Hydrology and Earth System Sciences Discussions

Hydrol. Earth Syst. Sci. Discuss., referee comment RC3
https://doi.org/10.5194/hess-2022-73-RC3, 2022

Comment on hess-2022-73

Anonymous Referee #3

Referee comment on "Forward and inverse modeling of water flow in unsaturated soils with discontinuous hydraulic conductivities using physics-informed neural networks with domain decomposition" by Toshiyuki Bandai and Teamrat A. Ghezzehei, Hydrol. Earth Syst. Sci. Discuss., https://doi.org/10.5194/hess-2022-73-RC3, 2022

The paper is very interesting and introduces a physics-informed neural networks (PINNs) method in a Richards' equation context, aimed at approximating the solution to the RRE using neural networks while concurrently matching available soil moisture data. In particular, in this paper authors consider domain decomposition for handling infiltration into layered soils.

The topic is definitely up to date, the paper is well written, and provides all the details for implementing and understanding this approach. Nevertheless, I think authors should address some comments and issues before it can be accepted.

- This PINN approach appears really fascinating because it allows to integrate physics-based models (such as RRE) with machine learning features. Authors ascribe the uncertainties in Richards'equation to the choice of boundary conditions, which is surely right. Nevertheless, I think they do not consider the (even more) cumbersome uncertainties arising from the choice of model parameters, which are the result of some non-linear fitting in laboratory experiments (I am referring to the parameters in the WRC and HCF). This point is a main concern for me: as a matter of facts, unsaturated flow dynamics strongly relies on functions parameters, rather than on ICs and BCs, which are generally easier to assess. On the other hand, I see authors already published a paper on this topic: I think it would be valuable to stress the differences between the two papers
- Lines 197-198: few more words for sketching how the partial derivatives are computed would be valuable
- Figure 1: I think there is a typo in the box "Physics and Data Constraints", since the partial derivative at the left-hand side should be accomplished with respect to time.
- I understand that the residual is computed between the synthetic data and the computed (by the PINNs) ones; in this framework, what is the rationale of comparing the PINNs output with any Richards solver (as Hydrus)?
- As far as I understand, the power of this approach is to combine physics-based models

with data driven ones; according to my knowledge, this is also the spirit of Data Assimilation (DA) methods, which incorporate measurements into a physics based model, albeit in a very different framework; these methods have also been treated in Richards' equation context (see for instance Berardi et al CPC https://doi.org/10.1016/j.cpc.2016.07.025, Medina et al HESS https://doi.org/10.5194/hess-18-2521-2014, Liu et al JoH https://doi.org/10.1016/j.jhydrol.2020.125210); what is authors' opinion about this? What would be the pros and cons of PINNs approach with respect to DA one? Also DA methods allow to assimilate boundary conditions, as in this case, and hydraulic parameters, as well as states. As a matter of fact, with respect to DA methods, this PINNs approach seems to me more on the theoretical side (which is definitely fine, of course) rather than application oriented.

- Authors mention the possibility to drop loss terms for IC or BC at line 225. However, they have not presented any experiment for this scenario. Could you please comment on this ill-posed configuration? How would it perform with respect to classical solver?
- Figure 11 and 3. Please replace "Fintie" with "Finite".
- Authors make use of synthetic data: I had hard times to find where the reference to used data is described. Of course the use of synthetic data is fine, but they should highlight it at the beginning of the paper. Moreover, could you please explain how your method of synthetic data generation could compare to real measurement data? In other words, how robust is your result with respect to outliers, sensor noise and other technical issues when it comes to real data?