Answer to Mr García Rodríguez, Referee #1

We thank Mr García Rodríguez for his comments and suggestions. We answer point by point in the following with the reviewer's comments added in *red/italics*.

From a software engineering perspective applied to climate models, it would be interesting to gain a deeper understanding of the architectural decisions behind the development of the model's code. Therefore, we pose the following questions:
For the new configuration, were specific design patterns such as Facade or Dependency Injection used to structure the model's code? This would ensure the separation between physical parameterisations, dynamic processes, and the model logic.

The upper atmosphere extension of the ICON model was already developed by Borchert et al. (2019) for ICON's two physical packages available at that stage. We base our adaptations on the existing model structure and design, which was created by the partners of the ICON consortium. Changing this model structure is beyond the scope of this study.
The aim of the paper is the tuning of the model performance in the middle atmosphere. We focused on gravity wave drag (GWD) because the orographic waves are usually taken to cure the "cold bias" in the winter stratosphere and non-orographic gravity waves are the main forcing of the dynamics in the mesosphere and lower thermosphere. Systematic variations of GWD parameters were undertaken (see Tab. 3), and their success was evaluated by analysing the typical wind and temperature structures in the stratosphere and mesosphere. This does not exclude the additional influence of other factors, such as prescribed SST and ozone on planetary wave structures and some warm bias in the lower stratosphere. However, the goal of qualifying UA-ICON for middle atmosphere applications of global circulation and stratospheric warming anomalies was achieved.

To coordinate the interactions between the different physical and dynamic parameterisations, do you use patterns such as Observer or Mediator? If so, what benefits have you observed in terms of performance, maintainability, or scalability?

None of these tools have been used.

To manage complex parameterisations (such as orographic gravity waves), have specific techniques based on patterns like Strategy been implemented?

To enable the testing of multiple parameter sets, we have implemented the possibility to run the model in perpetual month mode. This allows for much shorter model simulations for the evaluation of the effects of parameter changes on the climatology of the model.

To ensure the reliability of both the simulations and the implemented code, have automated testing frameworks or static code analysis tools, such as pFUnit and FortranAnalyser, been employed in the development process? If so, how have they contributed to identifying and addressing potential issues in the codebase? The use of tools such as FortranAnalyser would be interesting to mention in order to be able to verify that the quality of the developed code is maintained or improved with the development of new versions of the software.

Tools for code analysis, such as the mentioned FortranAnalyser, have not been used during the work for this manuscript. However, we have checked the ICON code published on Zenodo with FortranAnalyser and got a final score for the ICON/src directory of 4.1.

References

Borchert, S., Zhou, G., Baldauf, M., Schmidt, H., Zängl, G., and Reinert, D.: The upper-atmosphere extension of the ICON general circulation model (version: Ua-icon-1.0), Geosci. Model Dev., 12, https://doi.org/10.5194/gmd-12-3541-2019, 2019.