















OGSTools

Tobias Meisel ^{1*}, Florian Zill ^{1,2*}, Julian Heinze ^{1*}, Lars Bilke ^{1*}, Max Jäschke ^{3*}, Feliks Kiszkurno ^{1*}, Dominik Kern ^{2*}, and Jörg Buchwald ^{1*}

¹ Helmholtz Centre for Environmental Research, Germany  ² TU Bergakademie Freiberg, Germany  ³ Leipzig University of Applied Sciences, Germany  * These authors contributed equally.

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Published: unpublished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

OGSTools (OpenGeoSys Tools) is an open source (3-Clause BSD) Python library for streamlined usage of OpenGeoSys 6 (OGS), also an open-source software [Bilke et al. (2025)] for simulating thermo-hydro-mechanical-chemical (THMC) processes in porous and fractured media [Kolditz et al. (2012)]. OGSTools [Zill et al. (2025)] provides an interface between OGS-specific data and well-established data structures of the Python ecosystem, as well as domain-specific solutions, examples, and tailored defaults for OGS users and developers. The libraries' functionalities are designed to be used in complex automated workflows (including pre- and post-processing), the OGS benchmark gallery, the OGS test-suite, and in automating repetitive tasks in the model development cycle.

Statement of need

Development efficiency

Modelers of OGS iteratively run simulations, analyse results, and refine their models with the goal to enhance the accuracy, efficiency and reliability of the simulation results. To improve efficiency, repetitive tasks in the model development cycle should be formalized and automated through computer programs. Python is a great choice for developer efficiency and scientific data analysis due to its simple syntax and vast ecosystem of libraries. The Python library introduced here is needed as a central platform to collect and improve common functionalities.

Complex workflows

A workflow is a structured sequence of tasks, that processes data and executes computations to achieve a specific goal [Diercks et al. (2022)]. In our scientific research, workflows need to integrate multiple steps—such as geological data preprocessing, ensemble simulations with OGS, domain specific analysis and visualization—into complex and fully automated and therefore reproducible sequences. Typically, one specific workflow is implemented to answer one specific scientific question. Workflow-based approaches have been proofed to adhere to the FAIR principles [Goble et al. (2020)], [Wilkinson et al. (2025)]. In our approach we use existing workflow management software that covers domain independent parts like dependency graph description, computational efficiency, data management, execution control, multi-user collaboration and data provenance [(?)]. Since we are building on the Python ecosystem, using Python-based workflow managers such as Snakemake [Köster & Rahmann (2012)] and AiIDA [Huber et al. (2020)] is a natural choice.

The usage of workflow managers shifts the actual development to the workflow components. Common and frequently used functionality found within workflow components are made reusable and provided as a Python library their repeated use in future workflows makes

workflow development more efficient. The library focuses on functionalities directly related to (1) the OGS core simulator and its specific input and output data, (2) domain-specific definitions in geo-science, (3) finite element modeling (FEM), and (4) numerical computation. Our workflow components are then built of generic Python libraries, our library and workflow manager dedicated integration code. To ensure integration of the library's functionalities with the workflow management software, a list of functional and non-functional requirements (e.g., thread safety), imposed by all workflow management software we intend to use, are maintained and regularly validated through continuous testing.

Code example

Test suite

Centralization of Python-related development

Previously, our code base for Python-related tasks in OGS was fragmented, with components, that were often developed for specific use cases, with varying degrees of standardization and sharing. The lack of centralization led to inefficiencies, inconsistent quality, and challenges in maintaining and extending the code. With OGSTools, all Python-related code is now centralized under the professional maintenance of the OGS core developer team. It ensures better collaboration, standardized practices and improved code quality. Further it enables the transfer of years of experience in maintaining the OGS core [Bilke et al. (2019)] to the pre- and post-processing code. For the centralized approach, preceding work from msh2vtu [Kern & Bilke (2022)], ogs6py and VTUInterface [Buchwald et al. (2021)] and extracted functionalities from the projects (1) AREHS [Meisel et al. (2024)], and (2) OpenWorkFlow - Synthesis Platform [Environmental Research UFZ (2023)] have been adapted and integrated into OGSTools.

