





ODINN.jl: Scientific machine learning glacier modelling

Jordi Bolibar^{1,2}, Facundo Sapienza^{3,4}, Alban Gossard¹, Mathieu le Séac'h¹, Vivek Gajadhar², Fabien Maussion^{5,6}, Bert Wouters², and Fernando Pérez⁴

¹ Univ. Grenoble Alpes, CNRS, IRD, G-INP, Institut des Géosciences de l'Environnement, Grenoble, France ² Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands ³ Department of Geophysics, Stanford University, Stanford, United States ⁴ Department of Statistics, University of California, Berkeley, United States ⁵ Bristol Glaciology Centre, School of Geographical Sciences, University of Bristol, Bristol, UK ⁶ 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Published: unpublished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](https://creativecommons.org/licenses/by/4.0/))

Summary

ODINN.jl is a glacier model leveraging scientific machine learning (SciML) methods to perform forward and inverse 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 ([Bezanson et al., 2017](#)) packages, by leveraging the data preprocessing tools of the Open Global Glacier Model (OGGM, [Maussion et al. \(2019\)](#)). We do so via an auxiliary library named *Gunǵnir*, 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 Consortium ([2023](#)), RGI), digital elevation models (DEMs), ice thickness observations from *GlaThiDa* ([Consortium, 2019](#)), ice surface velocities from different studies ([Millan et al., 2022](#)) and many different sources of climate reanalyses and projections ([Eyring et al., 2016](#); [Lange, 2019](#)). This implies that ODINN.jl, like OGGM, is virtually capable of simulating any of the 200,000 glaciers on Earth.

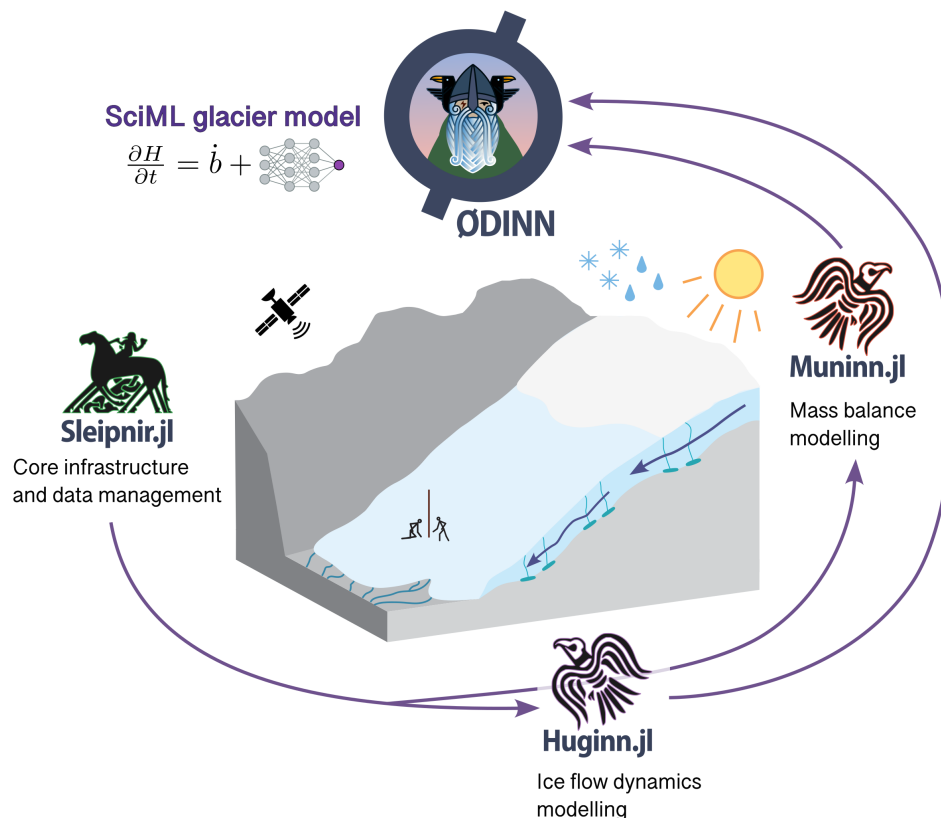


Figure 1: Overview of the ODINN.jl ecosystem.

