Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-100-RC2, 2020

Geoscientific
Model Development
Discussions
Open Access
EGU

Interactive comment on "ESM-Tools Version 4.0: A modular infrastructure for stand-alone and coupled Earth System Modelling (ESM)" *by* Dirk Barbi et al.

Anonymous Referee #2

Received and published: 30 November 2020

General comments

Utilising Earth System Models (ESMs) for numerical experiments in climate research is undoubted a complex task for many scientists and any attempt to overcome the complexity is welcome. The ESM-Tools constitute such an attempt and the list of supported models, components, configurations, and research groups testifies that the approach has successfully used for a wide range of use cases. The article gives, in more than one respect, a good overview about the concept, implementation, and usage of the ESM-Tools. As such, the topic is very relevant and well within the scope of the GMD journal.

Printer-friendly version

Discussion paper

The ESM-Tools cover support for all common workflow steps for numerical climate experiments. The software appears well structured and build on clear concepts. The effort to document, provide information and generally interact with users via different channels is impressive.

The manuscript is well structured and covers the essential aspects of the ESM workflow, as well as a good overview about concepts and implementation. Nevertheless, revision is needed, in my opinion, with respect to the following aspects:

(1) General overview of the overall workflow

It should, throughout the text, be made clearer how the ESM-Tools work in detail. Examples should be given in terms of specific ESM-Tool commands for workflows such as:

* getting an existing model to run from scratch

* adapt a new model to the ESM-Tools

* add a new function/feature

The examples could be given in the form of YAML snippets (see next point) to better illustrate the workflow description.

(2) Examples or YAML snippets

The ESM-Tools are controlled by YAML files and as such a crucial concept, it needs a clear description. The YAML files in the appendix are good, but it would help the understanding to give short YAML snippets in the text to illustrate the features explained. A particular example is section 2.1.2 where extensions to YAML are listed.

(3) Language

The language of the manuscript is sometimes rather informal or unprecise, such as in the following examples:

* p2l37 "often unnecessary"

* p4l107: "as many as sensible"

* p14l283: "cost the nerves"

These phrases leave sometimes the impression of opinions, rather than facts. Furthermore, the consistency of terms can be improved. The most prominent example being the term "model", which is sometimes apparently confused with "model component", "model configuration", or "experiment" (see below in specific comments).

Overall, the manuscript is considered well written, despite this criticism.

Specific comments

p2l2: The claim that ESMs are written for specific purposes and lack modularity seems not well founded and seems to contradict the fact that ESMs are often composed of (independent) component models. I do agree with the implication that users need profound knowledge and time to learn a new ESM, but, in my opinion, this is due to complex configurations and lack of configuration management options. There is, often, a lack of modularity *within* component models, but this is not addressed by ESM-Tools either.

p2l39: It is not clear what "script-based coupling" is.

p2l44: The list of tools seems very selective. Without having a good overview of ESM infrastructure tools, it strikes me that three of the four listed tools are from Germany.

p3l68ff: The description of Mkexp seems rather subjective. The language (e.g. "lengthy", "unintuitive") indicates opinions rather than facts.

p3l83: It is not clear what the "specified criteria" are or where they are specified.

p4l94: The sentence "To achieve this, the code ... is organised such that all information about the model (e.g. HPC systems, ...) ..." does not make sense to me. First, it seems

Printer-friendly version

Discussion paper

to me that not the "code" is organised, but that it is the concept of ESM-Tools to provide information in YAML files. Second, the "HPC system" is not, in my view, information about the "model". I would argue that the HPC system is a property of an experiment, or even more specific, one experiment run. You could have the same model in various experiments on different HPC systems.

p4l96: It is not clear what the term "re-coded" is referring to. Is it previous versions of ESM-Tools?

p5l113: Again, it is not entirely clear to me what the term "model setup" refers to. Is it the model configuration (e.g. components, grids, ...)? Or does it include experiment parameters?

p5l124ff: Maybe this part of the section should not be a bullet point list.

p6l131ff: The detailed workflow for pulling component models from their respective repository and putting them into a coupled model is not clear. It says that "compile time switches" are activated but that means that the components must be prepared for the coupling in the first place. This would be reasonable assumption, but what if further changes are needed? In some way, the section explains the downstream view of things ("huge improvement for model developers") but not upstream or deliberate isolated contributions.

p6l43: Does the sentence indicate that dependencies are "baked into" ESM-Tools? That would indeed make installation easier, but it has obviously disadvantages as well (e.g. outdated dependencies). If external packages are included with ESM-Tools (and the fact mentioned in the article), it should be mentioned which and why.

p7l165: It is not clear which metric "most extensive" refers to. Also, "optimized implementations of all the functionalities required to run an ESM" confuses me. Which functions are meant and in which ways are they optimised?

p8f2: What are "user defined functions", more specifically, who is the "user" in this case? Is it the user that runs an experiment or the developer that prepares one model (configuration) for use with ESM-Tools?

p8l174: It would be good to explain how esm_viz interfaces to the raw model output for monitoring. An example YAML file/snippet illustrating the monitoring configuration would be helpful.

p8l188: What is the "modular layout of the repositories used by the ESM-Tools"? Does it refer to the ESM-Tools code or the repositories of the models (model components)? How does the GUI make use of the modularity?

p9l191ff: It would be nice to give an overview, either in this section or earlier, about the specific steps needed to port a new model to ESM-Tools.

p11f4: It is not clear what information is given (above Table 2+3) and if the colours used in the figure have a specific meaning.

p12l231ff: Since the contribution workflow seems to differ from the standard Github clone-pullrequest scheme, it could be more detailed how users "develop their own extensions to the existing code". If(!) I understand correctly, the actual code is not hosted on Github, so a clarification could be needed.

p12l298: It should be made clearer what part of ESM-Tools is hosted at Github (and covered by the GPLv2 license?) and what "development access" means. It seems that the software at Github comprises mostly the configurations for models/components and platforms, but not the actual code.

Technical corrections

p7f1: The upper part of the figure is about the build process, but I'm not sure if the "esm_master get-model" command is consistent with that. I would expect something like "esm_master comp-model" or similar. But that's without knowing the ESM-Tool commands.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-100, 2020.