We recommend our users to setup a project specific (virtual) python environment or conda environment and freeze the dependencies. Otherwise these Python environments follow a rolling-release model, in which dependencies are updated continuously without centralized coordination, potentially leading to unexpected behavior or incompatibilities. To address The Need for a Versioned Data Analysis Software Environment [Blomer et al. (2014)] OGSTools provides additionally a pinned environment, updated at least once per release. While reproducibility requires environments with pinned dependencies. OGSTools is additionally tested with the latest dependencies to receive early warning of breaking changes and support long-term sustainability of the codebase.

To support broad adoption within the OGS user community, the library is deliberately integrated at key points of interest, such as the official OGS benchmarks, executable test cases, and further contexts where previously used libraries were employed.

Features

The implemented features are covering pre-processing, setup and execution of simulations and post-processing.

Preprocessing for OGS includes mesh adaptation, conversion, refinement, and creation, as well as defining boundary conditions, source terms, and generating project files (OGS specific XML-Files). Building upon this, a FEFLOW converter (from FEFLOW models to OGS models) is integrated.

Postprocessing includes domain specific evaluation and visualization of simulation results, for temporal and spatial distribution analysis. OGSTools helps to create detailed plots by defining sensible defaults and OGS-specific standards. It offers functionalities for the comparison of numerical simulation results with experimental data or analytical solutions. Just as preprocessing and analysis are essential for single simulations, tooling becomes critical for efficiently handling ensemble runs. Ensemble runs enable evaluation of model robustness, parameter sensitivity,

numerical behavior, and computational efficiency, with spatial and temporal grid conversion currently supported.

A more complete list of examples covering a significant part of the features set is found in the online documentation of OGSTools¹. Containers are provided for reproducibility, benefiting both developers and users ([?]). Like OpenGeoSys, OGSTools is available on PyPI and Conda.

Applications

OGSTools has been integral part of complex workflows. OGSTools is designed to support the implementation of such workflow components where OGS is involved.

AREHS

The AREHS-Project (effects of changing boundary conditions on the development of hydro-geological systems: numerical long-term modelling considering thermal-hydraulic-mechanical(-chemical) coupled effects) project [Kahnt et al. (2021)] focused on modeling the effects of the glacial cycle on hydro-geological parameters in potential geological nuclear waste repositories in Germany. [Zill et al. (2024)] and [Silbermann et al. (2025)] highlighted the importance of an automated workflows to efficiently develop models to answer the scientific question and to ensure the reproducibility of the results. This workflow covers all necessary steps from a structured layered model and geological parameters over the simulation with OGS to the resulting figures shown in [Zill et al. (2024)] and [Silbermann et al. (2025)]. It is composed as Snakemake workflow and all material available on [Meisel et al. (2024)].

OpenWorkflow

OpenWorkFlow [Environmental Research UFZ (2023)], is a project for an open-source, modular synthesis platform designed for safety assessment in the nuclear waste site selection procedure of Germany. Automated workflows as a piece of the planned scientific computational basis for investigating repository-induced physical and chemical processes in different geological setting are essential for transparent and reproducible simulation results. OGS together with OGSTools has been used in a study of thermal repository dimensioning - named ThEDi. TheDi focuses on determining the optimal packing of disposal containers in a repository to ensure temperature limits are not exceeded. The fully automated workflow generates the simulation models based on geometric and material data, runs and analyses the simulations. For scalability and parallelization the workflow is embedded optionally within the workflow management Snakemake. The workflow components are implemented reusing OGSTools functionalities.

OpenGeoSys benchmarks

The OGS benchmark gallery is a collection of web documents (mostly generated from Jupyter Notebooks) that demonstrate, how users can set up, adjust, execute, and analyse simulations. They can be downloaded, executed, and adapted in an interactive environment for further exploration. With OGSTools code complexity and code duplication could be reduced, and it allows especially inexperienced users to focus on the important part of the notebook.

