

SICOPOLIS-AD v2: tangent linear and adjoint modeling framework for ice sheet modeling enabled by automatic differentiation tool Tapenade

Shreyas Sunil Gaikwad^{1¶}, Laurent Hascoet², Sri Hari Krishna Narayanan³, Liz Curry-Logan¹, Ralf Greve^{4,5}, and Patrick Heimbach^{1,6,7}

¹ Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, USA ² Institut National de Recherche en Informatique et Automatique, France ³ Mathematics and Computer Science Division, Argonne National Laboratory, USA ⁴ Institute of Low Temperature Science, Hokkaido University, Japan ⁵ Arctic Research Center, Hokkaido University, Japan ⁶ Jackson School of Geosciences, University of Texas at Austin, USA ⁷ Institute for Geophysics, University of Texas at Austin, USA [¶] 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Published: unpublished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

We present a new framework for generating derivative code, i.e., tangent linear, adjoint, or Hessian models, of the open-source Simulation COde for POLythermal Ice Sheets ([SICOPOLIS](#)). These derivative operators are powerful computational engines to efficiently compute comprehensive gradients or sensitivities of scalar-valued model output, including least-squares model-data misfits or important quantities of interest, to high-dimensional model inputs (such as model initial conditions, parameter fields, or boundary conditions). The new version 2 ([SICOPOLIS-AD v2](#)) framework is based on the source-to-source automatic differentiation (AD) tool [Tapenade](#) which has recently been open-sourced. The switch from a previous AD tool (OpenAD) used in SICOPOLIS-AD version 1 to Tapenade overcomes several limitations outlined here. The framework is integrated with the SICOPOLIS model's main trunk and is freely available.

Statement of need

The two contemporary ice sheets, Greenland and Antarctica, are dynamic entities whose evolution is governed by a set of nonlinear partial differential equations (PDEs) that describe the conservation of mass, momentum, and energy, as well as constitutive laws for the material properties of ice. In general, these equations cannot be solved analytically but must be solved numerically. Ice sheet models are a computer representation of these PDEs. They require as input parameters (i) initial conditions of the state of the ice sheet, (ii) surface boundary conditions, such as precipitation, (iii) basal boundary conditions, such as geothermal flux, and (iv) model parameters, such as flow law parameters. Despite advances in numerical modeling of ice sheets, the effects of ad-hoc initialization and the uncertainties in these independent input parameters propagate to quantities of interest (QoI), such as future projections of sea-level rise, which is of economic and societal importance ([Schinko et al., 2020](#)). It is thus desirable to evaluate the sensitivities of our QoI to these independent input variables.

In the context of ice sheet modeling, sensitivities of model-data misfits or other QoI are a key ingredient for performing model calibration, state estimation, or uncertainty quantification (UQ), which guide the improvement of model simulations through PDE-constrained gradient-based optimization.

SICOPOLIS-AD v2 leverages the recently open-sourced AD tool Tapenade (Hascoet & Pascual, 2013) to generate code for the adjoint model of the open-source ice sheet model, SICOPOLIS (Greve, 1997; Greve et al., 2011; Greve & Blatter, 2009). Sensitivities can be calculated using a single forward and adjoint model evaluation, instead of the $\mathcal{O}(N)$ forward model evaluations. Empirically, one adjoint model evaluation is about 5-10 times as expensive as a forward model run. The adjoint computation is highly efficient for calculating sensitivities when N is large (typically, $N \sim 10^4 - 10^6$).

The functionality to generate a tangent linear version of the forward model is also included, which was not available in SICOPOLIS-AD v1. This is valuable for UQ of the inferred parameters, as well as uncertainty propagation to Qols. It can also be used to verify the results of the adjoint model.

Target Audience

This package is intended as a resource that enables sensitivity analysis, model calibration, and uncertainty quantification of a continent-scale ice sheet model. Our package is also intended to serve as a guide for future work in the application of open-source AD tools for physics-based simulation codes written in Fortran.

