



EGUsphere, author comment AC1
<https://doi.org/10.5194/egusphere-2022-875-AC1>, 2022
© Author(s) 2022. This work is distributed under
the Creative Commons Attribution 4.0 License.

Response to the Comments RC1 and RC2

Maximilian Gelbrecht et al.

Author comment on "Differentiable programming for Earth system modeling" by
Maximilian Gelbrecht et al., EGU Sphere,
<https://doi.org/10.5194/egusphere-2022-875-AC1>, 2022

We'd like to thank both reviewers for their review of our manuscript and the editor for the handling of our manuscript. In the following we will address both reviews.

Response to RC1 / Review 1

We thank the reviewer for the positive assessment of our article.

With regards to the inclusion of benefits for Data Assimilation (DA) and NWP, we can fully understand the reviewer's suggestion to optionally include it. In fact, we also discussed this within our author group before. Ultimately, in this article, we want to focus on Earth System Modelling and the benefits and challenges of differentiable programming therein. For the revised version of the article we would add a paragraph to the benefits section, in which we also mention the potential and benefits for DA and NWP, similar to the already included references in the adjoint section, but would not go into too much detail. However, we would welcome the opportunity to extend this into a possible follow-up article, for which we would need the expertise of the reviewer or other experts like Alan Geer, as DA and NWP are not within our core areas of expertise.

We are aware of the ongoing research of the team at Google that the reviewer mentioned. To our knowledge they haven't published it yet. The involved scientists like Stephan Hoyer and Dmitrii Kochkov did however publish noteworthy papers on scientific machine learning and how to integrate knowledge into machine learning methods before. We would include those in a revised manuscript in the section on Challenges of Differentiable ESMs.

Response to RC2 / Review 2

We thank the reviewer for their detailed assessment of our article that will certainly help to improve a revised version of the manuscript.

Comments of the anonymous reviewer in the bullet points, answers and comments from the authors below in regular paragraphs and font.

- *Many relevant works are not cited. There are several PDE software packages that employ differentiable programming: FEniCS, Firedrake, and devito, to name a few in chronological order. While not originally built to use differentiable programming, it is also possible in deal.II using dual numbers. There are yet more software packages built*

on these tool kits for modeling individual components of the earth system, for example Gusto, Thetis, icepack, VarGlaS. Granted, no one has built, say, a coupled atmosphere-ocean GCM using these tools, but they're worth mentioning nonetheless The biggest omission regarding differential programming for PDE solvers is Farrell et al (2013), Automated derivation of the adjoint of high-level transient finite element programs. This paper won the SIAM Wilkinson Prize for Numerical Software in 2015.

We thank the reviewer for this comment. So far in our article we didn't go into detail on the different discretisation techniques that ESMs use. As far as we know, all of the projects that the reviewer lists here are concerning FEM methods. FEM methods are just one possible discretisation technique: (pseudo-)spectral, finite differences, finite volume and other forms of discretisation also all play a role for ESMs. We consider it therefore outside of the scope of the article to go into a lot of detail on FEM solvers. However, in our revised article we would add a paragraph on discretisation techniques in general. In this paragraph we will mention that one can realize differentiable ESMs independent of the chosen discretisation method, as e.g. demonstrated by the Farrell paper that the reviewer suggested, which shows that differentiable programming can also be applied to FEM models. Additionally, in our revised article we will include additional references showcasing the prior research on combining ML techniques with ice-sheet models.

- *32, "Modern AD systems are able to differentiate most typical operations that appear in ESMs": What about flux or slope limiters? Do you believe in discretize then optimize, or the other way around?*

In general, slope limiters can also be part of differentiable ESMs. The practical implementation of an ESM component that includes a slope limiter will depend on the model in question, its solvers, discretisation and the choice of slope limiter. In the revised manuscript we would add comments on slope limiters near the paragraphs about enforcing constraints in the Challenges of Differentiable ESMs section.

- *40, "Third, additional information from observations can be integrated into ESMs with Machine Learning (ML) models." I'd say that ML tools enable you to construct very complex statistical models and train them with the data you have, but ML as such does not somehow enable you to integrate more information from this data into process-based physical models of the earth system than you could with a more old-school statistical system identification or parameter estimation viewpoint. This is classic information theory, see Kullback's 1958 book.*

We thank the reviewer for this comment, but it is not quite clear to us to which old-school statistical system identification or parameter estimation methods they refer. Machine learning methods like artificial neural networks are also not really new. They built upon statistics and optimisation theory like many other methods. ANNs however do provide extremely flexible universal function approximators that, through their very high capacity, are able to model more complex behaviour than many other methods. In our article we also cite various papers, e.g. the work from Um et al., Yuval et al., and Rasp et al., which showcase how ANNs can be used to improve a more traditional subgrid parametrisation. The point we were trying to make here is that, once a process-based ESM (component) is formulated such that it is automatically differentiable, it will also be much easier to seamlessly combine it with ML components; in addition, optimizing both the parameters of the process-based component and the parameters of the ML part will only be possible if both are automatically differentiable.

- *91-94, "Artificial neural networks (ANNs) can be seen as a subset of these models, but differentiable programming goes far beyond these building blocks": A lot of the wording here is conflating what problem you're trying to solve with how you're trying to solve it. Fitting the parameters of a model, whether it's an ANN or process-based physics model,*

is the answer to the "what" question. There are many ways you could solve this fitting problem. You could use derivative-free optimization methods -- it's not a very good idea, but you could do it. Using gradient-based optimization methods is the answer to a "how" question, and using AD to compute the gradient as opposed to deriving it on pen and paper (which you can still do for some PDE models) is a subset of that "how" question. The fact that you can differentiate through control flow or user-defined types is definitely a compelling reason to use AD. You do address this and quite well in section 4, but it's really important to make the distinction clear.

We thank the reviewer for their careful review of this section; indeed we should have made this clearer. In the revised manuscript, we will again go over this section and be more concise on the "how" and "what" from the perspective of an Earth system modeler.

- 170: I think it's worth making a bigger deal out of the fact that you can get the second derivative so easily with AD. It's often painful but still possible to manually derive a first-order adjoint model, but going to second order by hand is really atrocious.

We agree with the reviewer that computing second derivatives can have considerable benefits in theory, and we do mention this in a number of places in the manuscript, e.g. in the overview figure. However, computing second derivatives also comes at a cost. In particular, computing the Hessian can consume too much memory to be worth considering. Often, methods rather try to estimate a Hessian-vector-product in ways that don't actually need second derivatives at all. That being said, if more models and tools are able to compute second derivatives easily, it is possible that more algorithms will be developed that might avoid the huge memory cost of the full Hessian. Therefore we'll add another sentence on Hessians to the manual vs automatic adjoint section.

- 175: *Here it's worth citing some of Noemi Petra's work, including here paper on stochastic Newton MCMC as well as her more recent work on hIPPYlib.*

We thank the reviewer for pointing us to this work and will add it to the revised manuscript.

On behalf of the authors,

With best regards,

Maximilian Gelbrecht