plopm: A lightweight and flexible tool for visualization and postprocessing of OPM Flow geological models

David Landa-Marbán*[1]

[1]NORCE Research AS, Bergen, Norway

16 February 2026

Abstract

Reservoir simulations help the energy industry understand how underground pressure affects the movement of fluids such as water, oil, or gas and how it influences the stability of the surrounding rocks. These insights support better decisions in areas like energy production, carbon storage, and geothermal development. An important step in working with simulation results is postprocessing, which involves visualizing and analyzing the data. `plopm` is a tool designed to make this step faster, easier, and more consistent. `plopm` is operated through the command line, where users specify their postprocessing tasks using simple flags. This approach supports reproducibility and encourages efficient workflows. `plopm` is highly efficient, able to postprocess models containing hundreds of millions of cells in only a few tens of seconds. The tool offers several ways to visualize geological models. For example, users can generate 2D maps and choose how values are projected along a particular dimension, such as taking the maximum, minimum, or a pore-volume-weighted average. These options make it easier to compare scenarios and understand key patterns in the subsurface. Although `plopm` was originally intended to produce PNGs, GIFs, and VTK files from OPM Flow simulations, it has since grown more flexible. It can now also process results stored in CSV files, broadening its usefulness beyond the standard OPM Flow output format. The graphical abstract is shown in Fig. 1.

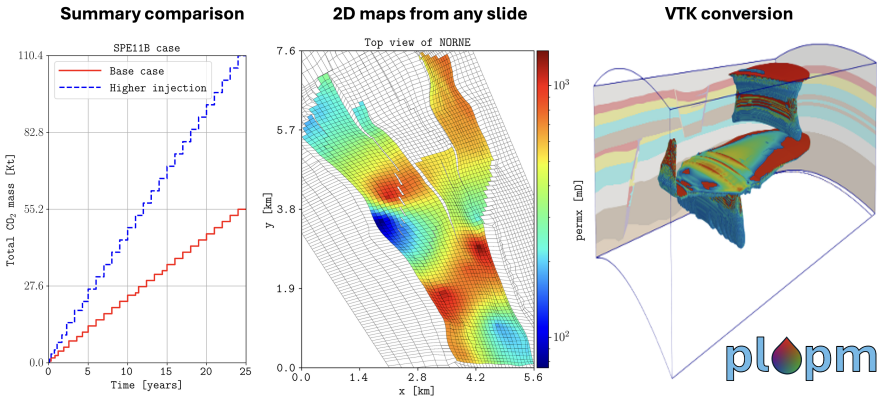

Figure 1: Graphical representation of `plopm`'s functionality. See the online documentation for reproduction details.

*Corresponding author. ORCID: 0000-0002-3343-1005

1 Statement of need

OPM Flow is an open-source simulator for subsurface applications such as CO_2 storage, hydrogen storage, and hydrocarbon production [Rasmussen et al., 2021]. There is growing demand for tools that make it easier to visualize and analyze simulation results, particularly for large-scale models. `plopm` is a user-friendly Python tool designed to support this need by providing efficient visualization and postprocessing of OPM Flow outputs. It is intended for researchers, engineers, and students who need to quickly interpret simulation results and compare scenarios. `plopm` offers flexible 2D visualization options, allowing users to choose among several projection methods depending on their analysis goals. This flexibility is especially relevant in current studies where rapid exploration of model behavior is essential. A key motivation behind `plopm` is to improve reproducibility, an increasingly important requirement in scientific research and technical reporting. Each generated figure can be reproduced by using a clear set of command-line flags, ensuring that results can be regenerated, shared, and further processed in a consistent and transparent manner.

2 State of the field

ParaView is a widely used open-source post-processing and visualization tool written in C++. It offers a comprehensive graphical user interface and excels at interactive 3D rendering, slicing, contouring, and animating simulation results. However, ParaView can have a steep learning curve, may experience performance and stability limitations, and is demanding in terms of memory and setup.

