





# ODINN.jl: Scientific machine learning glacier modelling

Jordi Bolibar<sup>1,2</sup>, Facundo Sapienza<sup>3,4</sup>, Alban Gossard<sup>1</sup>, Mathieu le Séac'h<sup>1</sup>, Vivek Gajadhar<sup>2</sup>, Fabien Maussion<sup>5,6</sup>, Bert Wouters<sup>2</sup>, and Fernando Pérez<sup>4</sup>

<sup>1</sup> Univ. Grenoble Alpes, CNRS, IRD, G-INP, Institut des Géosciences de l'Environnement, Grenoble, France <sup>2</sup> Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands <sup>3</sup> Department of Geophysics, Stanford University, Stanford, United States <sup>4</sup> Department of Statistics, University of California, Berkeley, United States <sup>5</sup> Bristol Glaciology Centre, School of Geographical Sciences, University of Bristol, Bristol, UK <sup>6</sup> Department of Atmospheric and Cryospheric Sciences, University of Innsbruck, Innsbruck, Austria  Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

## Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

## Summary

ODINN.jl is a glacier model leveraging scientific machine learning (SciML) methods to perform forward and reverse simulations of large-scale glacier evolution. It can simulate both surface mass balance and ice flow dynamics through a modular architecture which enables the user to easily modify model components. For this, ODINN.jl is in fact an ecosystem composed of multiple packages, each one handling a specific task:

- Sleipnir.jl:** Handles all the basic types, functions and datasets, common through the whole ecosystem, as well as data management tasks.
- Muninn.jl:** Handles surface mass balance processes, via different types of models.
- Huginn.jl:** Handles ice flow dynamics, by solving the ice flow partial differential equations (PDEs) using numerical methods. It can accommodate multiple types of ice flow models.
- ODINN.jl:** Acts as the interface to the whole ecosystem, and provides the necessary tools to differentiate and optimize any model component. It can be seen as the SciML layer, enabling different types of inverse methods, using hybrid models combining differential equations with data-driven models.

The ODINN ecosystem extends beyond this suite of Julia packages, by leveraging the data preprocessing tools of the Open Global Glacier Model (Maussion et al. (2019), OGGM). We do so via an auxiliary library named Gungnir, which is responsible for downloading all the necessary data to force and initialize the model, such as glacier outlines from the Randolph Glacier Inventory (RGI), digital elevation models (DEMs), ice thickness observations from GlaThiDa, ice surface velocities from different studies and many different sources of climate reanalyses and projections. This implies that ODINN.jl, like OGGM, is virtually capable of simulating any of the 200,000 glaciers on Earth.

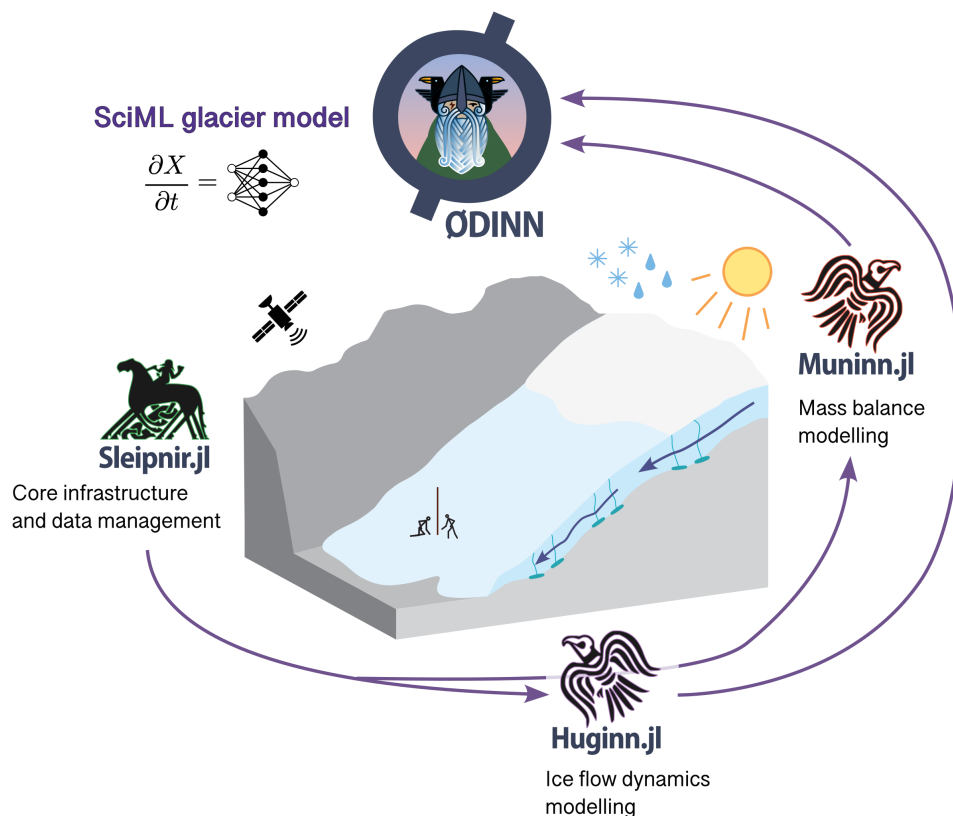


Figure 1: Overview of the ODINN.jl ecosystem.

