher? The model described in this paper has mainly two components, (i) a chemistry module, which is local and should, in principle, have a near perfect scaling (without any MPI instruction), while the second component (ii) is advection, which is not local and requires halo filling (or communication on demand) for parallel implementations. The only possible reason for publishing such papers would be to provide a reference for the model in question.

Detailed comments:

1. I think the paper needs a table summarizing the definitions of all the acronyms (GEOS-Chem, CTM, GCHP, ect...) used, because I kept going forward and backward to look for their meanings in the text.

2. My understanding of atmospheric chemistry models is that they solves a system of coupled ODEs at each grid of the model and each grid point is, in principle, independent of its neighbors. In other words, the chemical model needs information on the grid only and therefore such process should scale perfectly and these models are ideal for parallelism. However, advection needs information about the characteristic (hyperbolic problems) and this part that needs effort to make it work with parallel implementation (MPI). This looks like a coding task of combining FV3 transport (Lin et al.) with an existing chemistry model (with OpenMP). I think, the whole description of the new system and how its differs from the original GCC could be improved. There should be more clarifications and a detailed description of what changed from the original code (probably a table listing all the components of GCHP and how it differs from the original GCC).

3. Page 1, Abstract, Line 2. I am not sure how large the system of chemical species is? It would be better to give an order of unknowns for a typical system, i.e., order 100-1000000?.

4. The discussion at page 7 in relation to semi-Lagrangian, cube-sphere and lat-long grids is a bit subjective!

   (a) I am not sure what the authors means by "inconsistency in technique" in the sentence "The problem can be mitigated .....in ensuring mass conservation"?

   (b) I think the whole passage "In an MPI environment, ... sharing the pole." cannot be justified. Semi-Lagrangian (SL) schemes on any grid with the right halo size can achieve near perfect scaling [see for example with up to 100000 cores SL scalability experiments in the paper: *High-performance high-resolution semi-Lagrangian tracer transport on a sphere, J.B. White III and J.J. Dongarra, Journal of Computational Physics, Vol. 230, pp. 6778–679 (2011)*] or up to 10000 cores scalability results in *Allen and Zerroukat, Journal of Computational Physics, Vol. 319, p. 44-60 (2016)]*. Certainly the only argument against SL is the lack of inherent conservation, but in terms of accuracy, stability, simplicity, computational cost, and scalability there is no valid arguments against SL.

   (c) Of course there is no perfect scheme and every approach has some advantages and disadvantages and one has to be objective about these things. For example the cube-sphere has the disadvantages of the complexity of dealing with 6 panels and their orientations and the non-orthogonality of the grid and its associated grid-imprinting (for example see your figure 5 for C24 where it is very clear that the pattern of the tracer distribution is very much influenced by the grid). It would be interesting to show the equivalent pictures (blowup pictures at Figure 5 for C24) due to the original GCC!.

5. Page 8, 1st paragraph of section 2.3.2. In parallel domain decomposition, I would suggest to use the right nomenclature of halo-filling instead of "providing boundary conditions", because there is no real boundaries between processors.

6. Page 8, end of 2nd paragraph of section 2.3.2. It is not clear what does mean the comment: "GCHP defaults instead to a simple global air mass correction also applied to tracers"?? My understanding is that the main purpose of using FV3 is to achieve mass conservation in an inherent way so I don't understand what is this correction is doing? Does it mean the inherent conservation is lost? Further clarifications are needed here.

7. I don't see any use for to the bottom sub-figure (time/1000colums) of figure 3.

8. Section 4 (example simulation). It is good to show how the scalability of GCHP can be exploited to run higher-resolution simulations and the improvement that comes as results of that (Figure 5). However, I feel there are some figures missing in this section/paper. Although there is a lot figures comparing GCC and GCHP in terms of computational efficiency (scaling and wall clock time), for fidelity there should be at least a couple of figures comparing (at least visually) GCC and GCHP for a benchmark test to show that the solutions of the two models are, at least visually, in a good agreement.