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

This manuscript presents the development of a new Python package that uses physics-informed neural networks (PINNs) to solve inverse problems in ice-sheet and ice-shelf dynamics. The authors provide a comprehensive description of their methods, including the network architecture, the physical constraints, the loss function, and the training hyperparameters, all of which are significant for successful training. The authors also provide concrete examples and technical details for using the package to solve various types of problems. Based on the provided example code, the package appears to be efficient and user-friendly. The manuscript is well-written and has a clear structure. I recommend this manuscript for publication, provided the authors address the following questions regarding novelty and reproducibility:

1. **Loss function weight:**

As mentioned in the manuscript, the PINN loss function comprises different terms, including data misfit and equation residuals. Each term requires a prefactor to weigh its contribution to the total loss. Table 2 provides the default values for these weights in PINNICLE, which show order-of-magnitude differences.

- Regarding robustness, how can a user determine if these default values are suitable for their specific problem?
- If adjustments are necessary, could the authors provide some rules-of-thumb for modifying these values, particularly for the equation residual weights?
- Regarding implementation, could the authors include an example in one of Listings 1-3 demonstrating how to set weights different from the default values?

  .

2. **Fourier Feature:**

The authors mention the use of Fourier Features in PINNICLE, expressed as $f(x)=[\cos(Bx),\sin(Bx)]$, where B is sampled from a Gaussian distribution $N(0,\sigma)$. According to Tancik et al., σ is a hyperparameter that users must carefully determine to capture high-frequency features in the data without overfitting.

- Therefore, does PINNICLE automatically select the optimal σ value for generating the Fourier Feature net weights based on different training data, or should the user set this value manually?
- If the latter, what is the default value of σ in PINNICLE?

3. **Random sampling:**

In Line 197, the authors state that "...PINNICLE employs a random sampling strategy to automatically load data…".

- Regarding the random sampling strategy, does PINNICLE exclusively use a uniform distribution for random sampling, or are other options available for users to select?
- Does PINNICLE permit users to implement their own customized sampling strategies?
- Another question regarding random sampling is whether the data sampling points and the collocation points used for evaluating equation residuals are fixed throughout the training process or re-sampled at certain iteration intervals?

4. **Mass Balance:**

The mass balance equation (Equation 1) includes a net mass accumulation term, a.

- When incorporating this equation into the PINN training, does the user need to provide data for a across the entire domain, or can a be treated as a constant?
- If a can be treated as a constant, what is its default value in PINNICLE?

5. Novelty: Another Python package, DIFFICE-jax, which also uses PINNs to study ice dynamics, was recently published. Regarding novelty, could the authors elaborate on the differences between PINNICLE and DIFFICE-jax?

Minor Point:

- Figure 1 appears somewhat sparse. Consider shrinking the image or reorganizing its content to make it more compact.

Suggested Additional Citations:

The following relevant publications on PINNs in glaciology could also be cited:

Riel, Bryan, and Brent Minchew. "Variational inference of ice shelf rheology with physics-informed machine learning." *Journal of Glaciology* 69.277 (2023): 1167-1186.

Wang, Yongji, et al. "Deep learning the flow law of Antarctic ice shelves." *Science* 387.6739 (2025): 1219-1224.

Wang, Yongji, and Ching-Yao Lai. "DIFFICE-jax: Differentiable neural-network solver for data assimilation of ice shelves in JAX." *Journal of Open Source Software* 10.109 (2025): 7254.