Features

AD tools such as the commercial TAF (Giering & Kaminski, 1999) and the open-source OpenAD (Utke et al., 2008) have been used previously with SICOPOLIS (Heimbach & Bugnion, 2009; Logan et al., 2020). OpenAD is no longer actively developed because it is based on the Open64 compiler which ceased development in 2011. The differentiation of SICOPOLIS, therefore, must be performed using a different tool. Compared to OpenAD, the Tapenade enabled implementation has the following advantages:

- It is up-to-date with the latest SICOPOLIS code.
- The AD tool Tapenade is open-source and actively maintained.
- A new tangent linear code generation capability is introduced.
- The AD-generated codes can accept NetCDF inputs.
- The external library LIS, its tangent linear code, and adjoint code are correctly incorporated which can improve the simulation of Antarctic ice shelves and Greenland outlet glaciers.
- Gitlab-CI, a Docker, and the pytest framework are leveraged for Continuous Integration (CI) to track changes in the trunk that “break” the AD-based code generation.
- The entire code is parsed by Tapenade, preventing cumbersome manual maintenance of subroutines to initialize the adjoint runs.
- Python scripts are provided for quick setup of the compilation, I/O, and execution processes based on user-provided metadata.
- The setup is well-documented, along with tutorials.

Software requirements and external usage

SICOPOLIS-AD v2 is built on top of the ice sheet model SICOPOLIS and uses Tapenade to differentiate this model. All the prerequisites of using SICOPOLIS and Tapenade need to be satisfied. A Python installation is needed to use the automation tools.

Example

We illustrate the use of our tool with the example of a steady state simulation of the Greenland ice sheet under modern climate conditions. The corresponding SICOPOLIS configuration header file, `v5_grl16_bm4_ss25ka`, is provided as a reference template in the standard SICOPOLIS distribution. We shorten the total integration time to 100 simulated years to keep the computational cost of the tangent linear and finite differences reasonable. Our Qol (i.e., dependent variable) is the total volume of the ice sheet at the end of the run (fc). The sensitivity is evaluated with respect to the geothermal heat flux, q_{geo} (independent variable), a 19,186-dimensional field. The results are shown in Figure 1.

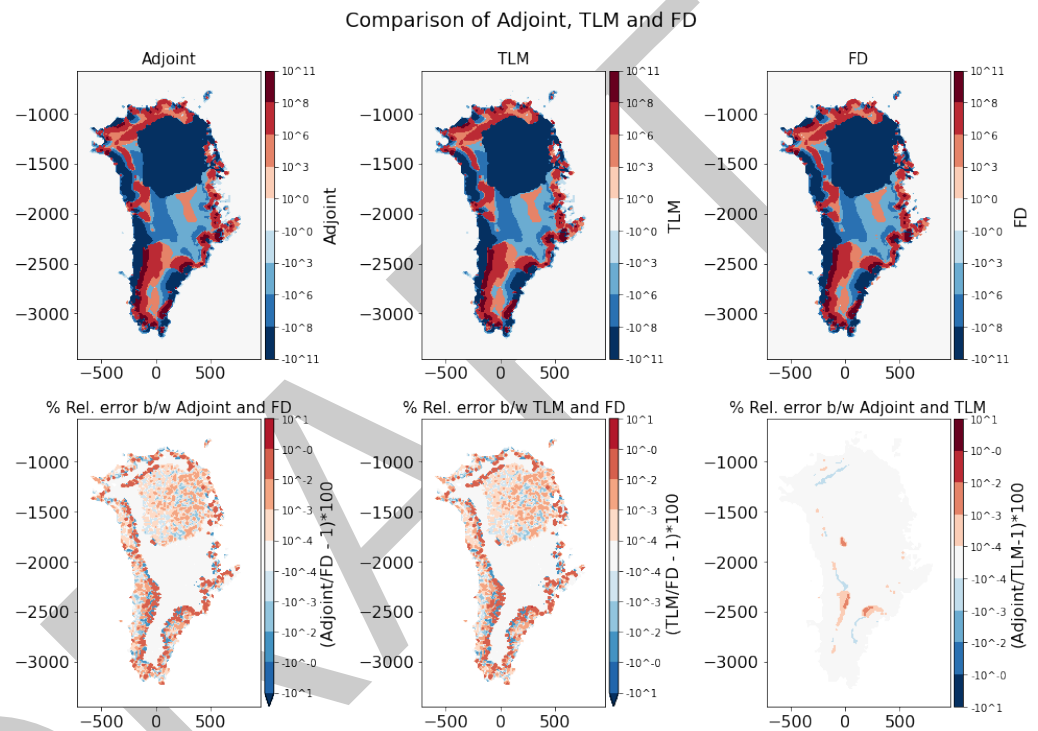


Figure 1: Validation exercise for adjoint (ADM) and tangent linear (TLM) models using the finite differences (FD) results for the sensitivity of fc with respect to q_{geo} . The upper row shows the sensitivities computed using the adjoint model (reverse-mode AD), tangent linear (forward-mode AD), and finite differences, respectively. The bottom row illustrates the relative error between (ADM, FD), (TLM, FD), and (ADM, TLM) respectively. For the bottom row, note that the values of relative error are only shown for points where the value of the gradient is “significant”, i.e. within 4 orders of magnitude of the maximum absolute value of the gradient.

