The Journal of Open Source Software

r2ogs6: An R wrapper of the OpenGeoSys 6 Multiphysics Simulator

Ruben Heinrich[*1], **Johannes Boog**[†2], **Philipp Schad**[‡2], **and Thomas Kalbacher**[§2]

1 Leipzig University of Applied Sciences, Karl-Liebknecht-Strasse 132, 04277 Leipzig, Germany **2** Helmholtz Centre for Environmental Research, Department Environmental Informatics, Permoser Str. 15, 04318 Leipzig, Germany

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Jemma Stachelek

Reviewers:

- @jmp75
- @waternumbers

Submitted: 21 December 2022
Published: unpublished

Summary

Understanding the impacts of climate change and hydrologic extreme events on our sub–surface earth system is even in temperate zones of utmost importance. Key tools to develop such understanding are physics–based simulation models that describe the manifold interactions of involved natural phenomena across time and space.

Our R package r2ogs6 provides a file–based R interface to the multi–physics simulation code OpenGeoSys 6 (Bilke et al., 2019; Kolditz et al., 2012) and, therefore, enables R users to perform and analyze simulation models of the sub–surface earth system in R. r2ogs6 allows to access the capabilities of OpenGeoSys 6 to simulate thermo-hydro-mechanical-chemical and biological (THMC/B) processes in porous and fractured media within R. In this way, r2ogs6 enables R users to model sub–surface phenomena and technologies such as groundwater flow, reactive transport, geothermal energy usage and/or nuclear waste repositories as well as to analyze and further process simulation output. r2ogs6 enables users to prepare and manipulate OpenGeoSys 6 simulation models, run the simulations and retrieve corresponding output, all within an R session or simple R scripts. Therefore, R classes and functions were designed to communicate with the respective OpenGeoSys 6 input and output files as well as executables. In addition to single-simulation runs, r2ogs6 supports ensemble runs that can be used to set up uncertainty and sensitivity analyses as well as parameter studies. Furthermore, r2ogs6 allows conducting and documenting OpenGeoSys 6 simulations in reproducible R scripts or notebooks. As OpenGeoSys 6 is continuously being developed further, code generation functions for r2ogs6 developers were included to speed up the package updating process in case of future changes to OpenGeoSys 6.

r2ogs6 was designed to be used by domain researchers, data scientists and students working with OpenGeoSys 6. Moreover, r2ogs6 was intended to include OpenGeoSys 6 into R based scientific workflows and aims to bridge the gap between data produced by a scientific simulation code and data science.

Statement of need

Major challenges humanity has to face in the coming decades are climate change and hydrologic extremes. Understanding the impacts of climate change and hydrologic extreme events on our sub–surface earth system is even in temperate zones of utmost importance for ensuring

[*] co-first author
[†] co-first author
[‡] co-first author
[§] co-first author

adequate domestic and drinking water supplies, together with functioning lake and river systems with healthy aquatic ecosystems and ecosystem services. Of course, the needs of the population and the needs of nature are often in conflict, which increases the necessity to study the complex interaction of both within different scenarios. The core of such studies is most often the system and scenario analysis based on physics simulations of individual or coupled earth systems compartments. The multiple coupled natural processes implemented in physics simulation models are usually described with partial differential equations. Solving these equations requires appropriate numerical methods such as the finite element method (FEM). For reasons of performance, (multi) physics simulators are mostly implemented in languages such as `FORTRAN`, `C` or `C++`.

One of these tools is `OpenGeoSys` (OGS) (https://www.opengeosys.org/), a scientific open source project for the development of numerical methods to simulate thermo-hydro-mechanical-chemical and biological (THMC/B) processes in porous and fractured media (Bilke et al., 2019; Kolditz et al., 2012). OGS has applications ranging from small-scale geotechnical investigations, to reservoir studies and even groundwater management of entire landscapes.
Since the model setups get more and more complex and larger, faster and more efficient setup and parametrization procedures are needed to create meaningful ensembles that allow to identify the right action needs and to derive further decision support for the above described problems. But while `OpenGeoSys` is a powerful FEM code, setting up, running and evaluating multiple simulations can prove complicated.

