

The article **fv3gfs-wrapper: a Python wrapper of the FV3GFS atmospheric model** presents a Python wrapper for an atmospheric model written in Fortran. The wrapper method is detailed, with reasons behind each choice presented. This article is useful as it presents a tool that can improve the accessibility and reproducibility of scientific studies that make use of the FV3GFS model, and also because its presentation can act as a guide to other researchers and developers who use other large models written in Fortran. I believe this article may be suitable for publication once improvements are made to the referencing, figures, and grammar.

## 1 Title and abstract

The title is clear and appropriate. Although it contains two acronyms, the type of object that the acronyms are referring to are clear in context.

In the abstract, acronyms are not defined. In addition, on lines 5 and 7 the way the word “state” is used on these lines is grammatically incorrect. I suggest adding an article or even expanding it to be clear exactly what the authors are referring to, so that a developer or user who is not as familiar with the code or is new to using models would know what they’re talking about.

## 2 Issues and questions in the main text

1. Line 12: FV3GFS acronym needs to be defined at its first use.
2. Line 16: Missing reference for Fortran.
3. Line 16: Missing reference for OpenMP/MPI.
4. Line 16: MP/MPI acronyms need to be defined.
5. Line 20: Missing a comparison element, a possible element that the authors will likely want to replace is included in all caps as an example : “...available or widely used in LANGUAGES LIKE FORTRAN as the are in...”
6. Line 22: Remove “Python”, to instead say: “...enable a much larger user base to interact...”
7. Lines 32, 108, and 275: “which” should be “that”, since it is not preceded by a comma. Or you can add a comma before “which”. Examples:
  - “I reviewed a journal article that contained a lot of words.”
  - “I reviewed a journal article, which contained a lot of words.”
8. Line 32: This sentence is awkward, recommend rewording. Potential example: “For example, user code that modifies standard model behavior by providing training through machine learning routines must first be run

in Python, then exported for use by the model code, and finally the model code must also be modified to accept the trained input.” This example could likely be improved by the authors, who perhaps could use my interpretation of their intended meaning to further clarify their example.

9. Line 50: Use either “IO” or “I/O” consistently throughout paper and define upon first use.
10. Line 56: “...Python, it is advantageous to be able to..”
11. Line 60: Awkward sentence structure. Recommend something like, “This is not the first time Python and high-performance compiled programs have been combined.”
12. Line 65: “In astronomy, Python...”
13. Line 66: recommend replacing “give” with “provide”
14. Line 71: “...Fortran models (an action that may be taken to, for example, allow the code to run on GPU architectures) may benefit...”
15. Line 71: Define GPU
16. Lists near Line 75 and 85: Recommend formatting list as part of a sentence (adding a comma at the end of each non-terminal item and a period at the end of the terminal item).
17. Line 84: “...model behavior, we will show how to:”
18. List near Line 85: Recommend rewording the beginning of each item to follow logically from the end of line 84.
19. Lines 88 and 94: “...to reproduce, bit-for-bit, the results...”
20. Line 95: “...wrapper executes Fortran code identical to the original model, bit-for-bit...”
21. Line 115: Recommend pointing the reader to the “code availability” section here, so that they don’t think you’ve forgotten to link to the wrapper repository.
22. Lines 115-116: “The timings for each of these examples are included...”
23. Line 119: The public repository information is not included in the acknowledgements, update this to the correct section name.
24. Code Examples: PEP8 compliance should be improved for all code examples. One obvious violation is the import order: <https://www.python.org/dev/peps/pep-0008/#imports>
25. Line 194: Potential bug/typo in method name: `MPI.COMM_WORLD.Get_rank()` or `MPI.COMM_WORLD.get_rank()`?

26. Line 204: “Github” should be “GitHub”.
27. Line 218: “getters” and “setters” need to be defined, so that a non-experienced reader can understand what they mean.
28. Lines 222-224: Sentence spanning these lines is hard to follow. On the first two readings I thought it was merely a run-on sentence, but now I think it is a list that is just missing some articles. Recommend making this clearer.
29. Line 225: Need a reference for Jinja templates.
30. Line 237: “However, once defined, the Docker container...”
31. Line 248: “...involves writing more code that would have been necessary if an automated wrapping tool such as f90wrap (Kermode, 2020) had been used, it provides...”
32. Lines 250-252: Some parts of this sentence could be clearer. Perhaps: “With this approach, Python wrappers can be produced for very complex build systems with only minimal modifications (such as ensuring the necessary variables and routines externally accessible) to the existing model code.”
33. Line 253: Recommend replacing “On the other hand” with “However”, since the bookending phrase “On one hand” is not used.
34. Line 257: Recommend clarifying text to something like “...split the FV3GFS model main loop into...”
35. Line 264: Provide a reference for python-mpi-bcast. This can be done by providing a webpage reference to the GitHub repository.
36. Line 271: “...FV3GFS that, with or without a wrapper, it is...”
37. Line 272: Recommend providing references supporting the statement “Parallel code is difficult to test and can result in race conditions.”
38. Line 272-275: These sentences are hard to follow, please rewrite.
39. Line 282: “...unless routines to copy the model state in and out...”

### 3 Figures

Figures 1 and 2 would be improved by supplying:

1. Blocks indicating where user input and output lie in the workflow
2. Arrows on all lines, double headed if need be