Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-358-RC1, 2020

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on "Flex_extract v7.1 – A software to retrieve and prepare ECMWF data for use in FLEXPART" *by* Anne Philipp et al.

Anonymous Referee #1

Received and published: 19 May 2020

The manuscript 'Flex_extract v7.1 - A software to retrieve and prepare ECMWF data for use in FLEXPART' describes a tool used for the pre-processing of data from various ECMWF model streams, preparing them for the use with the atmospheric transport model FLEXPART. The tool builds on a long history going back to a loose script collection that was previously used for FLEPXART data retrieval. The manuscript and the software are a great opportunity to bring to light some of the somewhat obscure pre-processing steps that were applied by many groups using FLEXPART and were based on unpublished descriptions of the pre-processing. However, the manuscript could be significantly improved in terms of clarity and detail. Currently, it is quite difficult to use it as a user's manual. Detailed comments are provided below. Next to reviewing the manuscript I followed the installation and use instructions and took some peeks at the

underlying code. Some comments concerning this are provided below. However, these experiences/tests are far from complete and I feel further feedback by other test users would be very beneficial. In summary, I suggest major revisions to the manuscript (as outlined below) and suggest that the authors also consider comments concerning the code before this work can be published in GMD.

Major Comments

1) Type of manuscript The manuscript was submitted as a 'model description paper'. However, it does not describe a geoscientific model as such, but a pre-processor for the FLEXPART transport model. As such I strongly feel it should rather be in the category 'development and technical papers'.

2) Paper structure

Overall the manuscript is too long. Considering that this is 'only' a pre-processor and only few scientific methods are described, it should be possible to present the material in a more concise way, focussing more on the application side of things.

First of all, the manuscript should more clearly distinguish between methods/scientific background and the application/user's manual section in the appendix. Right now there seems to be too much of a mix between the two, especially in section 3. Here are my suggestions for improving the flow of the main manuscript. Section 1: Start with description of FLEXPART/FLEXTRA and the need for appropriate input data. Then the history of flex_extract. Wherever possible move historical information from sections below into this section. Section 2: This section is very valuable, especially the provided Tables summarising the ECMWF data availability, which is cumbersome to obtain from ECMWF/MARS itself. Section 3: Especially convoluted part between methods and application. Large parts of section 3 especially sub-sections 3.4 and 3.5.x are focussing only on application and should be moved to appendix. Section 4: Very valuable, but could have clear description of settings for recommended best practices. Section 5: Although, documenting, testing and benchmarking of a code are as important as the code itself, I suggest to significantly shorten this section and put the details into supplementary material. I can see that a lot of work went into this part, but for most readers/users this information is less relevant and of no direct consequence. Appendix: This part should work as a user's manual. A such I think the structure would work better if those parts that differ between application modes would be organised by application mode and not by different installation/application steps. This would allow a user, who will usually be running the software in only one of the modes, to follow along without having to jump from section to section.

3) Code structure and maintenance

As stated by the authors themselves, flex_extract is a collection of shell scripts and python scripts and FORTRAN code. As such there seem to be many possibilities to break the code in case external dependencies change. I am wondering if it would not be a much cleaner approach to provide this tool as a single python package. This should make maintenance, documentation and installation a lot easier. Python codes are able to work with external FORTRAN codes as well. So it would still be possible to make use of the arguably faster computations done in the FORTRAN code. But there seems to be little need for mixing in korn shell scripts (other than for job execution on ECMWF servers). Some interesting information in this respect can be found here: https://packaging.python.org/tutorials/packaging-projects/ https://www.numfys.net/howto/F2PY/ This is more a comment about the future of the code and may not be achieved quickly as part of a revision of this submission. It would also be beneficial to host the code on a more commonly used platform like github or gitlab. I was not able to git clone the code from the address given in the manuscript and would not know how to provide code feedbacks in form of merge/feature requests on the platform that is currently used for hosting the code.

Minor Comments

P2,L39: 'non-authorised' is a bit strange. Users still need to be registered and as such

are authorised to use the system. I would rather call this non-member state users.

Version number, title and first mention P2,L44: If a totally revised version was created that is fundamentally different from the previous versions and as such not compatible with, for example, previous config files, it may be more appropriate to express this by incrementing the major version number. So why not go for 8.0?

P3,L70f: New version of what? I guess flex_extract, but could also refer to FLEXPART. There is also no clear mentioning above about how the first version looked like.

P3,L85: Also mention what happened to the FORTRAN part.

P4,L100: 'most accurate data source'. That's a bit general. So ECMWF is better than any other atmospheric model? I would not go that far, but if you do you should link this comment to some kind of model evaluation that provides a similar conclusion.

P4,L103: Last sentence can be removed, should be clear from context.

P4,L108: I would not say that they share the code, since these are two completely different software packages. But one could say that some of the code goes back to the same original routines. Sharing the code would mean that their would be a common specific library of routines that would be used by both models.

P4,L108: 'It ingests ...', It should be referring to FLEXTRA, but previous sentence on FLEXPART. Replace.

