

ODINN.jl: Scientific machine learning glacier modelling

Jordi Bolibar^{1,2}, Facundo Sapienza^{3,4}, Alban Gossard¹, Mathieu le Séac'h¹, Lucille Gimenes¹, Vivek Gajadhar², Fabien Maussion^{5,6}, Ching-Yao Lai³, 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Published: unpublished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.

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 (where differentiable means the ability to compute model derivatives with respect to parameters ([Shen et al., 2023](#))), 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 (time-independent simulation) or transient (time-dependent simulation) way the following model parameters:

- The initial or intermediate state of glaciers (i.e. their ice thickness) or the equivalent ice surface velocities.
- Model parameters (e.g. the Glen coefficient A related to ice viscosity 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 statistical 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, leveraging Universal Differential Equations (UDEs, Christopher Rackauckas et al. ([2021](#))).

For this, it is necessary to differentiate (that is, computing gradients or derivatives) through complex code, including numerical solvers, which is a non-trivial task ([Sapienza et al., 2024](#)). We use reverse differentiation based on the adjoint method to achieve this. We have two strategies for computing both the adjoint and the required vector-jacobian products (VJPs): (1) manual adjoints, which have been implemented using automatic differentiation (AD) via Enzyme.jl ([Moses et al., 2021](#)), as well as fully manual implementations of the spatially discrete and spatially continuous VJPs; and (2) automatic adjoints using SciMLSensitivity.jl ([Chris Rackauckas et al., 2019](#)), available with different AD back-ends for the VJPs computation. These two approaches are complementary, with the manual adjoints being ideal for high-performance tasks by providing more control on the implementation, and serving as a ground

truth for benchmarking and testing automatic adjoint methods from `SciMLSensitivity.jl`. Beyond all these inverse modelling capabilities, `ODINN.jl` can also act as a more conventional forward glacier model, simulating glaciers in parallel, and easily customizing different model parametrizations and choices within the simulation. Its high modularity, combined with the easy access to a vast array of datasets coming from the Open Global Glacier Model (OGGM, Maussion et al. (2019)), makes it very easy to run simulations, even with a simple laptop. Multiple ice flow dynamics models can be easily swapped, thanks to a modular architecture (see Software design). Models based on partial differential equations (PDEs) are solved using `DifferentialEquations.jl` (Christopher Rackauckas & Nie, 2017), which provides access to a huge amount of numerical solvers. For now, we have implemented a 2D Shallow Ice Approximation (SIA, Hutter (1983)), but in the future we plan to incorporate other models, such as the Shallow Shelf Approximation (SSA, Weis et al. (1999)). Validation of numerical forward simulations are evaluated in the test suite based on exact analytical solutions of the SIA equation (Bueler et al., 2005). Multiple surface mass balance models are available, based on simple temperature-index models. Nonetheless, the main addition of the upcoming version will be the machine learning-based models from the `MassBalanceMachine` (Sjursen et al., 2025), which will provide further mass balance models.

Statement of need

`ODINN.jl` addresses the need for a glacier model that combines the physical interpretability of mechanistic approaches with the flexibility and data-assimilation capabilities of data-driven methods (Bolibar et al., 2023). By integrating both paradigms, it enables targeted inverse methods to learn parametrizations of glacier processes, capturing unknown physics while preserving the physically grounded structure of glacier dynamics through differential equations.

While purely mechanistic and purely data-driven glacier models already exist (e.g. Gagliardini et al. (2013), Maussion et al. (2019), Rounce et al. (2023), Bolibar et al. (2022)), they often lack the flexibility needed to fully exploit the growing wealth of glacier observations, such as ice surface velocities, ice thickness, surface topography, surface mass balance or climate reanalyses. Existing empirical laws do not always link directly to these observables, making their calibration challenging. Approaches based on differentiable programming and functional inversions offer a path forward, allowing the derivation of new empirical relationships from carefully chosen proxies and providing a framework to test hypotheses about poorly understood physical processes such as basal sliding, creep, or calving.

Improving the representation of these complex processes is crucial for accurate projections of glacier evolution and their impacts on freshwater availability and sea-level rise (IPCC, 2021). To this end, `ODINN.jl` provides a unified modelling ecosystem that supports both advanced inverse methods for model calibration and efficient, modular forward simulations for large-scale glacier studies.

