After the end of the discussion phase we like to thank the referees again for their helpful comments. From both reviews it is clear, that we failed to clearly state our motivation and the part of technical model developement that we want to publish in this article. For clarification and in order to introduce the additional information requested by the referees,

- the introduction was completely rewritten,
- a subsection shortly introducing the MESSy system was added,
- the subsections about GRID and IMPORT(_GRID) have been restructured and additional text was added, and
- the "Summary" section was extended to a "Summary and Outlook" section.

In the following we repeat the statements of the reviewers and our answers from the discussions phase. Additionally, the revisions initiated by the respective comment are indicated in blue.

We hope that the revisions meet the referees requests.

Best regards,
Astrid Kerkweg

Review #1:

This paper describes a new component of the MESSy model that aims at improving the reading and preprocessing of gridded data by ensuring a single entry point and offering common grid processing functionality. This topic is very relevant to the users of MESSy and could also be of interest to others models dealing with gridded data. Unfortunately, the paper suffers from several major flaws. First, it requires English editing and specially for the first half that is sometimes a little hard to understand (for example the introduction on page 8609). Some of the vocabulary is a little surprising (on page 8610, line 18, what is an "abstract time series"?).
Fortunately, GMD articles are copy-edited upon final publication. Nevertheless, we will check again the manuscript w.r.t. the language before submitting a revised version. The "abstract" refers to "data", not to time series. In order to avoid this misunderstanding and because data is always abstract, we remove the "abstract" in the revised manuscript. We removed the word "abstract". As the introduction was rewritten completely, the "introduction on page 8609" was completely revised.

The paper also relies heavily on concepts, vocabulary as well as acronyms of the MESSy community making it quite obscure outside this community (see for example page 8623) or the lack of even a brief definition of what MESSy is). Quite a few of these acronyms are defined in the text, but not all and generally after being introduced.
We will wade through the text and make sure that really all acronyms are introduced the first time they are mentioned. Moreover, we will add a short review on what is special about the concepts of MESSy and we will make clear that the names of the MESSy submodels are written in capital letters even though they are no acronyms.

We added the new section 2 introducing shortly the MESSy interface. We checked for all acronyms and added links to the model web pages, were model names/acronyms are used. The names of the MESSy submodels are following the typical MESSy notation, written in capital letters, however, they are no acronyms.

Overall, the paper lacks clarity.

A more precise statement would help to improve it. Nevertheless, we will try to identify potentially unclear passages and improve the text accordingly.

To gain clarity, we rewrote and restructured the introduction section completely using bullet points list to yield an easily to conceive structure.

The paper also lacks structure: some details are given in what should be the general introduction to a new section while important concepts of the general infrastructure are not provided until several paragraphs after being first used (for example the NREGRID and SCRIP third party modules are mentioned multiple times on pages 8609, 8610, 8611 and finally briefly defined on page 8612 but without enough details to show what are the key differences between the two).

The most important differences are already listed in the introduction on page 8610. SCRIP provides "transformations to/from curvi-linear or unstructured grids." We will rewrite the sentence to clarify that NREGRID can not handle curvi-linear and unstructured grids and make sure that additional information is introduced as early as possible.

In the end, we rewrote the introduction section and restructured parts of the GRID and IMPORT sections taking also comments of referee #2 into account.

Some sections should be merged together (the introduction to section 3 is mostly a rephrase of things written previously,[...]

The introduction is not that long and we wanted to emphasise the point, that IMPORT constitutes one single point of data input, while CHANNEL one single point of data output. Nevertheless, we will try to shorten the introduction.

Finally, the introduction to section 3 (now 4) is now even longer than in the discussions version. However, the points made in the introduction to section 3 are now really new and not discussed in previous sections.

[...] the whole section 4 should be condensed in a few sentences and merged into the introduction).

