Reply on RC2
Jeremy McGibbon et al.

Author comment on "fv3gfs-wrapper: a Python wrapper of the FV3GFS atmospheric model" by Jeremy McGibbon et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-22-AC2, 2021

Thank you for your review! We appreciate your very precise and clear comments.

Title and abstract

- Added definitions of NOAA and FV3GFS
- replaced “state” with “variable”

Issues and questions in the main text

1. Acronym defined.
2. Citation added.
3. Citations added.
4. Acronyms defined.
5. Added.
6. Updated.
7. Updated where noted, and elsewhere in the document.
8. The sentence and the preceding one have been re-worked.
10. Updated.
11. Updated.
12. Updated.
13. Updated.
14. Updated, with slightly briefer phrasing.
15. Defined.
16. Reformatted lists as part of a sentence.
17. Updated.
18. Updated.
19. Updated.
20. Updated.
22. Updated.
23. Updated.
24. Placing the fv3gfs.wrapper import at the top of the file is done intentionally, because depending on the system configuration, the import can cause a segmentation fault if done after importing numpy. This happens if there is a conflict between libraries used to
compile numpy and the wrapper. We ran all code examples through flake8 and found no errors. Based on the reviewer’s suggestion we moved the standard library import to come before external library imports in the machine learning example.

25. This is not a typo, mpi4py provides capitalized routines which take in numpy arrays (or anything providing a buffer interface) which are low-level and efficient, and non-capitalized routines which take in arbitrary python objects and use pickle under the hood. Several non-communication routines like Get_rank() are also capitalized.

26. We assume the line number here is mis-labelled since github does not appear on this line. Updated to “GitHub” spelling in code and data availability section.

27. Defined “getters and setters”.

28. These two sentences have been reworked and are hopefully more clear.

29. Added hyperlink to Jinja documentation

30. This text was removed in response to another reviewer comment.

31. To avoid adding another tense to the sentence, we fixed the grammatical error by saying “than using automated wrapping tools”.

32. Replaced with the suggested wording.

33. Updated.

34. Updated.

35. Added a webpage reference.

36. Updated.

37. Added a reference and re-worded statement.

38. Re-worded the explanation in concrete terms, which should be easier to follow.

39. Updated.

Figures

1. This is either too simple or too complex to include in the diagram, depending on how you interpret “user input”. The only user code included in the diagram is the “User script” block, so user input and output should be only to and from that block. If you mean “files provided by the user”, this also includes files read by the Fortran model, which we don’t want to document here. Depending on the routines used, the file handle open might occur in the user script or in a library such as xarray or gcsfs. However the place where the purpose of those files and how they are operated on is defined would be the user script. To keep the figures simpler, we would like to exclude I/O details from the figure and leave it up to the reader’s imagination how they might perform I/O in their user script.

2. We experimented with several alternatives to arrow-heading on these two figures before settling on the final design. The initial design had arrow heads on all lines, as suggested by the reviewer. We felt this made the meaning of the arrowheads unclear, and decided instead to use arrows to emphasize the Fortran model main loop and inputs or outputs of the Fortran model. We have added a description of the purpose of the arrowheads to both figure captions.