# Responses to the Interactive comments on "A note on precision-preserving compression of scientific data"

Rostislav Kouznetsov[1,2]

[1]Finnish Meteorological Institute, Helsinki, Finland
[2]Obukhov Institute for Atmospheric Physics, Moscow, Russia

**Correspondence:** Rostislav Kouznetsov (Rostislav.Kouznetsov@fmi.fi)

I would like to thank all the reviewers again for their valuable comments The general points of the reviewers have been replied in corresponding short replies. More detailed and specific replies are below.

## 1 Response to the Short comment by Milan Klöwer

In addition to the response already given to the comment SC1, the following changes have been added.

*I suggest to discuss how your method is different from round-to-nearest tie-to-even which is already mentioned in IEEE-754 from 1985 as "An implementation of this standard shall provide round to nearest as the default rounding mode.*

**Response:** The description of integer-arithmetic rounding has been included together with an example of error accumulation due to half-to-infinity rounding method. Implementations of the standard half-to-even rounding both in Python and in high-level Fortran 95 are now included in the appendix.

## 2 Response to the Reviewers comment by Mario Acosta

...

*How should the rounding bit size would be decided? Is this a parameter so set up through NetCDF?*

**Response:** Conclusion has been extended to explicitly clarify that precision trimming can be used to facilitate NetCDF compression, while it is applicable to any format that stores floating-point data. Also a note on the precision control via NCO has been added.

*Will the precision depend on the particular application or not? Apart from the array examples presented, the author should explain more in detail this.*

**Response:** The precision needed to store the data depends on both the data the intended application. A section with discussion on the precision needed for specific fields added.

*line 14: Consider 32 bits as float can be confused since this depends on the program language. Actually, some languages define float as double precision, using integer (int) for this declaration. If a more in detail explanation is not done, I would remove float and double*

**Response:** Removed

*lines 17-23: Have you consider different scientific fields to affirm that level of accuracy (less than 7 bits) or entropy? I agree that in most cases single precision is more than enough but you should enumerate cases where this could not be true and provide a state of the art about this. As an example, I consider that in the inputs for data assimilation of chaotic applications, as weather models, the level of accuracy or entropy to ensure the number of bits used could affect the results*

**Response:** A section with real-world examples has been added. In particular, there are examples where a precision beyond the verifiable range is needed, and an example where virtually any precision would be sufficient, but the size is critical.

Concerning the inputs for the data assimilation, I would argue that if a small variation of the assimilation input (below the accuracy of the input data) significantly affects the results, there must be something wrong with the modelling setup.

*Introduction: I consider that the state of the art should be extended. In order to support the argument explained in the second paragraph, some successful examples should prove that the reduction of precision does not affect to the accuracy. Both of them should be listed, applications using the reduction of precision and the reduction of precision to save data.*

**Response:** The introduction was extended with summary of commonly-used lossy compression methods. Successful examples might be misleading, since there are also examples of the opposite. Instead a section with specific cases was added to illustrate that acceptable distortion strongly depend on the data and the applications.

*Introduction: Is precision-trimming the single compression method available? Apart from a default method, is it the most used or one of the best methods for Netcdf files? The author should explain why this particular method and why the comparison done should be interesting for the reader, including some state of the art about this (apart from the single reference provided).*

**Response:** A description of Linear Packing, the most common to-date lossy compression method, added to the introduction, and a section illustrating drawbacks of LP added. Besides that a note on a modification of precision-trimming to keep absolute precision added.

*The mantissa-rounding technique explained here should highlight if it is novel or there are other works which present similar approaches, or at least, differences among them.*

**Response:** I am not aware of any publications on mantissa-rounding, but I would be very surprised if no one used mantissa-rounding technique before. It has been implemented in the Silam model for several years already, without considering it worth a publication. The motivation for the current publication was the need to explore the features of precision-trimming, and the fact that a very popular software uses Bit Grooming that has no advantages over the rounding and introduces unnecessary distortions.

*Section 2 should be rewritten. Second paragraph is difficult to follow and the explanation about the problem of trimming LSBs difficult to understand.*

**Response:** Section was modified. Hopefully the problem of trimming LSBs is more clear now. Besides that a simpler implementation of rounding is used, and an importance of half-to-even rounding highlighted.

*Five methods are described but only three of them are commented and only two discussed in details without more explanations about the reasons.*

**Response:** The reason has been explained and a short comment added.

*It is also not clear how these results in Figure 1 are obtained.*

**Response:** Figure 1 was generated by the script from Supplementary material. The note added to the "Code availability" section. *Moreover, it is not clear if these results and the explanation given is coming from Zender (2016) paper or from Section 3 and 4 of this paper. The author should highlight the novel contribution of this paper for the different sections.*

**Response:** The paragraph re-phrased to more clearly distinguish the novel contribution.

*As minor details about the figures, X and Y axes for all of them do not contain units. Figure 2: a) and b) are difficult to see.*

**Response:** X and Y have same arbitrary units. Axis titles added, fonts increased.

## 3 Response to the Reviewers comment by Seth McGinnis

*. . . The subject is a little dry; I think the paper would benefit from a brief discussion in the Introduction of the real-world use-case that led to the discovery of the two-point distortion, which would help the reader to follow the analysis using the structure function and understand why the issue matters.*

**Response:** A note on the stumbling on the issue of structure functions added to the introduction.

*The figure placement is a bit off. . . .*

**Response:** Should get better now.

*Figures 3 and 4 need axis labels. I would also recommend placing the legend outside the panels*

**Response:** The labels added. The legend shortened, so it does not overlap with curves.

## 4 Response to the Reviewers comment by Ananda Kumar Das

*. . . A little bit of description may be added in the introduction to establish the need of such a method. As the same has been mentioned by other reviewers, the elaboration on this point is skipped here.*

**Response:** A paragraph justifying the need for lossy compression added to the introduction.

*The methodology part of the paper which describes different precision algorithm based on the static specification of the bits of mantissa is lacking a little visibility without schematics through diagram. The diagram showing distribution of values in the 32 bits may be schematically presented for a broad class of readers.*

**Response:** A diagram added.

*If within the scope of the paper, the use of the specified method within NCO for real data may be tested and test result statistics may be added in the appendix.*

**Response:** Any example of a successful application of specific trimming parameters would allow for another realistic example when the same parameters were suboptimal for slightly different application or data, or even led to the data loss.

The intended topic of the paper is somewhat wider than the improvement and testing of NCO. For specific uses of NCO it is better to refer to the NCO documentation. While NCO has a means to control the precision, it can be done much finer: at low precisions every next tail-bit brings noticeable size reduction, which can be successfully utilized if a user clearly understands gains and losses of the precision-trimming, understands the needed precision, and has a fine-grain control over it. Therefore, the paper has been extended with illustrations of the features of precision-trimmed data including their compressibility, and with a discussion of needed precision and trimming methods for several real-world fields and applications.

I hope, it is a better alternative.