Here we disagree with the referee. It was one important intention of the article to also document the history of data import in MESSy. As will become clear when reading Section 4 the emission and deposition submodels have been published under certain names (ONLEM, OFFLEM and DRYDEP, respectively) still performing individually the data import. In more recent MESSy articles using the new emission and deposition models (ONEMIS, OFFEMIS and DDEP) with the "out-sourced" data import, we had to argue why these submodels are basically still the same as the ones published, but are named differently. In future it would be nice to have a citable publication for this submodel renaming.

For the reasons given above, we kept section 4 (in the revised version it is section 5).

On the other hand, some details are missing: for example the programing language that has been used is not even mentioned.

As MESSy always employs the programming language Fortran90/95 we forgot to mention it explicitly in the article. We will add this information. It would be good to know, what the other missing details are. We did not omit them by intention. Yet, we will re-check.

We added the information about the programming language in the new section 2 about the MESSy interface. We hope that this new chapter provides all the information referee #1 was missing in the discussions version.

The vocabulary is also not very consistent with different names for the same ideas in different sections ("grid routines" vs. "mapping routines" vs. "mapping algorithms" for example) and not defined when this would be needed (what does this mapping means? Is it not a reprojection of the grid? This is not very clear outside this community.)
The phrase "grid routines" does not occur in the article. The only phrase that is near to it, is "grid handling routines". Actually, we thought it would be understandable, that these are the routines required to work with or on the grid structures, e.g. comparison of defined grids etc. . It will be clarified in the revised manuscript. The words "regridding", "remapping", "grid transformation" and "mapping" are indeed used as synonyms (we will state so in the revised manuscript), as we see for this application no big difference between them and tried to avoid tedious repetition of the same word over and over again. "Mapping", as a contrast to "interpolation" calculates the overlap between individual grid cells and calculates the data field on the target grid by summing up the individual contributions of the source grid cells overlapping with a target grid cell weighted with the overlapping area of the single grid cells. This will also be clarified.
The decription of this terminology was added to Sect.3.2 (GRID_TRAFO).

Finally, this paper fails to demonstrate the originality of the work. The results that are presented have actually been produced by third party modules and it seems that the work presented here mostly consists of a wrapper around these modules that fully perform the heavy duty processing.
This is true with respect to the mapping routines, which, in their core use SCRIP and NREGRID as detailed in the article. Nevertheless, the definition and handling of grids is completely new and thus the work is original. We will clarify this.
We clarify this in the introduction to section 3.

Moreover, these third party modules were already used in the past by MESSy (although it is now done in a cleaner way). if this is not the case, the authors should clearly explain it and show actual scientific content and results of their own work.
GMD is not about scientific content. It is for documenting technical model developments. Yes, IMPORT and GRID are wrappers for third party (or second party; actually NCREGRID was written by the second author of this article) code. But here, we document an important step in the development of the MESSy infrastructure. This article is part of the special issue on MESSy and GMD invites also papers documenting new development steps or updates of model parts. Thus we think this article fits very well within the scope of GMD and we do not have the show scientific results here.
The newly written introduction section now already clarifies that this is the publication of the implementation of a specific technical concept within the MESSy interface.

Therefore, although some explanations about how to use this new module are given (that are obviously only relevant to the users of this new submodel), the paper does not bring any usable information or new knowledge to the scientific community.
As stated above, GMD does not only publish papers containing new scientific knowledge,

but also documentations of the tools on which scientific knowledge is and will be based. Additionally, we object to the statement, that this is not relevant for non-MESSy-Users. The stand-alone model, which is part of the supplement and briefly described in the article can be used by everybody, independent of the other parts of the MESSy framework. Indeed, it can also be coupled to other models as well. In the past, the previously used NCREGRID stand-alone model was used outside the MESSy community. We will clarify this in the revised manuscript.
The section about the stand-alone tools now clearly states the usability of these tools outside of the other MESSy models. This is stated again in the "Summary and Outlook" section.