OGS-GIScape

OGS-GIScape is a Snakemake-based workflow for creating, simulating and analysing numerical groundwater models (NGM). OGS-GIScape enables scientists to investigate complex environmental models to study the groundwater flow and the associated environmental impact or conduct scenario analyses. The models could be used to estimate the impact due to changes in groundwater resources. Furthermore, the outcome of the models could be used for the management of groundwater resources.

¹<https://ogstools.opengeosys.org>

134 An important part of the NGM creation is the geometric model (mesh). It is build using
 135 geographic information system (GIS) tools at the landscape scale and combining various
 136 meshing tools. The workflow also comprises the parametrisation of the geometric model with
 137 physical parameters as well as defining boundary conditions, for instance groundwater recharge
 138 on the top of the computational domain or the integration of rivers. For these workflow steps
 139 it is mainly necessary to change parts of the OGS project file which is done with OGSTools.

140 Acknowledgements

141 This work has been supported by multiple funding sources, including AREHS under grant
 142 4719F10402 by Bundesamt für die Sicherheit der nuklearen Entsorgung(BASE), and
 143 OpenWorkflow under grant STAFuE-21-05-Klei by Bundesgesellschaft für Endlagerung
 144 (BGE). The authors also acknowledge ongoing support from SUT0GS (Streamlining Usability and
 145 Testing of OpenGeoSys) under (grant [Grant Number]) by Deutsche Forschungsgemeinschaft
 146 (DFG)

147 Bilke, L., Flemisch, B., Kalbacher, T., Kolditz, O., Helmig, R., & Nagel, T. (2019). Develop-
 148 ment of Open-Source Porous Media Simulators: Principles and Experiences. *Transport in*
 149 *Porous Media*, 130(1), 337–361. <https://doi.org/10.1007/s11242-019-01310-1>

150 Bilke, L., Naumov, D., Wang, W., Fischer, T., Kiszkurko, F. K., Lehmann, C., Max, J.,
 151 Zill, F., Buchwald, J., Grunwald, N., Kessler, K., Aubry, L., Dörnbrack, M., Nagel, T.,
 152 Ahrendt, L., Kaiser, S., & Meisel, T. (2025). *OpenGeoSys* (Version 6.5.4). Zenodo.
 153 <https://doi.org/10.5281/zenodo.14672997>

154 Blomer, J., Berzano, D., Buncic, P., Charalampidis, I., Ganis, G., Lestaris, G., & Meusel, R.
 155 (2014). The need for a versioned data analysis software environment. *CoRR*, *abs/1407.3063*.
 156 <http://arxiv.org/abs/1407.3063>

157 Buchwald, J., Kolditz, O., & Nagel, T. (2021). ogs6py and VTUinterface: Streamlining
 158 OpenGeoSys workflows in python. *Journal of Open Source Software*, 6(67), 3673. <https://doi.org/10.21105/joss.03673>

160 Diercks, P., Gläser, D., Lünsdorf, O., Selzer, M., Flemisch, B., & Unger, J. F. (2022).
 161 *Evaluation of tools for describing, reproducing and reusing scientific workflows*. <https://arxiv.org/abs/2211.06429>

163 Environmental Research UFZ, H. C. for. (2023). *OpenWorkFlow - synthesis platform*.
 164 <https://www.ufz.de/index.php?en=48378>

165 Goble, C., Cohen-Boulakia, S., Soiland-Reyes, S., Garijo, D., Gil, Y., Crusoe, M. R., Peters, K.,
 166 & Schober, D. (2020). FAIR computational workflows. *Data Intelligence*, 2(1-2), 108–131.
 167 https://doi.org/10.1162/dint_a_00033

