Review of “A Python library for solving ice sheet modeling problems using Physics Informed Neural Networks, PINNICLE v1.0”

Facundo Sapienza
May 2025

The manuscript contributes to the scientific literature by introducing the Python package PINNACLE for ice sheet forward and inverse modelling based on PINNs. The paper successfully addressed the technical aspects behind the implementation of these techniques. The manuscript is well written and represents an important contribution to the literature that fits perfectly well the scope of the journal.

I strongly recommend the publication of the manuscript in GMD after the minor revisions and comments here presented are addressed. If I decided to divide my comments between major and minor, please interpret my major comments as rather being intermediate. I think it would be important to address these points in the revised version of the manuscript, but this does not shade the overall quality of the work.

All comments, both major and minor, are aimed to improve the quality of the paper or to elucidate some of my doubts or questions.

Major Comments

- **Automatic Differentiation machinery.** The manuscript does not discuss much about the automatic differentiation techniques used for training the PINN. There are mentions of the use of different AD backeds in the abstract and introductions, and there is a mention of the use of “backpropagation” in line 74. This last mention made me conclude (maybe wrongly) that all the derivatives in the code are computed using reverse AD. The reason I am bringing this up is because it is well known that derivatives with respect to a few parameters are more efficiently being computed using forward AD rather than reverse AD (backpropagation). In the case of a PINN, this implies that derivatives with respect to the input layer (t, x, y) should be computed with forward AD and derivatives with respect to the parameters of the neural network should be computed with reverse AD. The combination of these two types of derivatives results in nested AD codes that may be more complicated to implement but they are definitely more both memory and time efficient.

  It will be important in the manuscript to clarify how these two types of derivatives are computed. Furthermore, this has a major impact on the performance of the code, reason why I will encourage the authors to add a mention to this in Section 9 since this directly has an impact on the performance of the method.

- **Data section.** There is no description of the data that will be used in the model (from a generic perspective). Up to this point in the manuscript, it is not clear if these data correspond to ice thickness, velocities, etc; How are they sampled? What are the temporal-spatial resolutions of these? etc. This gets more clear with the examples, but up to this point it is unclear what the data are. I will recommend starting this section with an introduction to the data products or types considered for the simulation rather than the data format of these hypothetical datasets. The second paragraph [136-141] does not really fit in this section, and I would recommend moving it to Section 6 which addresses the topic of loss functions.

- **Confusion about the use of weights in the losses and the normalization of them.** I am a bit confused with Table 2: are all these quantities normalized in the code? I think there is some confusion in the use of weights. One one side, there are the weights of the different terms in the loss functions that need to be calibrated with some model selection technique (in this case, by running multiple simulations and selecting the best one). On the other hand, the weights mentioned in the context of Table 2 seem to refer to the weights of individual physical variables. These two are different in nature, and the use of one does not exclude the other. I find the discussion in this section to be mixing these two notions of weighthing and scaling.

Minor Comments

- [30] No need to capitalize “Partial Differential Equation”.

- [33] The authors comment that classical methods are difficult in part due to the demand of regularization. However, this is also the case with PINNs and many other techniques. The physical term in the loss function is nothing else than a regularization term. The use of regularization is common in most inverse modelling techniques, both based in numerical solvers and in machine learning techniques. I am aware the authors are familiar with this concept, as it is clear from line 46. However, when introduced in the context of classical techniques it seems to me it is overemphasized the negative effect of having to use regularization techniques. I will suggest expanding on the difficulties of regularization in classical settings, or mention that this challenge is common to both families of methods.

- [39] Please check the consistency between capitalizing or not Physics Informed Neural Network in the manuscript (in the current version, both occur). Personally, I don’t think the method should be capitalized, but it is up to the authors.

- I am not sure Figure No1 really helps much with the reading of the manuscript in its current format. The figure does not present much content, and it is very similar to the figure in Cheng (2024) and to the later figures in the manuscript, which make a better point in explaining the design of the PINN. I will suggest to abstract this figure a bit more to really represent a high-level scheme of the tool (e.g., with illustrations showing how the data and physics may look) or directly to remove the figure.

- [72] Maybe change “depending on the governing equations” -> “for time-dependent PDEs”.

- [79] The manuscript states “ the training procedure is to minimize the loss function, such that the output of the neural network satisfies the governing PDEs and also matches the data provided”. This is not true for PINNs, at least not in principle. PINNs include the differential equation as a soft constraint in an optimization procedure, and there is no guarantee (in principle) that the residual of the physical term loss is zero (or close enough to zero). I think this is an important point that should be clear in the manuscript (which is covered in more detail in Cheng 2024).

- [98] I will suggest moving the reference to the end of the line so reference and acronym are not mixed.

- [Eq 1 & 2] Please check the consistency between the use of u, v and \bar{u}, \bar{v} between equations. I imagine in Eq 2 the velocities are the vertical averaged ones, but this is not clear from the text.

- [102] H has been already introduced in [91].

- [110] Please notice that the friction law derived in Weertman 1957 actually corresponds to m=2, no m=3. If variations and corrections to the original Weertman theory have been made, I will try to include one of these references if m=3 is going to be the standard value (see for example Section 3.2 in Law et. al (2025) "What is glacier sliding?").

- [125] I am not sure "b" (bed) has been introduced in the text. Probably a good place to introduce it is after Equation 5, where it would be important to show that the normalized depth factor goes from zero to one between bed and surface.

- [139] There may be cases where \epsilon may be selected larger than machine precision.

