Reviewer’s 3 Comments

> *Regarding section 2.4 “Code optimization”: Could you give a rough estimate of how much real time running ISO_simulator takes? Are you using any of the parallel/distributed/out-of-core computing functionalities provided by cf-python or Iris via Dask? This could potentially lead to large performance improvements.*

ISO_simulator is a relatively fast code to run. Reading of the model data is the single, most time-consuming step; therefore the number of variables in the model hourly output file will also have an impact on the ISO_simulator run time. An estimate of the time required to run ISO_simulator is added at line 149 with the following sentence:

“In practical tests, when run over a large number of years, ISO_simulator takes around 2-3 minutes to process one variable for one model year.”

VISION tools use Dask for reading of data with cf-python.

> *Please consider publishing your code as a Python package on PyPI and/or conda-forge to enable installing it via “pip install ” or “conda install ”. This will greatly simplify the installation process, the dependency handling, and the inclusion of your code into other software. For example, other software products could simply do a “import <name_of_your_package” in their code.*

We understand the reviewer’s point here. However, given that version 2 of the VISION toolkit is currently under development and will soon replace version 1, we decided to include these recommendations on the next version of the code release, which will be provided as a Python package on PyPI as suggested and has already been minimally packaged such that it is installable locally from the open GitHub repository.

Specific Comments

1. *23: I think it would be helpful to also include the acronym “ISO_simulator” into the title. You mention it very often in the paper, so I think it deserves to be there.*

We have now added the acronym to the title.

2. *45: There are many models which also use unstructured grids (ICON, FESOM, etc.), so it’s probably better to avoid the term “regular grid”, which really is the opposite of an unstructured (or irregular) grid.*

We have now replaced the text in lines 44-46 from:

“what makes such comparisons with model data inherently difficult is the difference between the orderly model data (defined on a regular 3D grid and at regular time intervals) and the unstructured observational data (with variable coverage in space and time).”

to:

“what makes such comparisons with model data inherently difficult is the difference between the orderly model data, defined on the model grid at regular time intervals, and the unstructured observational data, with variable coverage in space and time.”

3. *83-86: Mention what you need Iris for? Both other tools are mentioned here.*

This sentence has now been added at the end of line 81:

“Iris libraries are used in some CIS functions to read gridded model data.”

4. *Table 1: Please mention that input files can also be other formats than PP (like you do in the next paragraph).*

This has now been added.

5. *Table 2: Could you please explain what you mean by “Iris + structured UM loading” and why the difference is so big between “Iris” and that?*

The following sentence has been added to the caption in Table 2:

“The structured UM loading[1] method is a context manager which enables an alternative loading mechanism for ‘structured’ UM files, providing much faster load times.”

With the footnote:

[1]https://scitools-iris.readthedocs.io/en/stable/generated/api/iris.fileformats.um.html#iris.fileformats.um.structured_um_loading

6. *143: Iris 3.1.0 is very old (Sep. 2021), have you considered using a later version?*

We agree with the referee and have decided that, for the purpose of Table 2, we will be using more recent versions of the same Python libraries, specifically CIS v1.7.9, Iris v3.10.0 and cf-python v3.16.2.

7. *189-191: You already mentioned a lot of this in the paragraph l.180-185, maybe you can unify this?*

We have now removed the sentence in line 180-181:

“In this section, we show some examples of using ISO_simulator to co-locate UKESM data to the same time and location as different types of observational datasets.”

Technical Corrections

1. *35: “NERC” is not defined*

Done

2. *114-115: “input variable” -> “command line argument”*

Done

3. *188: “UAV” is undefined*

Done