Here is where languages such as `R` and `Python` can prove useful. Via an interface that adds a layer on top of `OpenGeoSys 6`, the user can access preprocessing tools, the solver itself and postprocessing tools alike, thus increasing usability and accessibility. The developement and application of user interfaces from geoscientific simulators to high–level programming languages has been gaining increasing attention in recent years; examples are `FloPy` (Bakker et al., 2016), `ogs5py` (Müller et al., 2021), `RedModRPhree`(De Lucia & Kühn, 2021), `r2ogs5` (Schad et al., 2021) and `toughio` (Luu, 2020). For `OpenGeoSys 6`, a `Python` interface is currently under development (Buchwald et al., 2021). Nonetheless, we consider an `R` interface to be just as important, as `R` is a well known language in the environmental and geosciences. Furthermore, since `R` is a popular language in the field of data science with many powerful packages for data analysis and visualisation, e. g. `dplyr` (Wickham, François, et al., 2021) and `ggplot2` (Wickham, Chang, et al., 2021), it's a natural choice for processing data generated by simulation tools such as `OpenGeoSys 6`. Especially, `r2ogs6` can facilitate the calibration of `OpenGeoSys 6` models due to the implemented functions to design ensemble runs as well the available `R` functions and packages for modeling such as `lhs` (Carnell, 2021), `mlrBO` (Bischl et al., 2017). For `R` users who do not have a lot of (or any) experience with, yet an interest in environmental and geoscientific sub–surface simulations, `r2ogs6` provides a good starting point. Utilizing `r2ogs6`, users can easily set up their first `OpenGeoSys 6` simulations by choosing one of the provided benchmark files. Moreover, with `R` scripts and `R--Markdown` or `JupyteR` notebooks, modeling workflows can easily be documented, published and shared with peers.

`r2ogs6` has already been applied in (Heinrich, 2021), where its ensemble functionality was tested utilizing the `OpenGeoSys 6` Theis' problem benchmark files (Walther, 2020; Wang, 2020); or to calibrate groundwater flow models (Boog et al., 2021).

Package Structure

`r2ogs6` is thought to set up an `OpenGeoSys 6` simulation inside an `R` session by executing specified model creation functions or by reading existing `OpenGeoSys 6` input files. With further functions, the simulation can be executed and corresponding output can be read into the `R` session again. Figure 1 highlights the structure of `r2ogs6`. The central element that represents an `OpenGeoSys 6` simulation is the `OGS6` object, which is an instance of a `R6` class. This object represents a single simulation; multiple simulations can be defined with the `OGS6_Ensemble` class. An `OGS6` object contains several child objects that represent the simulation input and output. The main OpenGeoSys 6 input files are the project file *.prj, geometry file *.gml and input FEM mesh file(s) *.vtu. These are read in or written via S3 class based functions (block read_in* / export* in Figure 1). When reading in, the XML–based *.prj input file is parsed. Individual tags are represented as S3 class objects which are available via active fields in the OGS6 object. Individual *.prj tags may change due to ongoing development activities in OpenGeoSys 6, therefore, future updates of the related classes may be necessary. To simplify updates like this, helper functions for analyzing *.prj files as well as suggesting and creating classes were implemented.

As the *.gml and the *.vtu files are less complex and less likely to change, these files are represented as R6 class objects and also available as active fields inside the OGS6 object. To execute simulations, functions for writing the OpenGeoSys 6 input (ogs6_export_simfiles()) and call the OpenGeoSys 6 executable (ogs6_run_simulation()) were implemented. Note that an OpenGeoSys 6 executable or singularity container needs to be present. Default executables and paths can be defined in a configuration file.

