Reply on RC1

Sophy Oliver et al.

Author comment on "A derivative-free optimisation method for global ocean biogeochemical models" by Sophy Oliver et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-175-AC1, 2021

Hello,

We thank the reviewer very much for their comments and corrections.

Regarding the 5 technical corrections, we thank the reviewer for these, we agree with them all and will edit accordingly.

- "The CMA-ES subsubsection and the DFO-LS subsection have different depth levels (2.3.1 and 2.4). Using 2.3.1 and 2.3.2 (or 2.4 and 2.5) would be more consistent."

  We thank the reviewer, and will change them to 2.4 and 2.5.

- "The sentence beginning in line 157 seems incomplete. Is there a word missing?"

  We thank the reviewer, and we agree, there were some words missing. The sentence beginning line 157 will now read "Real oceanic observations have a degree of uncertainty associated with them due to spatio-temporal oceanic processes, e.g., from small scale processes such as eddies."

- "I would remove the first three words "Results table of" from the captions of Tables 3 and 4."

  We agree and will remove "Results table of".

- "In Figure 4 the parameter boundary lines are not "red dotted" as stated in the caption but black thin lines."

  We thank the reviewer for highlighting this error, and will correct this to "thin black lines".

- "Line 320: "verses" => "versus""

We shall do this.

Regarding the two questions, we have replied to each below.

- "Did you initially work without a partition into 27 biomes and 3 tracers?"

  That is an interesting question. Essentially, because of how DFO-LS works, no we didn't start off without partitioning the misfit into regions and tracers. DFO-LS requires an input vector of misfit terms. When DFO-LS uses these terms to create the quadratic approximation, it solves a system of linear equations, which would become underdetermined if the misfit vector length were less than the number of parameters being optimised. This is not a problem for DFOLS - which then uses a Tickhonov-like regularization to the ensuing inverse problem. Still, as we are trying to perform parameter tuning, it is better, if possible, to provide DFO-LS with at least $n+1$ misfit terms (n=number of parameters) as this gives the code more problem information to exploit. For the upper limit, there is not really a maximum preferred length for the misfit vector. For example, one could provide a misfit for each grid point of the ocean biogeochemical model being optimised, providing a misfit vector length of many thousands. However, the problem with this is many of the misfits physically close to each other in the model will respond very similarly to perturbations in the biogeochemical parameters being optimised, which will essentially result in a heavier weighting to this location of the ocean model. Therefore, the misfit terms should be provided in such a way that they respond to parameter perturbations independently of each other, such as by providing misfits for different observational types (e.g. nitrate, oxygen, phosphate, silicate, etc), for different spatial regions (e.g. North Atlantic, Southern Ocean, etc) and for different depths in the water column (e.g. 0-1000m, 2000-4000m, etc). This is why we initially chose 19 regions, and 3 tracers, which we didn't deviate from.

- "Does the required number of function evaluations (e.g. to reach "baseline optimality") significantly increase if the objective function is provided as a single sum of squared differences, only?"

  Indeed we have tried this. We cannot use DFOLS for such an experiment, but a related code called BOBYQA (derivative free, builds a quadratic approximation, minimises within a trust region etc). BOBYQA is meant for general function minimisation, and so when one applies it to our problem, one indeed would supply only (calls to) the scalar sum of squared differences (not the individual misfit terms). We gave DFO-LS the 19x3 misfits, while we gave BOBYQA the summed square of the 19x3 misfits, and let them both run for about 70 iterations. Their success was very similar, though the parameter recovery by BOBYQA was slower than for DFO-LS by about 10 iterations (which is why we chose to continue with DFO-LS). The main paper for DFOLS [1] and references therein compared DFOLS to BOBYQA on standard data fitting test problems and found DFOLS to be significantly more efficient. Thus in our experience, there are some computational gains in supplying the misfit terms individually, if a code such as DFOLS is able to exploit them.

We hope these explanations are clear but please do not hesitate to let us know if further questions arise.

[1] Cartis, C., Fiala, J., Marteau, B., and Roberts, L.: Improving the flexibility and robustness of model-based derivative-free optimization solvers, *ACM Transactions on Mathematical Software*, 45, 1–35, https://doi.org/10.1145/3338517, 2019.