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 (Bolibar et al., 2023). 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 (Hutter, 1983)), 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.

59 This enables the exploration of empirical laws describing physical processes of glaciers,
60 leveraging Universal Differential Equations (UDEs, Christopher Rackauckas et al. (2021)).

61 For this, it is necessary to differentiate through complex code, including numerical solvers,
62 which is non-trivial (Sapienza et al., 2024). We use reverse differentiation to compute the
63 required vector-jacobian products (VJPs). We have two strategies to achieve this: (1)
64 manual adjoints, which have been implemented using AD via Enzyme.jl (Moses et al., 2021),
65 as well as fully manual implementations of the discrete and continuous adjoints; and (2)
66 automatic adjoints using SciMLSensitivity.jl (Chris Rackauckas et al., 2019), providing
67 both continuous and discrete versions and available with different AD back-ends. These two
68 approaches are complementary, with the manual adjoints being ideal for high-performance
69 tasks, and serving as a ground truth for benchmarking and testing automatic adjoint methods
70 from SciMLSensitivity.jl.

71 Beyond all these inverse modelling capabilities, ODINN.jl can also act as a more conventional
72 forward glacier model, simulating glaciers in parallel, and easily customizing almost every
73 possible detail of the simulation. Its high modularity, combined with the easy access to a vast
74 array of datasets coming from OGGM, makes it very easy to run simulations, even with a
75 simple laptop. Huginn.jl is responsible for the ice flow dynamics models, with an architecture
76 capable of integrating and easily swapping various models. Models based on partial differential
77 equations (PDEs) are solved using DifferentialEquations.jl (Christopher Rackauckas &
78 Nie, 2017), which provides access to a huge amount of numerical solvers. For now, we
79 have implemented a 2D Shallow Ice Approximation (SIA, Hutter (1983)), but in the future
80 we plan to incorporate other models, such as the Shallow Shelf Approximation (SSA, Weis
81 et al. (1999)). In terms of surface mass balance, Muninn.jl incorporates for now simple
82 temperature-index models. Nonetheless, the main addition of the upcoming version will be the
83 machine learning-based models from the MassBalanceMachine (Sjursen et al., 2025), which
84 will become the de-facto solution. Frontal ablation (i.e. calving) and debris cover are not
85 available for now, but we plan to add it to future versions of the model.

86 Statement of need

87 ODINN.jl has been designed to address the need for a glacier model which can leverage both the
88 interpretability and established knowledge coming from the literature in the form of mechanistic
89 models based on differential equations, with the flexibility and data-assimilation capabilities of
90 data-driven models (Bolibar et al., 2023). The combination of these two paradigms enables a
91 targeted approach to inverse methods for learning parametrizations of glacier physical processes,
92 learning only the unknown physics and keeping a reliable structure in the dynamics in the form
93 of a differential equation. While purely mechanistic and data-driven modelling approaches exist
94 in glaciology, there is a need for flexible models which can leverage existing widely available
95 observations at the glacier surface, to simulate complex physical processes of glaciers, such as
96 basal sliding, creep or calving. Existing laws do not necessarily map available observations with
97 these physical processes, diffculting the finding and calibration of parametrizations and laws.
98 Approaches based on functional inversions and differentiable programming offer the needed
99 flexibility to derive new empirical laws based on carefully chosen input proxies, which can help
100 to test hypothesis of what can constitute and drive new parametrizations.

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

Acknowledgements

We acknowledge the help of Chris Rackauckas for the debugging and discussion of issues related to the SciML Julia ecosystem, Redouane Lguensat for scientific discussions on the first prototype of the model and Julien le Sommer for scientific discussions around differentiable programming. JB acknowledges financial support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek, Stichting voor de Technische Wetenschappen (Vidi grant 016.Vidi.171.063) and a TU Delft Climate Action grant. FS acknowledges funding from the National Science Foundation (EarthCube programme under awards 1928406 and 1928374). AG acknowledges funding from the MIAI cluster and Agence Nationale de la Recherche (ANR) in the context of France 2030 (grant ANR-23-IACL-0006).