During execution OpenGeoSys 6 generates output data as *.vtu files. These files are produced at user defined timesteps of the simulation and are referenced in a *.pvd file. The function ogs6_read_output_files() then attaches the output files to the OGS6 object as OGS6_pvd objects (which in turn reference OGS6_vtu objects). In this way, all data required for and produced by OpenGeoSys 6 can be represented as R native objects and results can be processed further using R functions.

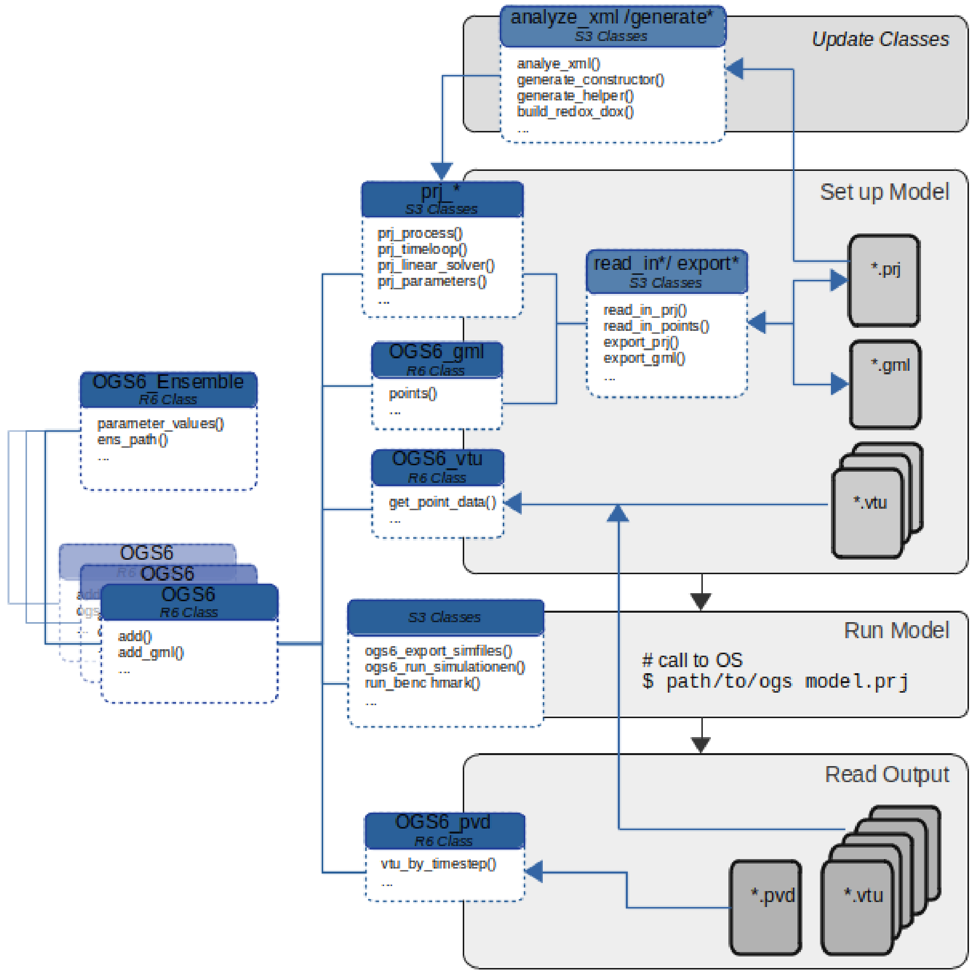

Figure 1: Schematic of the r2ogs6 structure.

The package comes with tutorials demonstrating how to set up and run a single simulation (link), set up simulation ensembles (link) and further develop the package (link). Moreover, r2ogs6 includes functions to create R scripts from existing OpenGeoSys 6 benchmarks that will allow a quick start for new users.