P4,L131: 'are to be used' better 'are available' or 'can be used'.

P4,L131: 'Flex_extract automatically ...': Not clear how flex_extract comes in at this point. So for this section was describing the different access kinds to ecmwf. It needs to be made clear here what are the access possibilities and then in a second step how flex_extract is using them. Maybe don't mention Flex_extrac in this section at all.

P5,L141: Later on this dataset/model version is referred to as DET-FC. Why not keep the ECMWF terminology.

P6,L157ff and Table 1: While the text nicely refers to the evolution in the IFS operational setup, I am missing a similar warning in Table 1. Seen on its own the Table may be misleading. Please add reference to Tables 2 and 3.

P6,L177: 'member-state' instead of 'member'.

P6,L178: 'not good'. Can you more specific what is not good about? Not robust/reliable or simply (too) slow?

P7,L184ff: The use of access and applications modes is a bit confusing here. Especially when the text refers first to access and application and then the combination of these is once more called application mode. Maybe the primary application modes could be described more clearly as 'execution location'?

P7,L213: 'module system': Rather call it 'environment modules framework'.

Figure 1: Could this not be drawn much simpler without the emphasized distinction between the two different web APIs? The main point of this schematic should be an illustration of the 4 application modes. To user it does not matter so much that there are two different web APIs.

P8,L231: Why not mention this python package in Table 7 as well?

P9,L254f: In the schematic (Figure 4) the "remote mode" is on top. So why start with explaining the "gateway mode" first?

Figure 4: The schematic may be easier to comprehend if ECMWF and local machines would always be on one side. Also the Labels for the different modes should be given rather on top with the sub-panel index. Right now the may be mistaken as only referring to the boxes on the right.

P9,L256: Once again the use of he different modes is confusing. While describing the 'gateway mode' it is mentioned that flex_extract is started in 'local mode'? Confusing. If I understand correctly, it is started locally to generate the ECMWF scripts and then

sends them through the gateway. 'local mode', however, is not using the gateway but the WebAPIs. Correct? The same confusion is created in the caption to Figure 2, where 'local mode' is mixed into both gateway and remote mode.

Figure 2: What is the purpose of the 'do local work' branch of the flow chart?

Figure 3: Again, I am not sure that I understand the purpose of the two side-branches 'submit' and 'PRINTING MARS_REQUEST'.

Section 3.4.1 and Tables 8 and 9: I feel that these tables include the core information of how to setup and run the data extraction correctly. As such I think the description of the individual parameters should be given more space in the main text (as part of the Appendix) and not only in the supplement. This is crucial information for using flex_extract!

P14,L340: 'via the gateway server': not in 'remote mode'!

P14,L351: Do these modules correspond to the according boxes in the flowchart (Fig3)?

P15,L363: Who is Paul James. Is there a reference to this work?

P16,L390ff: At this stage it is not clear how these additional inputs are treated. Will there be separate output files for these times? How does FLEXPART deal with these? From which FLEXPART version onwards is this feature supported?

P16,L394: Maybe it would be useful to explain briefly why other fluxes are treated differently from precipitation.

P18,L453: grib_api seems to removed from all the makefiles and is also not mentioned in the dependencies (Table 7).

P18,L456: Usually one would refer to these versions as optimised and debug versions, not fast and debug. It would also be better to have a single makefile and only use a switch in the makefile to create either of the desired executables.

P19,L483: Does the comment in braces mean, that no two instances of the submit.py script should ever be running at the same time? Where do you expect clashes if done anyway.

P20,L530: What about forecasts longer than 99 hours. How will they be written into a 2 digit number?

P22,L577: This is comment that cannot be fixed in the manuscript, but should be addressed in the future. I assume you are referring to the Global Forecast System of NCEP here? If the availability of model level data is a shortcoming to the community. NCEP should be approached to make model levels available as well. Internally GFS operates on a similar hybrid sigma-pressure coordinate system as ECMWF IFS.

Section 4: The section gives many valuable hints for best practices. However, I would find it very useful if these could be manifested in more concrete instructions, maybe in form of clear example CONTROL files. For example it is mentioned that only a subset of levels can be extracted, but how is this done concretely and does it work for all etadot modes (didn't in the past)?

P28,L746f: Actually one does not need any code for this task. It can be performed using standard bash cat command. That is one of the few beauties of grib format.

P28,L752ff: See also previous comment on package structure. The possibility of calculating etadot should not just be discarded, but there certainly are better ways of integrating the FORTRAN code into the python package.

P28,L755f: Again as mentioned above, instead of just providing a tar ball, a public git repository that would allow for swift user feedbacks and contributions would be beneficial.

P29,L790: Better refer to this as dependencies or prerequisites.

Code sections in Appendix (e.g., P31,L850ff): It would be easier if such sections would be directly given as example config files or scripts. Copy and paste from a manuscript

is rather cumbersome and error-prone.

P34,L994: Does testing CERA-20C refer to CONTROL_CERA or is there another prepared example to test it. Please mention.