- Caption in Table 1 is missing a period.

- [153] Is the denormalization also done with a min-max scaling?

- [155] I am confused about the normalization here used. All physical quantities are defined in the same units systems, as they should be. On the other hand, the neural network inputs and outputs dimensionless quantities, and dimensionalization of them is done in a pre- and post-processing step. However, this does not necessarily mean that there is no further need to normalize the physical equations. For example, Equation 6 still has dimensional quantities involved (e.g, rho and g). Can the author clarify on this point, both if they were trying to imply something different or if they effectively do a dimensional analysis of the equations. This point may be related to my major concern regarding Table 2.

- [159] It is clear here that the operation cos(Bx) is applied coordinate-wise on the vector Bx. Since this is not the notation used in the rest of the paper, it may be worth clarifying.

- [163] The example of the loss function depicted in Equation 9 is very simplistic, and at this point in the manuscript still leaves uncertainty about what exactly is the optimization problem to be solved. This is probably familiar to people familiar with PINNs, but maybe not to the general audience. I think it is ok to leave Equation 9 as it is, but I will suggest adding a sentence in this section clarifying that more concrete examples of losses will be provided in the later sections including examples. I am more of a fan of the introduction of the losses as it is presented in the PINNACLE documentation (https://pinnicle.readthedocs.io/en/latest/training/lossfunctions.html), where the authors use an example to show how a general loss function would look like. This presentation also helps illustrate better the role of the weight/hyperparameter in the physical terms in the loss function, which is not clear as it is in Equation 9 right now.

- [174] Can the authors clarify how they quantify "best performance" over the different experiments? Is this based in some other metric or based on the different contributions to the loss term to be of the same order? This last point is very important, since it is usually a challenge in any model involving multiple losses. Would the authors provide some references here that can help strengthen this point?

- [182] “to evolve alongside advances in physics-informed machine learning in the future” -> “to evolve alongside future advances in physics-informed machine learning”

- [191] I would suggest adding the references to the software packages mentioned in the manuscript (e.g., JAX). It does always help to cite open source libraries :)

- [195] If the random sampling strategy is important for the learning task, I will encourage the authors to include this mention in the “Loss function” section rather than in the “structure of the package”. Even when this can be mentioned in this later section too, it is important that a random sampling strategy is used for training. This is a form of implicit regularization applied to the optimization problem.

- [201] The authors mentioned the use of Hammersley sequence sampling which produces a random set of collocation points. My understanding of this method after a quick read based on the provided reference (Wong 1997) is that Hammersley sampling generates a deterministic set of points, not random. Random versions of this method exist though (see for example Section 8.6.2 of the book “Sampling and Reconstruction”). Can the authors please clarify if the sampling method is deterministic or not, and if not provide a reference to understand the sampling procedure? Notice that the use of a deterministic or random sampling technique for the collocation method can have an impact in the learning task. (I am glad the authors mentioned this in the manuscript and they motivate me to better understand this technique, so please take this comment as a suggested improvement of someone who cares about the little details in the methods.) Furthermore, are the polygons generated with these sampled points? Do they use a different technique?

- [217] The authors clarify the point that the friction coefficient C is not used as data. However, I still find it confusing that the friction is included in the data box in Figure 3. Considering a picture is worth a thousand words, I will suggest the authors re-arrange Figure 3 to not display the friction coefficient in the data box. In this way, it won’t be necessary to clarify this in the manuscript. Furthermore, it is a bit confusing that in Figure 4 the friction coefficient C is plotted alongside other observable quantities at the same level. I will suggest moving the figures with C to the right of the figure and maybe plot a line in between this and the plots for H, u, and s, to make it clear that this is not an observation.

- [223] Missing “+” symbol in Eq 10, beginning of second line.

- [231] Is it possible to run the experiment for longer? This will show reproducibility of the results compared with Cheng (2024).

- Missing period at the end of caption for Listing 1, 2, and 3.

- [243] I don’t think it is necessary nor helps the reader to include the name of the data files in the manuscript (both PIG.mat and BC.mat). Unless the authors are trying to make a point by naming the files, I will suggest removing such references.

- [297] I am not sure about this last point in the performance section. What do the authors mean by "effectively resolving inversion problems with smoother, physically consistent solutions"? Solutions of the classical adjoint method will actually have strict physical consistent solutions, at differences of PINNs that provide a relaxation of the numerical solution of the underlying differential equation.

- I was taking a quick look at the Python package and this looks great, very nice work. Just to improve the presentation of the work, please let me suggest further adding a link to the docs in the code availability section and an executable example if possible (for example, a colab or Binder link where the simplest of the model can be executed). This is of course not really required, but I think it will be an interesting further contribution to the manuscript.

- Comment about the listings 1, 2, and 3. I like the initiative of the authors to show how the software can be used from the user perspective and the different parameters of the models that can be specified inside PINNACLE. However, I am not convinced that the three listings do successfully communicate this point, since the scripts shown are rather specific and communicate rather the effort of the authors in writing a very modular library. I think it is ok to keep one of the listings for illustration, but I don't really see a win in having the three of them in the main manuscrupt. Instead, I will suggest having a table with the different parameters than can be customize in the model (the ones currently in the lines of code in the listing plus some others) and also reference the reader in the manuscript to a concrete example (e.g., something contain in a Jupyter notebook or Python script). Following my previous comment, even better if this example can be directly executed using colab or Binder (even a very small toy example will do the job). This is a suggestion as it is up to the authors to decide what to do with this comment.