168 Huber, S. P., Zoupanos, S., Uhrin, M., Talirz, L., Kahle, L., Häuselmann, R., Gresch, D., Müller,
 169 T., Yakutovich, A. V., Andersen, C. W., Ramirez, F. F., Adorf, C. S., Gargiulo, F., Kumbhar,
 170 S., Passaro, E., Johnston, C., Merkys, A., Cepellotti, A., Mounet, N., ... Pizzi, G. (2020).
 171 AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows
 172 and data provenance. *Scientific Data*, 7(1). <https://doi.org/10.1038/s41597-020-00638-4>

173 Kahnt, R., Konietzky, H., Nagel, T., Kolditz, O., Jockel, A., Silbermann, C., Tiedke, F.,
 174 Meisel, T., Rink, K., Wang, W., Zill, F., Carl, A., Gabriel, A., Schlegel, M., & Abraham,
 175 T. (2021). AREHS: Effects of changing boundary conditions on the development of
 176 hydrogeological systems: Numerical long-term modelling considering thermal-hydraulic-me-
 177 chanical(–chemical) coupled effects. *Safety of Nuclear Waste Disposal*, 1, 175–177.
 178 <https://doi.org/10.5194/sand-1-175-2021>

179 Kern, D., & Bilke, L. (2022). msh2vtu. In *GitHub repository*. GitHub. <https://github.com/dominik-kern/msh2vtu>

- 181 Kolditz, O., Bauer, S., Bilke, L., Grunwald, N., Delfs, J.-O., Fischer, T., Görke, U., Kalbacher,
182 T., Kosakowski, G., Mcdermott, C., Park, C.-H., Radu, F., Rink, K., Shao, H., Sun,
183 F., Sun, Y., Singh, A., Taron, J., Walther, M., & Zehner, B. (2012). OpenGeoSys:
184 An open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical
185 (THM/c) processes in porous media. *Environmental Earth Sciences*, 67, 589–599. <https://doi.org/10.1007/s12665-012-1546-x>
186
- 187 Köster, J., & Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine.
188 *Bioinformatics*, 28(19), 2520–2522. <https://doi.org/10.1093/bioinformatics/bts480>
- 189 Meisel, T., Zill, F., Silbermann, C. B., Wang, W., Bilke, L., & Kern, D. (2024). *AREHS -*
190 *OpenGeoSys workflow* (Version 1.0). Zenodo. <https://doi.org/10.5281/zenodo.11367280>
- 191 Silbermann, C., Zill, F., Meisel, T., Kern, D., Kolditz, O., Magri, F., & Nagel, T. (2025).
192 Automated thermo-hydro-mechanical simulations capturing glacial cycle effects on nuclear
193 waste repositories in clay rock. *Geomechanics and Geophysics for Geo-Energy and Geo-*
194 *Resources*, 11. <https://doi.org/10.1007/s40948-025-00960-4>
- 195 Wilkinson, S. R., Aloqalaa, M., Belhajjame, K., Crusoe, M. R., Paula Kinoshita, B. de, Gadelha,
196 L., Garijo, D., Gustafsson, O. J. R., Juty, N., Kanwal, S., Khan, F. Z., Köster, J., Peters-von
197 Gehlen, K., Pouchard, L., Rannow, R. K., Soiland-Reyes, S., Soranzo, N., Sufi, S., Sun, Z.,
198 ... Goble, C. (2025). Applying the FAIR principles to computational workflows. *Scientific*
199 *Data*, 12(1). <https://doi.org/10.1038/s41597-025-04451-9>
- 200 Zill, F., Bilke, L., Meisel, T., Heinze, J., Kizskurno, F. K., Kern, D., Jäschke, M., & Lehmann,
201 C. (2025). *OGSTools* (Version 0.7.0). Zenodo. <https://doi.org/10.5281/zenodo.15804988>
- 202 Zill, F., Silbermann, C., Meisel, T., Magri, F., & Nagel, T. (2024). Far-field modelling of
203 THM processes in rock salt formations. *Open Geomechanics*, 5, 1–16. <https://doi.org/10.5802/ogeo.20>
204