Developing such a framework places demanding requirements on scientific software. Inefficient codes and irreproducible implementations can severely restrict progress, emphasizing the importance of open-source, community-driven tools that follow modern research software engineering (RSE) practices (Combemale et al., 2023). For this end, software should support modular and adaptable design patterns that enable prototyping and augmentation of existing pipelines (Nyenah et al., 2024). The Julia programming language provides two key advantages in this context: it solves the two-language problem by offering Python-like high-level expressiveness with C-level performance (Bezanson et al., 2017), and it enables source-code differentiability, essential for modular gradient-based optimization in inverse modelling and setting the foundation for a strong ecosystem where hybrid modelling, and particularly UDEs, can thrive. With `ODINN.jl`, our goal is to provide a robust and future-proof modelling framework that bridges the gap between physical understanding and data-driven discovery. Its modular architecture, thorough testing, and continuous integration (CI) ensure reproducibility and reliability, while its

94 open design invites collaborations and both methodological and applied advancements across
95 the glaciological and Earth system modelling communities.

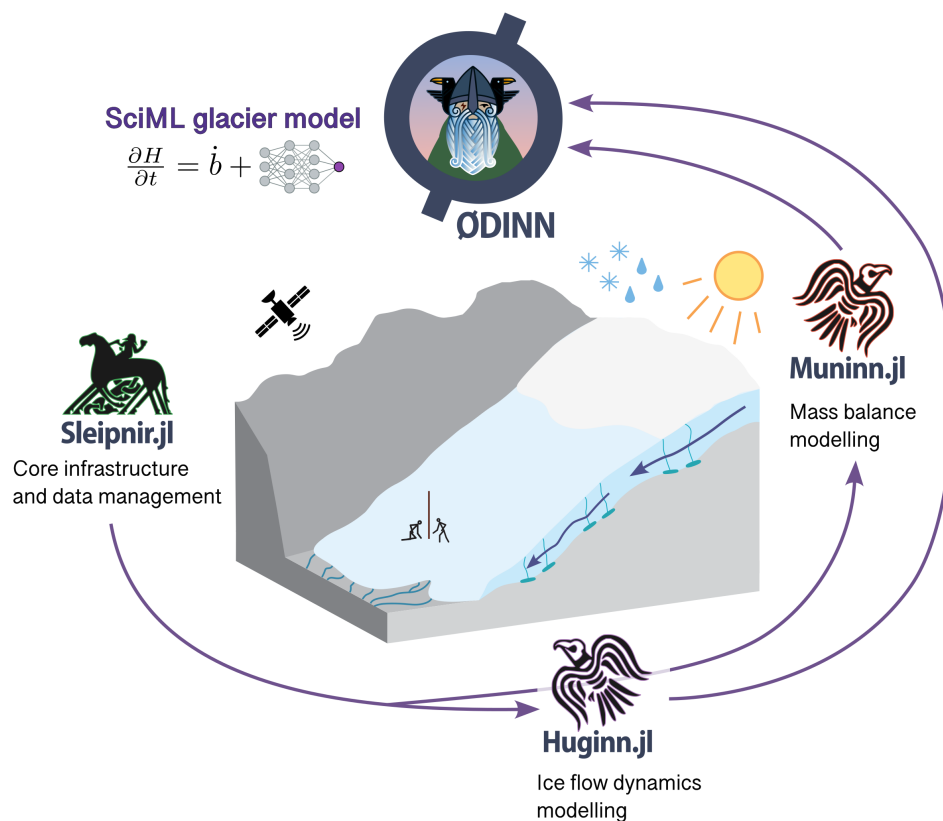


Figure 1: Overview of the ODINN.jl ecosystem.

Software design

96 ODINN.jl is in fact an ecosystem composed of multiple packages, each one handling a specific
97 task:

- 98 ▪ Sleipnir.jl: Handles all the basic types, functions and datasets, common through the
- 99 whole ecosystem, as well as data management tasks.
- 100 ▪ Muninn.jl: Handles surface mass balance processes, via different types of models.
- 101 ▪ Huginn.jl: Handles ice flow dynamics, by solving the ice flow partial differential equations
- 102 (PDEs) using numerical methods. It can accommodate multiple types of ice flow models.
- 103 ▪ ODINN.jl: Acts as the interface to the whole ecosystem, and provides the necessary tools
- 104 to differentiate and optimize any model component. It can be seen as the SciML layer,
- 105 enabling different types of inverse methods, using hybrid models combining differential
- 106 equations with data-driven models.
- 107

108 Splitting large Julia (Bezanson et al., 2017) packages into smaller, focused subpackages is
109 a good practice that enhances maintainability, usability, and collaboration. Modular design
110 simplifies debugging, testing, and updates by isolating functionalities, while users benefit from
111 faster precompilation and reduced memory overhead by loading only the subpackages they
112 need. This approach also lowers the barrier for new contributors, fosters clearer dependency
113 management, and ensures scalability as projects grow, ultimately creating a robust and
114 adaptable software ecosystem. The ODINN ecosystem extends beyond this suite of Julia

115 packages, by leveraging the data preprocessing tools of OGGM. We do so via an auxiliary
116 Python library named Gungnir, which is responsible for generating all the necessary data to
117 force and initialize the model, such as glacier outlines from the Randolph Glacier Inventory
118 (RGI Consortium (2023), RGI), digital elevation models (DEMs), ice thickness observations
119 from GlaThiDa (Consortium, 2020), ice surface velocities from different studies (Millan et al.,
120 2022), and different sources of climate reanalyses and projections (Eyring et al., 2016; Lange,
121 2019). This implies that ODINN.jl, like OGGM, is virtually capable of simulating any of the
122 ~274,000 glaciers on Earth (RGI Consortium, 2023).

