

Interactive comment on “pygeodyn 1.0.0: a Python package for geomagnetic data assimilation” by Loïc Huder et al.

Ciaran Beggan (Referee)

ciar@bgs.ac.uk

Received and published: 8 May 2019

This manuscript by Huder, Gillet and Tholland explains the Python code for accessing and executing the code for their geodynamo data assimilation code. The scientific work based on this code has been published in previous papers (Barrois et al, 2017/2018, for example) and so it is to be applauded that they have made the great effort to allow others to reproduce their scientific research in an open manner.

The visualisation tools on the webgeodyn site are also very impressive, easy to use and simple to understand and almost deserve a paper on their own merit. I spent a long time examining the different plots and combinations of data that can be placed together. Figure 2 is particularly stunning (on the website, it can be rotated and animated).

C1

The manuscript is probably of most use to researchers interested in the details of how to recreate data assimilation within the scope of the geodynamo snapshots though the authors have made it clear that a user can access the code at the different levels of expertise required. In reality, it is a companion paper to the Barrois et al and Aubert papers of the past five years. The parameters and assumptions are well explained and the flow of the work load is clear.

I have no major comments or suggestions to make. I would however, suggest that the authors revise their frequent use of the phrase 'allows to' which appears several times. For example on page 7, line 15, they could write: 'The fifth group allows the user to ...' or on page 8, line 1: '... states that allow an estimate of the background states ...'.

Minor corrections:

Page 1, line 9: generated by motion of the liquid

Page 1, line 14: that DNS are not capable of reproducing changes

Page 3, line 24: but wants to run their own data. (and again on Line 27)

Page 4, line 11: takes as input? (rather than basis)

Page 5, line 8: consists of time-stepping

Table 1: why is the default -m parameter equal to 10 in the Table but in the text you recommend 20?

Page 7, line 4: I'm slightly confused about the decimal representation explanation - surely the input magnetic or SV data are monthly means (or 4 monthly VO) so it hardly matters how the date is represented at 64-bit precision - we're not at the microsecond level of precision. It's OK if that is the default Python class to use.

Figure 1: Is it better to have runtime on a log scale to emphasis the point?

Page 10, line 5: output files are directly

C2

Acknowledgement: Perhaps add a note about the use of INTERMAGNET data (which I assume are used for the plots?)

Interactive comment on Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2019-82>, 2019.