ResInsight is an open-source C++ tool designed specifically for postprocessing reservoir simulations, including Eclipse and OPM Flow models. It performs well when analyzing large datasets and visualizing grids, wells, and dynamic properties. Nevertheless, ResInsight can face performance issues with very large models, and importing models with extensive fault information may be time-consuming. Users on macOS may also encounter installation challenges. In addition, the software struggles with models that have extremely small cell dimensions (below 1 mm) or very large cell counts (greater than 100 million).

While both ResInsight and ParaView are excellent tools for 3D visualization, lower-dimensional tasks (2D and 1D) often carry unnecessary overhead when performed through full-featured visualization environments. In such cases, Python scripts can significantly streamline the postprocessing workflow. To the author's knowledge, prior to the development of `plopm` there was no integrated, well-documented Python-based tool capable of generating PNGs, GIFs, and VTKs from OPM Flow output files using command-line flags. Python provides a more accessible and flexible environment than C++, lowering the entry barrier for researchers, engineers, and students who require lightweight and customizable tools for model analysis.

The VTK functionality in `plopm` was originally implemented to enable 3D visualization of a SPE11C benchmark model, which contained more than 100 million cells. This capability made it possible to generate visualizations such as this GIF. The motivation for developing this feature was the limited VTK support available in OPM Flow. In particular, several useful quantities are not exported (e.g., FIPNUM, FLORES), and the simulator writes the entire grid at every report step, leading to extremely large output files. In contrast, the VTK functionality in `plopm` allows users to efficiently extract selected data directly from OPM Flow output files (e.g., UNRST, INIT, EGRID) and convert them into lightweight VTK structures suitable for 3D visualization.

`plopm` offers a variety of options for visualizing simulation results, allowing users to compare methods and select the one that best fits their needs. For additional details about the capabilities

included in plopm, see the examples in the project documentation.

3 Software design

`plopm` leverages well-established and widely used Python libraries. The Matplotlib package [Hunter, 2007] forms the core of the plotting routines. NumPy [Harris et al., 2020] provides the foundational array operations used throughout the tool, and SciPy [Virtanen et al., 2020] supports interpolation tasks as well as the computation of normal and lognormal distributions used in ensemble visualizations. To display progress and estimate remaining runtime during postprocessing, the alive-progress package [de Moura, 2024] is employed. Parsing of OPM Flow binary output files is handled through the opm library.

The primary methods implemented in `plopm` include handling of corner-point grids, parsing input decks (for example, to visualize wells and faults), enforcing homogenized colorbar limits to compare multiple models within a single figure, mapping OPM Flow output to VTK format, and stacking filter and cell-value operations.

Interaction with the tool is performed via the terminal through an executable named `plopm`, which exposes a collection of command-line flags (70 at the time of writing; see the online documentation for the current list at current list). These flags control key functionality such as specifying input models, defining the desired postprocessing operations, and selecting output file names. This design enables users to apply consistent postprocessing settings across different simulations. Advanced users familiar with Python can also access the underlying modules directly, allowing integration into more complex workflows and customization of model transformations to meet specific research or engineering needs.

4 Research impact statement

`plopm` is already being adopted across several projects at NORCE Research AS. GitHub traffic insights also show a steady increase in repository clones, indicating broader interest and growing usage within the community.

The software has supported several research publications, including:

- Landa-Marbán et al. [2024], where it was used to generate 2D maps of static properties and dynamic quantities such as pressure, salinity, and CO_2 saturation in synthetic 2D radial geometries.
- Landa-Marbán et al. [2026a], for producing 2D maps of static properties and CO_2 mass for the FluidFlower/SPE11A models.
- Landa-Marbán et al. [2026b], for comparing 2D maps across different grid resolutions of the SPE11B benchmark model.
- Landa-Marbán et al. [2025], for generating 2D maps of static properties and pore-volume-weighted average pressure for the Troll aquifer model.

The software also supports the online documentation of several open-source tools, including:

- expreccs: A Python framework using OPM Flow to simulate regional and site reservoirs for CO_2 storage.

- pofff: An image-based history-matching framework for the FluidFlower Benchmark using OPM Flow.
- pycopm: An open-source tool to tailor OPM Flow geological models.
- pyopmnearwell: A Python framework to simulate near well dynamics using OPM Flow.
- pyopmspe11: A Python framework using OPM Flow for the CSP SPE11 benchmark project.

