

MESMER v1.0.0: Consolidating the Modular Earth System Model Emulator into a Sustainable Research Software Package

Final Author Response

Response to Referee Comments 2

Comment 0

The authors present a consolidated version of MESMER, a fast climate model emulator based on a previously published statistical methodology. This is a well-written technical contribution that I enjoyed reading. I commend the effort of the team in making the code more accessible, with a focus on software engineering aspects that (as the authors also note) are often overlooked in research code. I also appreciate the effort of bringing the different flavors of MESMER together, which I think could prove useful in making the emulator more widely used by a larger community of researchers and/or stakeholders that are hungry for climate information. I have a few minor comments on the paper that could be worth addressing, and some food for thought for future releases of MESMER given the current state of the model.

Response

Thank you for the positive evaluation of our study and the helpful comments. We address the comments made below and explain the modifications we made to the study.

Minor Comments to the authors

1. L56: “land surface annual mean temperatures” – I’m assuming this is actually surface air temperature over land (tas in CMIP6 jargon), and not the actual land surface temperature. Worth rephrasing for clarity?
2. On a similar note (L343): is there a reason why you chose to exclude the ocean from your emulation? It seems that the main audience of the code is researchers, and I can think of a lot of research problems where the ocean response matters significantly (evaporation, hydrological cycle changes, low cloud feedbacks and boundary layer stability, SST pattern effect, dominant modes of internal variability [ENSO/NAO/...], ocean heat content and TOA radiation,....) . Even for general stakeholders, things like marine heat waves are becoming more prominent and relevant for impact assessments, and knowing the ocean response seems quite key. Maybe worth including (not necessarily in this paper, but in the future)? It would probably even simplify the code to some extent as no masking would be necessary.
3. Figure 4 caption and elsewhere: I just wanted to flag the possibility for any web link to be broken in the near future (few years timescale), whereas the paper is usually thought to be a much longer-term stand-alone contribution that will likely outlive the current web links (e.g. if you decide to change domain, not maintain mesmer anymore, or other possible unforeseen circumstances the links will be broken with no real possibility of updating them, since the paper after publishing is usually not updated anymore). Not necessarily suggesting to remove links (I see why they could be useful), but something to think about. I’m flagging this because I’ve recently read a few papers from the 2010s with links embedded in them that are now broken and not really useful anymore.
4. L383: can you be a little more precise on what you mean by “annual maximum temperature”? i.e., what time resolution is used to calculate the maximum (for example, is that the annual maximum monthly temperature, the annual maximum daily temperature or the annual maximum hourly temperature). Might be worth using CMIP6 jargon if it is defined in CMIP6 and it is not a postprocessed variable.
5. I tried to pip install the mesmer-emulator package and run the “Emulating near surface temperature on land with MESMER” tutorial, but it looks like the mesmer.example_data file was not included in the pip install (I believe along all the other *.py files in the mesmer/mesmer directory, e.g. anomaly.py, etc). I am using an Ubuntu machine and just followed the installation commands in the doc page, i.e. ran `python -m pip install mesmer-emulator[complete]` in a virtual environment. In my experience, a research software package becomes quite successful if the user experience is frictionless, meaning that it works out-of-the-box without much debugging to be done on the user side. Small details like this one might discourage users to keep digging into mesmer, which would undermine lots of great work that could be really useful to a lot of people. Just a suggestion to pay attention to these details and extensively test with different people, machines, ...
6. On a similar note to #5, I wonder if it could be useful to develop an out-of-the-box version of mesmer (sort of like an old-fashioned executable, or maybe some high-level wrappers) that produces some basic output based on a few input parameters (e.g. N realizations of gridded

tas for a specified experiment and model, where N, model, experiment are input parameters). In other words, I found the tutorials a bit too detailed (there is still a lot of processing, data handling, ...) for the casual user that just wants to try the model out. Or perhaps there could be a range of tutorials, from a really simple one for the casual user, to the more processing/data handling ones that could be geared towards developers and people that want to dig more into mesmer. Not necessarily for this paper but it's something that might be worth thinking about for the future if the goal is to make mesmer a widely used emulator in the community.

Response

1. Yes, thank you. We changed the instances of “surface temperature” to “surface air temperature” and specified “annual mean of daily land surface air temperature (“tas”)” in section 2.1.
2. Thank you for this question. We agree that the ocean response is vital to the future climate. The impact of the ocean response on the land is still included in our emulations, since we train on fully coupled ESM output. MESMER applications to date were focused on terrestrial climate impacts (e.g. Schöngart et al. 2025). It should be possible to fit and emulate ocean grid-cells with the MESMER code (but this is not something that we tested). However, more grid-cells will lead to a more severely rank deficient covariance matrix, requiring stronger regularisation or making the use of multiple ensemble members for the calibration even more important. We also want to note that we tested the behaviour of MESMER for strongly non-linear climate responses, like for example AMOC tipping.
3. Thank you for flagging this. We removed the web links in the caption of Figure 4, as they also impeded readability. As for other weblinks in the manuscript, we decided to leave them as they are either tightly linked to the content of the paper, like the github page or readthedocs page, or are the only references where no doi exists, like the numpy documentation or the GNU license.
4. Yes, we specified “annual maximum of daily surface air temperature (“txx”)” in Section 2.1., but still refer to it as annual maximum temperature elsewhere for improved readability.
5. Thank you for testing the MESMER software! We updated MESMER to automatically download the tutorial data when executing the tutorials (<https://github.com/MESMER-group/mesmer/pull/846>).
6. For MESMERv1, the focus was primarily on improving the software so that it is a good starting point for further developments. We wanted to make the software flexible for users to extend and apply it to their own project by having understandable documentation and in-depth examples. However, we agree that there is a use case for out-of-the-box emulations, which we also want to facilitate, e.g. by providing pre-calibrated parameters. Our envisioned future for easy emulation is a light weight web-interface for MESMER, where a limited set of variables can be emulated at the click of a

button. Thank you for your feedback! It is very useful for us to see that there is interest for such an application!

Note to the referees and editors

We will release the MESMER v1.0.0 upon final acceptance of the manuscript. We uploaded an updated version of the analysis scripts to <https://doi.org/10.5281/zenodo.19203528>. Some of the numbers in Table 2 changed slightly, due to ongoing developments in the MESMER Code base, see <https://mesmer-emulator.readthedocs.io/en/latest/changelog.html#v1-0-0-unreleased> and screenshot of latexdiff of Table 2 in the manuscript below. The corresponding information in the text has been updated as well.

Target	Metric	Value
Modularity	Average number of lines of code per object	38 37
	Number of lines of code per file (avg [min, max])	238 244 [7, 1219]
Code quality	Pylint score	8.74
Data structure	Numpy arrays to xarray data structure	Completed
Documentation	% of public API documented	100 99 %
	% of private API documented	53 %
Testing	% of lines covered by tests	99 %