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

134 Research impact statement

135 ODINN.jl has evolved through the last 5 years with code contributions during 3 postdoc
136 positions, 1 PhD and 3 master internships. It has so far been used to explore the use of UDEs
137 to invert hidden empirical laws in a synthetic glacier setup, where a prescribed rheological
138 law was successfully recovered using a neural network (Bolibar et al., 2023). This proof-of-
139 concept then served as a backbone to create the current complex architecture of ODINN.jl,
140 finalized with the recent 1.0 release. The main changes and scientific goals of this large
141 software development investment, are the capacity to now apply these methods to large-scale
142 remote sensing data for multiple glaciers, which will enable the exploration of new glacier
143 basal sliding laws directly from heterogeneous observations, which remains a long-standing
144 problem in glaciology (Minchew & Joughin, 2020). The development of the differentiable
145 programming methods in ODINN.jl, also served as a catalyst to write an exhaustive review
146 paper, together with other key players in this community, on differentiable programming for
147 differential equations (Sapienza et al., 2024).

148 Additionally, ODINN.jl will be soon used as part of a newly funded 4-year project, to simulate
149 past and future glacier changes in several catchments in the Andes and the Alps. These model
150 outputs will then be combined with a hydrological model, to investigate the impacts of glacier
151 retreat on the hydrological regimes and drought mitigation under different climate change
152 scenarios.

153 With these two research venues, ODINN.jl will continue to be developed and used to pursue
154 both fundamental research on glacier sliding laws, and applied research to assess the impact of
155 glacier retreat on freshwater availability and drought mitigation.

156 AI usage disclosure

157 Generative AI, via GitHub copilot, has been used to partially generate some of the docstrings
158 for the documentation, and to assist in the coding of some tests and helper functions.

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. We thank all the developers of the SciML Julia ecosystem who work in each one of the core libraries used within ODINN.jl. 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 and CYL were supported by NSF via grant number OPP-2441132 and the Alfred P. Sloan Foundation under grant number FG-2024-21649. FS and FP 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., Rabatel, A., Gouttevin, I., Zekollari, H., & Galiez, C. (2022). Nonlinear sensitivity of glacier mass balance to future climate change unveiled by deep learning. *Nature Communications*, 13(1), 409. <https://doi.org/10.1038/s41467-022-28033-0>
- 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>
- Bueler, E., Lingle, C. S., Kallen-Brown, J. A., Covey, D. N., & Bowman, L. N. (2005). Exact solutions and verification of numerical models for isothermal ice sheets. *Journal of Glaciology*, 51(173), 291–306. <https://doi.org/10.3189/172756505781829449>
- Combemale, B., Gray, J., & Rumpe, B. (2023). Research software engineering and the importance of scientific models. *Software and Systems Modeling*, 22(4), 1081–1083. <https://doi.org/10.1007/s10270-023-01119-z>
- Consortium, G. (2020). *Glacier Thickness Database 3.1.0*. 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>
- Gagliardini, O., Zwinger, T., Gillet-Chaulet, F., Durand, G., Favier, L., Fleurian, B. de, Greve, R., Malinen, M., Martín, C., Råback, P., Ruokolainen, J., Sacchetti, M., Schäfer, M., Seddik, H., & Thies, J. (2013). Capabilities and performance of Elmer/Ice, a new-generation ice sheet model. *Geoscientific Model Development*, 6(4), 1299–1318. <https://doi.org/10.5194/gmd-6-1299-2013>
- Hutter, K. (1983). *Theoretical Glaciology*. Springer Netherlands. <https://doi.org/10.1007/978-94-015-1167-4>
- IPCC. (2021). *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change: Vols. In Press*. Cambridge University Press. <https://doi.org/10.1017/9781009157896>
- Lange, S. (2019). *WFDE5 over land merged with ERA5 over the ocean (W5E5)*. GFZ Data Services. <https://doi.org/10.5880/PIK.2019.023>