Significant effort has been dedicated to developing comprehensive online documentation that enables users to reproduce figures, tables, and computational workflows from recent publications. This work is motivated by the FAIR principles (Findable, Accessible, Interoperable, Reusable) [Wilkinson et al., 2016], which have not been consistently implemented in subsurface research in recent years [Liu et al., 2026]. For example, the Publication, TCCS-13, and Convergence documentation includes step-by-step terminal commands (such as `plopm`) required to generate the results published in Landa-Marbán et al. [2026a], Landa-Marbán et al. [2025], and Landa-Marbán et al. [2026b], respectively. These pages include the exact terminal commands (such as those invoking `plopm`) needed to regenerate the published outputs. This ensures that the scientific results are not only transparent but also directly reusable by other researchers, thereby enhancing methodological rigor and accelerating future developments.

5 AI usage disclosure

No generative AI tools were used in the development of this software. Microsoft M365 Copilot (powered by a GPT-5–class large language model developed by Microsoft) was used to check and improve the writing of this manuscript.

Acknowledgements

The author acknowledges funding from the Center for Sustainable Subsurface Resources (CSSR), grant nr. 331841, supported by the Research Council of Norway, research partners NORCE Norwegian Research Centre and the University of Bergen, and user partners Equinor ASA, Harbour Energy, Sumitomo Corporation, Earth Science Analytics, GCE Ocean Technology, and SLB Scandinavia. The author also acknowledges funding from the Expansion of Resources for CO_2 Storage on the Horda Platform (ExpReCCS) project, grant nr. 336294, supported by the Research Council of Norway, Equinor ASA, A/S Norske Shell, and Harbour Energy Norge AS.

References

Atgeirr Flø Rasmussen, Tor Harald Sandve, Kai Bao, Andreas Lauser, Joakim Hove, Bård Skaflestad, Robert Klöfkorn, Markus Blatt, Alf Birger Rustad, Ove Sævareid, Knut-Andreas Lie, and Andreas Thune. The open porous media flow reservoir simulator. *Computers & Mathematics with Applications*, 81:159–185, 2021. ISSN 0898-1221. doi: 10.1016/j.camwa.2020.05.014. URL https://www.sciencedirect.com/science/article/pii/S0898122120302182.

John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9 (3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020. doi: 10.1038/s41586-020-2649-2.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Rafael Salgado de Moura. alive-progress: A new kind of progress bar, with real-time throughput, eta and very cool animations. https://github.com/rsalmei/alive-progress, 2024. Accessed: 2026-02-16.

D. Landa-Marbán, N. Zamani, T. H. Sandve, and S. E. Gasda. Impact of intermittency on salt precipitation during CO2 injection. SPE Norway Subsurface Conference:D011S012R010, 04 2024. doi: 10.2118/218477-MS. URL https://doi.org/10.2118/218477-MS.

D. Landa-Marbán, Tor H. Sandve, Jakub W. Both, Jan M. Nordbotten, and Sarah E. Gasda. Performance of an open-source image-based history matching framework for CO2 storage. *Transport in Porous Media*, 153(2):21, 2026a. doi: 10.1007/s11242-025-02275-0. URL https://doi.org/10.1007/s11242-025-02275-0.

D. Landa-Marbán, K.-A. Lie, K. O. Lye, O. Møyner, A. F. Rasmussen, and T. H. Sandve. Exploring convergence and its limits in case b of the 11th spe comparative solution project. *SPE Journal*, pages 1–12, 01 2026b. ISSN 1086-055X. doi: 10.2118/231853-PA. URL https://doi.org/10.2118/231853-PA.

D. Landa-Marbán, T. H. Sandve, and S. E. Gasda. A coarsening approach to the troll aquifer model, 2025. URL https://arxiv.org/abs/2508.08670.

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, 2016. doi: 10.1038/sdata.2016.18. URL https://doi.org/10.1038/sdata.2016.18.

Na Liu, Jakub Wiktor Both, Geir Ersland, Jan Martin Nordbotten, and Martin A. Fernø. Trends in porous media laboratory imaging and open science practices, 2026. ISSN 0001-8686. URL https://www.sciencedirect.com/science/article/pii/S0001868626000564.