

Geosci. Model Dev. Discuss., referee comment RC1
<https://doi.org/10.5194/gmd-2021-130-RC1>, 2021
© Author(s) 2021. This work is distributed under
the Creative Commons Attribution 4.0 License.



Comment on gmd-2021-130

Anonymous Referee #1

Referee comment on "Harmonized Emissions Component (HEMCO) 3.0 as a versatile emissions component for atmospheric models: application in the GEOS-Chem, NASA GEOS, WRF-GC, CESM2, NOAA GEFS-Aerosol, and NOAA UFS models" by Haipeng Lin et al., Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2021-130-RC1>, 2021

I have read the paper "Harmonized Emissions Component (HEMCO) 3.0 as a versatile emissions component for atmospheric models: application in the GEOS-Chem, NASA GEOS, WRF-GC, CESM2, NOAA GEFS- Aerosol, and NOAA UFS models" by Lin et al. The paper is well written and described the HEMCO component well. HEMCO 3.0 looks to be much more flexible and useful for a wider range of models than the previous version.

A few specific comments.

One point to clarify, can scaling be spatially-, temporally, or sector-explicit? (for example, sector-specific, or regionally-specific NO_x -> NO₂/NO₂ scaling. Can this change over time?) This seems to be implied later in the text, but should be mentioned more explicitly.

There should be a software implementation section in the paper that gives an overview of how the software is implemented and structured. It appears the language is Fortran90, but this is not mentioned anywhere in the text. It would be useful to provide a few paragraphs that give an overview of how the code is structured at the "nuts and bolts" level. How do the HEMCO components communicate with each other? Are there common blocks of data that are key to HEMCO (or have more modern software encapsulation and abstraction techniques been used?)

While the paper gives a useful overview of the way HEMCO is used, a reader comes away from the paper with no idea of how to actually use HEMCO? For example, is it a matter of connecting to just a few key interface routines?

As part of such a section, it would be very useful to provide some details on how a user would add an extension. This is a potentially very powerful feature of the system, so more details on that aspect would be useful. Give an indication of how much work someone would have to do to implement a new extension. It would also be useful to point to some examples in the current code base to extensions implemented in the current system.