

# Reply to referee #1

I thank anonymous referee #1 for his/her comments which have improved the manuscript. New manuscript text is italicized in the replies.

**Comment:** The abstract and first paragraph end with ellipses (!), there are abundant grammatical errors and typos, oftentimes the language used is casual (e.g., “I will just list rapidly the most important ones”) and numerous statements are strikingly vague (e.g., “the tool can be used in 0D and 1D mode, with schemes coming from different models and with different time-advance methods to produce different kind of plots”; “it led to problems maybe specific to our environment or source code”).

**Response:** The manuscript has been corrected by a native writer of English.

**Comment:** The content of the paper encompasses atmospheric modelling techniques and technicalities of Python-Fortran interfacing. None of these two subjects are covered in sufficient detail in my opinion (perhaps focusing on just one of those would be a path forward)? Noteworthy, already the title of the paper implies description of improvements to a particular microphysical scheme. These improvements, described in section 3.1, are presented as purely textual description with vague statements, e.g. „the graupel growth mode choice was updated. It is now more continuous and, hence, less time-step dependent”. Such approach does not match GMD’s standards aimed at clarity with respect to model formulation and versioning. The author does explicitly state that the “purpose of the paper is not to give an extended review of all the modifications”, yet in my opinion the way the model development is documented in the paper goes against GMD policies.

**Response:** The goal of the paper is to present the 0D tool. The ICE microphysical scheme and the modifications that have been applied on it are only there to illustrate the PPPY behaviours. To make it more clear, I suppressed the name of the scheme from the title, the abstract is somewhat rewritten and the improvements to the ICE scheme have been moved to an appendix to suppress them from the manuscript body.

Moreover, some details have been added in the section 2.1.2, in addition to the ctypesForFortran details in section 2.1.1 (in response to your comment below), to improve the OD presentation.

**Comment:** The Python-Fortran interfacing subject, covered in section 2, is presented with similar level of vagueness. The key components of the presented software included in the 11k LOC ctypesForFortran.py file are not discussed at all. Overall, I expect that independent use of the presented PPPY package, would not be easier than obtaining analogous functionality “from scratch” using a general-purpose Python package providing abstractions for interfacing compiled code (e.g., CFFI which has numerous documented examples depicting its usage with Fortran code and NumPy arrays).

**Response:** I do not consider that ctypesForFortran is the key component of the software because PPPY users can use ctypesForFortran, f2py, directly ctypes or another tool such as CFFI. However, I included a short description of the main features ctypesForFortran includes:

*The PPPY user is free to use whichever Python-Fortran interfacing method he chooses (among the two aforementioned or other ones). The ctypesForFortran way intends to help the interfacing of Fortran functions and subroutines on a Linux system. It handles memory allocations and array memory order. Internally ctypesForFortran uses the Python ctypes module (which normally handles the C shared libraries) to interact with the library without*

*adding a C or Fortran layer. It deals with Boolean, strings, integers and floats (32- and 64-bits) but does not support structures. The array and string arguments must be explicitly defined (no ``:`, ``." or ``\*" are allowed in the interfaces) and no argument can be optional. If this is not the case, a wrapper must be written in Fortran meeting these requirements and calling for the original subroutine.*

In addition, to be more concrete, I included an annexe to give an example of PPPY usage.

When we encountered problems with f2py, we found easier to bypass these problems by writing the ctypesForFortran module that we can control. Maybe it exists a universal interfacing tool which is suitable in all circumstances but a rapid test shows that CFFI also brings problems concerning boolean scalars with some compilers (the binary representation of a Boolean scalar with Fortran is different depending on the compiler, eg. intel vs gfortran). I expect that this problem can be solved with a Fortran 2003 compatible compiler using "bind(c)" but this was not an option when we wrote ctypesForFortran.

**Comment:** Although, in principle, I would be reluctant to call something "too basic", reading the manuscript I felt puzzled with regard to the intended audience of the paper. I feel confident that GMD readers do not require repeated verbose explanations on what numerical diffusion is and why it vanishes for integer Courant numbers. The same concerns such statements as: „Python was chosen because it allows to make plots ...", "the computational time can be large when very small time steps are used" or "One process must take into account that a given specie can be consumed or produced, in the same time, by another process".

**Response:**

- For the verbose explanations on numerical diffusion: Sect. 3.2 have been rewritten
- For the python choice: I'm sure many readers know that Python can make plots but what is important is that a **single** language can produce a plot **and** interact with a compiled code. In the revised manuscript, two sentences have been merge to be more concise and to not appear to be too basic:

*The tool consists of a Python package which drives the simulations and performs the comparison: initialization, the calling of the Fortran routines (using the original source code of the parametrization), the saving of the results (in HDF5 files using the h5py module) and the plotting of the results (through the matplotlib module).*

- The remark about the computational time that can be large despite of being in a 0D mode is suppressed.
- For the interaction between processes, this can appear to be too basic but it is important to mention it because 1) this interaction is not taken into account in a number of microphysical schemes (except by preventing negative values for the hydrometeors) and this induces an uncertainty on the results, and 2) this is the reason why the splitting was introduced in the ICE scheme. I slightly reordered the sentences to exhibit more the relation between the interactions and the splitting:

*The modifications listed above aim at suppressing the time-step dependency present inside each of the microphysical processes. These modifications were sufficient to suppress or, at least, limit the dependency until time steps around 10 s (not shown). For greater time steps, each process must take into account that a given species can be consumed or produced, at the same time, by another process and that, therefore, this affects its efficiency. To address this issue, some kind of splitting was needed to reduce the effective time step used in the microphysical scheme.*

**Comment:** Below, I am listing some more specific comments that perhaps can be helpful for the author, and that support my opinion outlined above:

- avoid frequent use of the word "tool" (over 40 occurrences including all but one sentences

of the abstract)

- avoid ellipses
- do not use programming notation such as “1.E-5” in the text
- time step vs. time-step, etc - please be consistent
- ensure the use of the words “statistical” and “physical” is justified for all its occurrences
- please do not call something “classical” without reason
- following phrases have certainly better alternatives: “home made”, “some behaviors of a scheme”, “object made from a class”, “intensity of the 0D simulations”, “more the content is important, more the fall is rapid”, “weak content”, “leads to do approximately the same computation”, “content is artificially put higher”
- capitalise Python
- use vector graphics for figures
- ensure consistency in bibliographic entries: abbreviated (with dots or without) and non-abbreviated journal names

**Response:** I did my best to take into account your remarks. Some of them need a specific reply:

- 1.E-5 was an error, this is corrected in the text
- The manuscript has been corrected by an English native writer
- The graphics are outputs from PPPY, they are png files