The authors also fail to present their work in a way that would be less dependent on MESSy, therefore restricting the applicability of their work. This is exemplified by the figures 1 and 4 that mostly show how MESSy has been restructured instead of showing how the generic pre-processing of gridded data has been improved by their approach. Outside the MESSy community, these figures are not very helpful.
As GMD welcomes model code documentation and these information are helpful for the users of MESSy, we do not see why we should remove information relevant for model users from the model documentation. Nevertheless, we will strengthen the information, that IMPORT and GRID can also be used (and how) by other users in the revised manuscript (as stated above).
Changes as listed above.

Review #2:

GRID and IMPORT should be important submodels in the infrastructure MESSY. This manuscript presents the details of GRID and IMPORT. It may be interested to MESSY developers, users, as well as the ones who develop couplers or other model infrastructures. I have already implemented similar common modules in the latest version of our software recently while it is still very difficult for me to follow the details in the manuscript because the presentation is not good or even poor. I recommend that revisions are required before this work can be published in GMD. Authors should reorganize the manuscript so as to significantly improve the presentation.

This manuscript focuses on the GRID and IMPORT submodels in the infrastructure MESSY. However, as a paper, it should not be limited to MESSY. For revision, authors should well address the following questions:

1. The key idea of GRID and IMPORT is to make part of preprocessing online in model integration.
Yes, indeed. We will add a paragraph to the introduction, stating the advantages of performing (parts of) the "classical pre-processing" on-line, thus better motivating our developments.
The rewrite of the introduction section includes a clarification on this issue.

What are the requirements of preprocessing according to the current or even future status of Earth system modelling in the world? Which requirements are considered for the design and implementation of GRID and IM-

PORT, and why? There should be some discussions about the requirements that are not included in current GRID and IMPORT. Examples are welcome for the discussion of the requirements.
This will also be addressed in the introduction of the revised manuscript. We will strengthen this discussion (including examples) and show more clearly the current status of GRID and IMPORT.
As announced above, the introduction section now comprises answers to the questions raised by the referee. Discussion of the current status and additional needs of GRID and IMPORT have been added at various places in the text. Particularly, an outlook has been added to the "Summary" section.

2. How about the related works? It may be difficult to go through all related works because engineers always do not write papers. Many models already have modules for online "preprocessing". Authors try to achieve common modules for various models. I believe that authors can quickly know whether a model have common modules according to the code or configuration system. Similar models in some well known models and infrastructures (if have) such as CESM, WRF, FMS and ESMF should be discussed and compared.
The referee is right. We hardly found any literature on data import. From what we found, mostly by asking people working with the models, and not by published model descriptions, it seems that mostly off-line pre-processing and direct import on the models grid is applied. Model couplers are discussed more often, but this is off-topic for this article. Nevertheless, we will address this issue in the introduction of the revised article.
This issue is addressed in the newly drafted introduction section.

3. The common modules in this manuscript generally focuses on online interpolation. What are the requirements of online interpolation according to the current or even future status of Earth system modelling in the world? To answer this question, various types of grids (including 3-D grids) and various remapping algorithms (or requirement for interpolation) should be discussed.
Note that "interpolation" describes only one class of grid transformation (or grid remapping) algorithms. In most applications (e.g., emission flux remapping for CCMs) conservative remapping is required. As discussed by Jöckel (2006)[1], the corresponding algorithms might differ, depending on if the quantities to be remapped are intensive or extensive. Ideally, a common module should be able to handle both and provide a variety of algorithms, which can be selected depending on the application. Our impression is that most models / couplers utilise the remapping / interpolation tools based on the SCRIP toolkit.

We will discuss this in the revised version.
This point is also discussed in the introduction.