ODINN.jl provides a high-level user-friendly interface, enabling the user to swap and replace most elements of a glacier simulation in a very modular fashion. The main elements of a simulation, such as the Parameters, a Model and a Simulation (i.e. a Prediction or an Inversion), are all objects that can be easily modified and combined. In a few lines of code, the user can automatically retrieve all necessary information for most glaciers on Earth, compose a Model based on a specific combination of surface mass balance and ice flow models, and incorporate data-driven models (e.g. a neural network) to parametrize specific physical processes of any of these components. Both forward and reverse simulations run in parallel using multiprocessing, leveraging Julia's speed and performance. GPU compatibility is still not ready, due to the difficulties of making everything compatible with automatic differentiation (AD). Nonetheless, it is planned for future versions.

The most unique aspect of ODINN.jl is its differentiability and capabilities of performing all sorts of different hybrid modelling. Since the whole ecosystem is differentiable, we can optimize almost any model component, providing an extremely powerful framework to tackle many scientific problems. ODINN.jl can optimize, separately or together, in a steady-state or transient way:

- The initial or intermediate state of glaciers (i.e. their ice thickness  $H$ ) or the equivalent ice velocities  $V[x, y]$ .
- Model parameters (e.g. the ice viscosity  $A$  in a 2D Shallow Ice Approximation), in a gridded or scalar format. This can be done for multiple time steps where observations (e.g. ice surface velocities) are available.
- The parameters of a regressor (e.g. a neural network), used to parametrize a subpart or one or more coefficients of an ice flow or surface mass balance mechanistic model. This enables the exploration of empirical laws describing physical processes of glaciers.

59 For this, it is necessary to use reverse differentiation to compute the required vector-jacobian  
60 products (VJPs). We have two strategies to achieve this: (1) manual adjoints, which have been  
61 implemented using AD via `Enzyme.jl`, as well as fully manual implementations of the discrete  
62 and continuous adjoints; and (2) automatic adjoints using `SciMLSensitivity.jl`, providing  
63 both continuous and discrete versions and available with different AD back-ends. These two  
64 approaches are complementary, with the manual adjoints being ideal for high-performance  
65 tasks, and serving as a ground truth for benchmarking and testing automatic adjoint methods  
66 from `SciMLSensitivity.jl`.

67 Beyond all these inverse modelling capabilities, `ODINN.jl` can also act as a more conventional  
68 forward glacier model, simulating glaciers in parallel, and easily customizing almost every  
69 possible detail of the simulation. Its high modularity, combined with the easy access to  
70 a vast array of datasets coming from OGGM, makes it very easy to run simulations, even  
71 with a simple laptop. `Huginn.jl` is responsible for the ice flow dynamics models, with an  
72 architecture capable of integrating and easily swapping various models. Models based on partial  
73 differential equations (PDEs) are solved using `DifferentialEquations.jl`, which provides  
74 access to a huge amount of numerical solvers. For now, we have implemented a 2D Shallow  
75 Ice Approximation (SIA), but in the future we plan to incorporate other models, such as the  
76 Shallow Shelf Approximation (SSA). In terms of surface mass balance, `Muninn.jl` incorporates  
77 for now simple temperature-index models. Nonetheless, the main addition of the upcoming  
78 version will be the machine learning-based models from the `MassBalanceMachine`, which will  
79 become the de-facto solution. Frontal ablation (i.e. calving) and debris cover are not available  
80 for now, but we plan to add it to future versions of the model.

## 81 Statement of need

82 `ODINN.jl` has been designed to address the need for a glacier model which can leverage  
83 both the interpretability and established knowledge coming from the literature in the form of  
84 mechanistic models based on differential equations, with the flexibility and data-assimilation  
85 capabilities of data-driven models. The combination of these two paradigms enables a targeted  
86 approach to inverse methods for learning parametrizations of glacier physical processes, learning  
87 only the unknown physics and keeping a reliable structure in the dynamics in the form of a  
88 differential equation. While purely mechanistic and data-driven modelling approaches exist  
89 in glaciology, there is a need for flexible models which can leverage existing widely available  
90 observations at the glacier surface, to simulate complex physical processes of glaciers, such as  
91 basal sliding, creep or calving. Existing laws do not necessarily map available observations with  
92 these physical processes, diffculting the finding and calibration of parametrizations and laws.  
93 Approaches based on functional inversions and differentiable programming offer the needed  
94 flexibility to derive new empirical laws based on carefully-chosen input proxies, which can help  
95 to test hypothesis of what can constitute and drive new parametrizations.

96 At the same time, a good representation of this complex and poorly represented physical  
97 processes is key to accurate predictions of glacier evolution, crucial for their impact to both  
98 freshwater resources and sea-level rise. Therefore, with `ODINN.jl`, we provide a unified modelling  
99 ecosystem, capable of both flexible and advance inverse methods for model calibration, as  
100 well as efficient and modular methods for forward simulations for large-scale glacier modelling.

## 101 Acknowledgements

102 We acknowledge the help of Chris Rackauckas for the debugging and discussion of issues  
103 related to the SciML Julia ecosystem, Redouane Lguensat for scientific discussions on the first  
104 prototype of the model. JB acknowledges financial support from the Nederlandse Organisatie  
105 voor Wetenschappelijk Onderzoek, Stichting voor de Technische Wetenschappen (Vidi grant  
106 016.Vidi.171.063). FS acknowledges funding from the National Science Foundation (EarthCube

programme under awards 1928406 and 1928374).

## References

- Maussion, F., Butenko, A., Champollion, N., Dusch, M., Eis, J., Fourteau, K., Gregor, P., Jarosch, A. H., Landmann, J., Oesterle, F., Recinos, B., Rothenpieler, T., Vlug, A., Wild, C. T., & Marzeion, B. (2019). The Open Global Glacier Model (OGGM) v1.1. *Geoscientific Model Development*, 12(3), 909–931. <https://doi.org/10.5194/gmd-12-909-2019>

DRAFT