P35,L103: This unload was not mentioned above for the installation process. Is it necessary? Because once you unload python on ecgate the python command is linked to the system pyhton version 2.6! So I don't see the gain of the unload.

Table 7: It is mentioned in the text that compilers other than GNU compilers were tested as well. This should be reflected in the table. emoslib should just be emos. The information for fortan/pyhton is not required since this is given by the header line.

Table 11: With files job.temp and job.template it is difficult to distinguish their function just from file name. The second one should maybe be renamed to install.job.template Why does the first template not have the template file name extension. Is it treated in a different way than the other templates? Same for convert.nl. Even if the output of the template processing is a namelist shouldn't it have the same template ending.

Table 20: Is this table really needed? It is only a list of file names without further description. I can obtain this also by looking into the corresponding folder! Either remove or add a meaningful description of the listed files. The caption alone does not explain the details.

Technical Comments on Manuscript

P3,L83/84: Past tense in previous sentence. Why not here?

P4,L106: Style. Twice representing in same sentence.

P5,L146: Citation style for Laloyaux et al. (2018)

P6,L181: 'necessary' not 'ecessary'

P16,L387: Citation format of '(Hittmeir et al., 2018, p. 2513)'. No pages in journal reference.

P17,L414f: So in the end FLEXPART converts etadot (or etapoint) back to a Cartesian vertical velocity? So why the painful calculation of etadot in the first place? Only because of the historic development from FLEXTRA?

P17,L416: 'pressure' not 'perssure'

Code Comments

As stated above these are some comments on code and on my personal experience running the code in 'remote mode' on ECWMF's ecgate server. Some of the remarks document that the ease of using the code should still be improved before final publication of this manuscript. Other comments can be seen as recommendations for the future.

The FORTRAN code works with default FORTAN unit names for I/O (see table 17). This seems a bit outdated and difficult to follow/debug/etc. Please change to meaningful filenames wherever possible. This would allow removing Table 17 from the manuscript.

Besides the documentation in the main routine the comments in the FORTRAN code are mostly in German. This should be changed in future submissions.

Section 3.8.3: And why would the FORTRAN program need to save these variables to a file instead of storing them internally? Makes little sense if they are not used again, for example, by the python scripts.!

Installation process: I only tested the remote mode on ecgate. My experience was not as straightforward as the description may suggest and as one would expect on a system the code should have been tested on. Since the remote installation is the one installation mode that is running on a defined system it should be easier to get this done without having to change various parameters in the setup script. First of all, I tried the git clone command on page 28. However, this does not clone flex_extract, but FLEXPART!? In the following I could not figure out the correct git address for

flex_extract and resigned to to downloading the tar ball instead. Once unpacked, I was able to proceed. However, I don't see any reason why the installation process is submitted to the queueing system on ecgate. We are talking about a very small compile task here. It would be more simply done on the login nodes so that one has a more direct feedback when things go wrong. Anyway, first email I get back from the submitted install job was a very general compile error ("No rule to make target 'phgrreal.f', needed by 'phgrreal.o'. Stop"). So I checked if I selected the correct makefile. Yes, one would think: makefile_ecgate (the one system where this should work. Right?). Taking a look into makefile_ecgate shows that file extensions seem to have changed from f to f90, but this was not adjusted in the makefile. So I modified the makefile. Second try. Got an email which may indicate success of the FORTRAN compile, but does not tell me anything else about the rest of the installation. So let's see if it is there. Hm. Not in the directory I had specified in $INSTALLDIR. But there is a new folder flext_extract_v7.1 in my home. OK. Fine. Let's test it. So I follow the instructions in Appendix A4. However, the executable mentioned there is called calc_etadot_fast.out, while in my installation it is simply called calc_etadot. No problem. But shouldn't the exe name be harmonised in the makefiles? I did not change it. Running the executable seems to work but in my case only produced 2 lines of output in contrast to 7 mentioned in Appendix A4. Once again, did I do something wrong. Did it really work? There are 2 output files in the directory now and one can assume that that's sufficient. grib_ls on fort.15 (the obvious output file) at least shows some etadot fields. However, I was still worried about the missing lines of output. So I quickly searched the FORTRAN code for where these messages should be produced. I found them but they are commented out. So no wonder that I did not get these messages! But why are they given in the instructions.

Test retrieval: A test retrieval using the CONTROL_CERA example worked without much trouble. I would only suggest that the loading of the python3 module should be added to the run.sh script.

EcFlexpart.py: As far as I can see this is where the ectrans command is used to transfer the data. The ectrans command is used in a way that assumes that it would crash if an invalid transfer was submitted. But this is not how ectrans works. Much like a queuing system ectrans submits the requested file transfer into a queue returning a request ID. The transfer request will then be repeated if not successful at the first try. So if the settings for source and target file are not correct ectrans will still return a request ID and try the transfer until it reaches its repeat limit. Hence, it is not possible to simply capture the return code from ectrans to detect an error.

Work folders on $SCRATCH are called python<PID>. Why not use something that would clearly identify these as flex_extract folders?

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-358, 2020.