The results show good agreement between all three modes used to evaluate this sensitivity. The error is less than 4.5% between AD-generated (adjoint/tangent linear codes) and finite differences at all but one point with “significant” gradient values, i.e. within 4 orders of magnitude of the maximum absolute value of the finite differences gradient. The relative error between the AD-generated adjoint and tangent linear models is less than 0.004% at all points with values within 4 orders of magnitude of the maximum absolute value of the finite differences gradient. However, the adjoint model is much faster than the other two, as shown in Table 1, because the number of evaluations of the latter two scales linearly with the parameter dimension ($\sim \mathcal{O}(N)$). The discrepancy will be even larger if a finer mesh is used.

Table 1: Comparison of the time taken by various methods to evaluate the gradient for a scalar objective function with respect to a 19,186-dimensional 2D field (16 km mesh) in a typical SICOPOLIS run. The runs are performed on Intel Xeon CPU E5-2695 v3 nodes (2.30 GHz clock rate, 35.84 MB L3 cache, 63.3 GB memory).

Gradient calculation method	Time (in seconds) for 16 km mesh
Finite Differences	1.645×10^5
Tangent Linear Model	2.140×10^4
Adjoint Model	2.113×10^1

Acknowledgements

This work was supported by the Applied Mathematics activity within the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research Program, under contract number DE-AC02-06CH11357, and by National Science Foundation OPP/P2C2 grant #1903596.

Copyright

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ('Argonne'). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. <http://energy.gov/downloads/doe-public-access-plan>.

References

- Giering, R., & Kaminski, T. (1999). Recipes for adjoint code construction. *ACM trans. Math. software. ACM Transactions on Mathematical Software*, 24. <https://doi.org/10.1145/293686.293695>
- Greve, R. (1997). Application of a polythermal three-dimensional ice sheet model to the greenland ice sheet: Response to steady-state and transient climate scenarios. *Journal of Climate*, 10(5), 901–918. [https://doi.org/10.1175/1520-0442\(1997\)010%3C0901:AOAPTD%3E2.0.CO;2](https://doi.org/10.1175/1520-0442(1997)010%3C0901:AOAPTD%3E2.0.CO;2)
- Greve, R., & Blatter, H. (2009). Dynamics of ice sheets and glaciers. *Dynamics of Ice Sheets and Glaciers by Ralf Greve*. <https://doi.org/10.1007/978-3-642-03415-2>
- Greve, R., Saito, F., & Abe-Ouchi, A. (2011). Initial results of the SeaRISE numerical experiments with the models SICOPOLIS and IcIES for the greenland ice sheet. *Annals of Glaciology*, 52(58), 23–30. <https://doi.org/10.3189/172756411797252068>
- Hascoet, L., & Pascual, V. (2013). The Tapenade Automatic Differentiation tool: principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3). <https://doi.org/10.1145/2450153.2450158>
- Heimbach, P., & Bugnion, V. (2009). Greenland ice-sheet volume sensitivity to basal, surface and initial conditions derived from an adjoint model. *Annals of Glaciology*, 50(52), 67–80. <https://doi.org/10.3189/172756409789624256>

- 133 Logan, L. C., Narayanan, S. H. K., Greve, R., & Heimbach, P. (2020). SICOPOLIS-AD
134 v1: An open-source adjoint modeling framework for ice sheet simulation enabled by
135 the algorithmic differentiation tool OpenAD. *Geoscientific Model Development*, 13(4),
136 1845–1864. <https://doi.org/10.5194/gmd-13-1845-2020>
- 137 Schinko, T., Drouet, L., Vrontisi, Z., Hof, A., Hinkel, J., Mochizuki, J., Bosetti, V., Fragkiadakis,
138 K., Vuuren, D. van, & Lincke, D. (2020). Economy-wide effects of coastal flooding
139 due to sea level rise: A multi-model simultaneous treatment of mitigation, adaptation,
140 and residual impacts. *Environmental Research Communications*, 2(1), 015002. <https://doi.org/10.1088/2515-7620/ab6368>
141
- 142 Utke, J., Naumann, U., Fagan, M., Tallent, N., Strout, M., Heimbach, P., Hill, C., & Wunsch,
143 C. (2008). OpenAD/f: A modular open-source tool for automatic differentiation of fortran
144 codes. *ACM Trans. Math. Softw.*, 34(4). <https://doi.org/10.1145/1377596.1377598>

DRAFT