References

- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1), 65–98. <https://doi.org/10.1137/141000671>
- Bolibar, J., Sapienza, F., Maussion, F., Lguensat, R., Wouters, B., & Pérez, F. (2023). Universal differential equations for glacier ice flow modelling. *Geoscientific Model Development*, 16(22), 6671–6687. <https://doi.org/10.5194/gmd-16-6671-2023>
- Consortium, G. (2019). *Glacier Thickness Database 3.0.1*. World Glacier Monitoring Service, Zurich, Switzerland.
- Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., & Taylor, K. E. (2016). Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, 9(5), 1937–1958. <https://doi.org/10.5194/gmd-9-1937-2016>
- Hutter, K. (1983). *Theoretical Glaciology*. Springer Netherlands. <https://doi.org/10.1007/978-94-015-1167-4>
- Lange, S. (2019). *WFDE5 over land merged with ERA5 over the ocean (W5E5)*. GFZ Data Services. <https://doi.org/10.5880/PIK.2019.023>
- 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>
- Millan, R., Mouginot, J., Rabatel, A., & Morlighem, M. (2022). Ice velocity and thickness of the world's glaciers. *Nature Geoscience*, 15(2), 124–129. <https://doi.org/10.1038/s41561-021-00885-z>
- Moses, W. S., Churavy, V., Paehler, L., Hükelheim, J., Narayanan, S. H. K., Schanen, M., & Doerfert, J. (2021). Reverse-mode automatic differentiation and optimization of GPU kernels via enzyme. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. <https://doi.org/10.1145/3458817.3476165>
- Rackauckas, Chris, Innes, M., Ma, Y., Bettencourt, J., White, L., & Dixit, V. (2019). DiffEqFlux.jl - A Julia Library for Neural Differential Equations. *arXiv:1902.02376 [Cs, Stat]*. <http://arxiv.org/abs/1902.02376>
- Rackauckas, Christopher, Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., & Edelman, A. (2021). *Universal Differential Equations for Scientific Machine Learning*. arXiv. <https://doi.org/10.48550/arXiv.2001.04385>
- Rackauckas, Christopher, & Nie, Q. (2017). DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. *Journal of Open*

- 151 *Research Software*, 5, 15. <https://doi.org/10.5334/jors.151>
- 152 RGI Consortium. (2023). *Randolph Glacier Inventory - A Dataset of Global Glacier Outlines,*
153 *Version 7*. National Snow; Ice Data Center. <https://doi.org/10.5067/F6JMOVY5NAVZ>
- 154 Sapienza, F., Bolibar, J., Schäfer, F., Groenke, B., Pal, A., Boussange, V., Heimbach,
155 P., Hooker, G., Pérez, F., Persson, P.-O., & Rackauckas, C. (2024). *Differentiable*
156 *Programming for Differential Equations: A Review*. arXiv. [http://arxiv.org/abs/2406.](http://arxiv.org/abs/2406.09699)
157 [09699](http://arxiv.org/abs/2406.09699)
- 158 Sjurssen, K. H., Bolibar, J., Meer, M. van der, Andreassen, L. M., Biesheuvel, J. P., Dunse,
159 T., Huss, M., Maussion, F., Rounce, D. R., & Tober, B. (2025). Machine learning
160 improves seasonal mass balance prediction for unmonitored glaciers. *EGUsphere*, 1–39.
161 <https://doi.org/10.5194/egusphere-2025-1206>
- 162 Weis, M., Greve, R., & Hutter, K. (1999). Theory of shallow ice shelves. *Continuum Mechanics*
163 *and Thermodynamics*, 11(1), 15–50. <https://doi.org/10.1007/s001610050102>

DRAFT