- 205 Maussion, F., Butenko, A., Champollion, N., Dusch, M., Eis, J., Fourteau, K., Gregor, P.,
206 Jarosch, A. H., Landmann, J., Oesterle, F., Recinos, B., Rothenpieler, T., Vlug, A., Wild, C.
207 T., & Marzeion, B. (2019). The Open Global Glacier Model (OGGM) v1.1. *Geoscientific*
208 *Model Development*, 12(3), 909–931. <https://doi.org/10.5194/gmd-12-909-2019>
- 209 Millan, R., Mouginot, J., Rabatel, A., & Morlighem, M. (2022). Ice velocity and thickness of
210 the world's glaciers. *Nature Geoscience*, 15(2), 124–129. [https://doi.org/10.1038/s41561-](https://doi.org/10.1038/s41561-021-00885-z)
211 [021-00885-z](https://doi.org/10.1038/s41561-021-00885-z)
- 212 Minchew, B., & Joughin, I. (2020). Toward a universal glacier slip law. *Science*, 368(6486),
213 29–30. <https://doi.org/10.1126/science.abb3566>
- 214 Moses, W. S., Churavy, V., Paehler, L., Hükelheim, J., Narayanan, S. H. K., Schanen, M.,
215 & Doerfert, J. (2021). Reverse-mode automatic differentiation and optimization of GPU
216 kernels via enzyme. *Proceedings of the International Conference for High Performance*
217 *Computing, Networking, Storage and Analysis*. <https://doi.org/10.1145/3458817.3476165>
- 218 Nyenah, E., Döll, P., Katz, D. S., & Reinecke, R. (2024). Software sustainability of global
219 impact models. *Geoscientific Model Development Discussions*, 2024, 1–29. [https://doi.](https://doi.org/10.5194/gmd-2024-97)
220 [org/10.5194/gmd-2024-97](https://doi.org/10.5194/gmd-2024-97)
- 221 Rackauckas, Chris, Innes, M., Ma, Y., Bettencourt, J., White, L., & Dixit, V. (2019).
222 DiffEqFlux.jl - A Julia Library for Neural Differential Equations. *arXiv:1902.02376 [Cs,*
223 *Stat]*. <http://arxiv.org/abs/1902.02376>
- 224 Rackauckas, Christopher, Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner,
225 D., Ramadhan, A., & Edelman, A. (2021). *Universal Differential Equations for Scientific*
226 *Machine Learning*. arXiv. <https://doi.org/10.48550/arXiv.2001.04385>
- 227 Rackauckas, Christopher, & Nie, Q. (2017). DifferentialEquations.jl – A Performant and
228 Feature-Rich Ecosystem for Solving Differential Equations in Julia. *Journal of Open*
229 *Research Software*, 5, 15. <https://doi.org/10.5334/jors.151>
- 230 RGI Consortium. (2023). *Randolph Glacier Inventory - A Dataset of Global Glacier Outlines,*
231 *Version 7*. National Snow; Ice Data Center. <https://doi.org/10.5067/F6JMOVY5NAVZ>
- 232 Rounce, D. R., Hock, R., Maussion, F., Hugonnet, R., Kochtitzky, W., Huss, M., Berthier,
233 E., Brinkerhoff, D., Compagno, L., Copland, L., Farinotti, D., Menounos, B., & McNabb,
234 R. W. (2023). Global glacier change in the 21st century: Every increase in temperature
235 matters. *Science*, 379(6627), 78–83. <https://doi.org/10.1126/science.abo1324>
- 236 Sapienza, F., Bolibar, J., Schäfer, F., Groenke, B., Pal, A., Boussange, V., Heimbach,
237 P., Hooker, G., Pérez, F., Persson, P.-O., & Rackauckas, C. (2024). *Differentiable*
238 *Programming for Differential Equations: A Review*. arXiv. [http://arxiv.org/abs/2406.](http://arxiv.org/abs/2406.09699)
239 [09699](http://arxiv.org/abs/2406.09699)
- 240 Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi,
241 M., Fenicia, F., Kifer, D., Li, L., Liu, X., Ren, W., Zheng, Y., Harman, C. J., Clark, M.,
242 Farthing, M., Feng, D., Kumar, P., Aboelyazeed, D., ... Lawson, K. (2023). Differentiable
243 modelling to unify machine learning and physical models for geosciences. *Nature Reviews*
244 *Earth & Environment*, 1–16. <https://doi.org/10.1038/s43017-023-00450-9>
- 245 Sjursen, K. H., Bolibar, J., Meer, M. van der, Andreassen, L. M., Biesheuvel, J. P., Dunse,
246 T., Huss, M., Maussion, F., Rounce, D. R., & Tober, B. (2025). Machine learning
247 improves seasonal mass balance prediction for unmonitored glaciers. *EGU sphere*, 1–39.
248 <https://doi.org/10.5194/egusphere-2025-1206>
- 249 Weis, M., Greve, R., & Hutter, K. (1999). Theory of shallow ice shelves. *Continuum Mechanics*
250 *and Thermodynamics*, 11(1), 15–50. <https://doi.org/10.1007/s001610050102>