4. About the implementation of GRID and IMPORT. Here authors should answer how to make GRID and IMPORT support various types of grids, various remapping algorithms, 3-D interpolation in parallel, and various expressions of time information.
Most of the information answering those questions are part of the supplement. Nevertheless, we will provide more on that in the revised manuscript. GRID does support various types of grids and performs grid transformations in distributed memory parallelisation,

[1]Jöckel, P.: Technical note: Recursive rediscretisation of geo-scientific data in the Modular Earth Submodel System (MESSy), Atmos. Chem. Phys., 6, 3557-3562, 2006.

depending on the domain decomposition of the used model. No interpolation in time is implemented so far, mostly for the reason that conservation constraints (e.g., time integrated flux (= mass) conservation), are difficult to achieve within time-stepping procedures, if concurrent access to the entire time series of data to be imported is to be avoided.
Additional information/discussion concerning this topic was added to introduction and to the section about GRID (now Sect. 3).

There may be some limitations in GRID and IMPORT, while authors should clearly discuss these limitations. For example, are these limitations because of the whole MESSY or other reasons, and how to solve these limitations in the future? Why and how about the design of the API and configuration format should be presented, corresponding to what are supported in GRID and IMPORT.
Indeed, GRID in IMPORT also have limitations. We will extend the manuscript by discussing those (e.g., the current lack of interpolation in time) wherever appropriate.

The last sentence above is unfortunately unclear. The API is documented in detail in the supplement.
The respective discussions have been added partly in the respective subsections and in the "Summary and Outlook" section.

5. How to use GRID and IMPORT? Some comprehensive examples are required.
The detailed supplement comprises information about (1) the usage of the stand-alone tools, and (2) how GRID can be used from within other MESSy submodels. This supplement is referred to in the manuscript. In the manuscript itself, we provide example namelists showing how to add data to be imported with IMPORT_GRID and IMPORT_TS. So, it is unclear what "comprehensive examples" should be?
As no additional information was provided from the referee's side, we kept the examples provided in the article as they were.

6. How about the performance of GRID and IMPORT, especially the scalability of parallel interpolation? How about the comparison to the offline solution? I/O should be a bottleneck for both online and offline solution. How about the performance comparison when parallel I/O is used? It is possible that the online solution and offline solution outperforms in some cases and then the hybrid solution (for example, horizontal interpolation is processed offline and vertical and time interpolation is processed in parallel online) should be much better. Authors should discuss about that.
These are indeed important issues. And yes, the I/O is a bottleneck for both, on-line and off-line, solutions. The advantages of the on-line solution are a high flexibility w.r.t. to resolution. Initial and boundary condition data needs to be stored on disk only once, and not at every model resolution. Most important, some parts of the re-mapping can hardly be pre-processed off-line. For example, it is desirable to distribute emissions in global CCMs in the vertical. This vertical distribution might depend on the actual meteorological situation, or the vertical grid (e.g., a hybrid-pressure grid) might be time dependent. An adequate off-line pre-processing is not straightforward to achieve in those cases. Separating the horizontal and vertical remapping into off-line and on-line pre-processing, respectively, might be an option, though it will complicate the approach. Nevertheless, such split is naturally possible with IMPORT, because the grid resolution of the imported data is arbitrary, implying the possibility that the horizontal grid matches the model grid.

The performance of the different approaches (off-line vs. on-line vs. hybrid) depends very much on the application, e.g., the amount of data to be imported, the applied model resolution, the parallel domain decomposition of the model and last, but not least on the computer architecture. Thus, a universal statement on performance cannot be provided. Nevertheless, we will add a short discussion about these issues in the revised manuscript.

According to the scalability: Again it depends on the overlying model, because the distributed memory parallelisation of IMPORT utilises the domain decomposition of the overlying model. Two possibilities are implemented: either one task performs the grid-transformation for the entire grid and distributes (scatters) the results, or each task performs the grid-transformation only for its own part of the entire grid. It is obvious that the first option does hardly scale, but the second option does. Again, a universal statement cannot be provided, but we will add this discussion to the revised manuscript.

As IMPORT is currently implemented, we use serial netCDF for data input. Neither parallel I/O, nor asynchronous I/O are implemented. These are promising options for the future an we will state this in the outlook section.
We added these information to the article. Particularly, we extended the subsection about parallelisation (now Sect. 4.1.3) and added a new subsection about scalability (Sect. 4.1.4).