Acknowledgements

We acknowledge funding from the Initiative and Networking Fund of the Helmholtz Association through the project *Digital Earth* (funding code ZT-0025). Furthermore, we acknowledge the Helmholtz Centre for Environmental Research–UFZ for additional funding and support. We would like to express our gratitude to the *OpenGeoSys Community* for technical support and for hosting the GitLab server (https://gitlab.opengeosys.org) for our development. The package has in part been developed on the High-Performance Computing (HPC) Cluster EVE, a joint effort of both the Helmholtz Centre for Environmental Research - UFZ (http://www.ufz.de/) and the German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig (http://www.idiv-biodiversity.de/). We would like to thank the administration and support staff of EVE who keep the system running and support us with our scientific computing needs: Thomas Schnicke, Ben Langenberg, Guido Schramm, Toni Harzendorf and Tom Strempel from the UFZ, and Christian Krause from iDiv.

The Journal of Open Source Software

References

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J., & Fienen, M. N. (2016). Scripting MODFLOW model development using python and FloPy. *Groundwater*, *54*(5), 733–739. https://doi.org/https://doi.org/10.1111/gwat.12413

Bilke, L., Flemisch, B., Kalbacher, T., Kolditz, O., Helmig, R., & Nagel, T. (2019). Development of open-source porous media simulators: Principles and experiences. *Transport in Porous Media*, *130*, 337–361. https://doi.org/10.1007/s11242-019-01310-1

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J., & Lang, M. (2017). *mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions*. https://arxiv.org/abs/1703.03373

Boog, J., Sips, M., De Lucia, M., Eggert, D., & Kalbacher, T. (2021). DATA-driven CALibration of Large-Scale Physical Groundwater Flow Models using Meta-Modeling and Visual Analytics. *AGU Fall Meeting Abstracts*, *2021*, H34D–04.

Buchwald, J., Fischer, T., Naumov, D. Y., & Kolditz, O. (2021). *ogs6py: Python-API for the OpenGeoSys software*. https://doi.org/10.5281/zenodo.5387765

Carnell, R. (2021). *Lhs: Latin hypercube samples*. https://bertcarnell.github.io/lhs/index.html

De Lucia, M., & Kühn, M. (2021). Geochemical and reactive transport modelling in r with the RedModRphree package. *Advances in Geoscience*, *56*, 33–43. https://doi.org/10.5194/adgeo-56-33-2021

Heinrich, R. (2021). *Entwicklung und implementierung der schnittstelle R2OpenGeoSys zur workflow-optimierung am beispiel der modellierung eines stofftransportproblems* [Master's thesis]. HTWK Leipzig.

Kolditz, O., Görke, U.-J., Shao, H., & Wang, W. (Eds.). (2012). *Thermo-hydro-mechanical-chemical processes in porous media: Benchmarks and examples* (1. ed.). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-27177-9

Luu, K. (2020). Toughio: Pre- and post-processing python library for TOUGH. *Journal of Open Source Software*, *5*(51), 2412. https://doi.org/10.21105/joss.02412

Müller, S., Zech, A., & Heße, F. (2021). ogs5py: A python-API for the OpenGeoSys 5 scientific modeling package. *Groundwater*, *59*(1), 117–122. https://doi.org/https://doi.org/10.1111/gwat.13017

Schad, P., Boog, J., & Kalbacher, T. (2021). r2ogs5: Controlling and calibrating OpenGeoSys 5 simulation within r. *Submitted to Groundwater*.

Walther, M. (2020). *Theis solution for well pumping*. https://www.opengeosys.org/docs/benchmarks/hydro-component/theis/hc_theis/

Wang, W. (2020). *Theis' problem*. https://www.opengeosys.org/docs/benchmarks/liquid-flow/liquid-flow-theis-problem/

Wickham, H., Chang, W., Henry, L., Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H., & Dunnington, D. (2021). *ggplot2: Create elegant data visualisations using the grammar of graphics*. https://CRAN.R-project.org/package=ggplot2

Wickham, H., François, R., Henry, L., & Müller, K. (2021). *Dplyr: A grammar of data manipulation*. https://CRAN.R-project.org/package=dplyr