The Journal of Open Source Software

OpenTerrace: A fast, flexible and extendable Python framework for thermal storage packed bed simulations

Jakob Hærvig [1*]

1 Aalborg University, Denmark ***** These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 08 December 2023
Published: unpublished

Summary

Being able to store energy for extended periods of time is important for modern societies where increasing amounts of energy comes from renewable sources with time-varying production. Many storage technologies exist, each with their own set of disadvantages and advantages. Storing energy in the form of thermal energy is a promising, cheap and energy efficient solution, which can easily be scaled up.

A fluid is heated up and pumped through a storage tank. The storage tank contains either just a fluid or is filled with a bedding material (known as a packed bed). The bedding material could range from regular, readily available stones to more exotic capsules containing phase-change material encapsulated in a plastic shell. In either case the hot fluid transfers energy to the bed material, which then stores the energy in form of sensible and/or latent heat. The process is simply reversed to extract energy from the bed material and transfer it to the fluid.

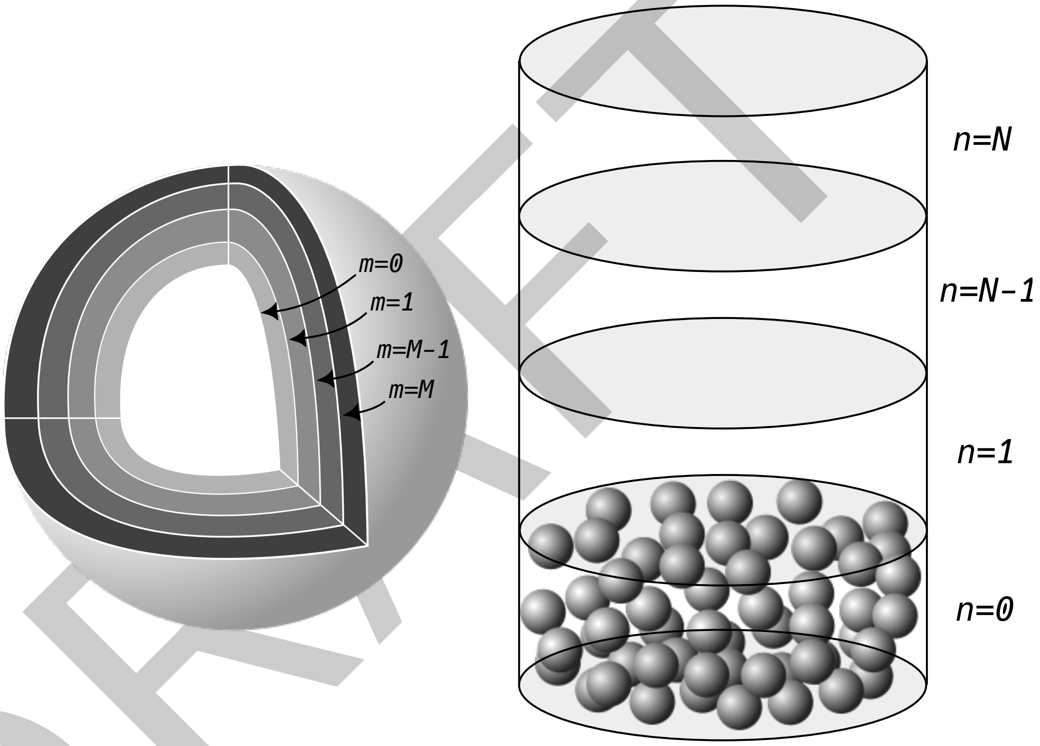
Open
Terrace
A fast, flexible and extendable Python framework
for thermal storage packed bed simulations

Statement of need

`OpenTerrace` is an open-source simulation framework providing easy access to complex simulations. The core part of the computations are made by calling low-level languages using packages such as NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020) and Numba (Lam et al., 2015) for speed while maintaining the flexibility and user-friendliness of Python. Plotting and animation capabilities based on Matplotlib (Hunter, 2007) are provided with the framework to allow easy access to visualisations.

`OpenTerrace` solves for a fluid phase, which is descretised in `N` nodes. For each node, the bed material is discretised in `M` nodes. The framework allows arbitrary number of phases to be defined and coupled in different ways.

Each phase is described by a non-linear, partial differential equation. The phases are coupled using source terms to account for various physical mechanisms. The framework is flexible in such a way that new types of source terms, new domain shape functions, new fluid and bed substances (including phase-changing substances) with both temperature dependent- and independent properties and new numerical discretisation schemes can easily be added. A set of tutorials is also provided with the framework to highlight its functionality in its current version and to ease the learning curve for new users. Also various unit tests are provided to verify different parts of the code in some well-defined benchmark cases.



`OpenTerrace` serves different purposes. First, it was designed to be used by both researchers doing research projects and industry making design decisions. Also, it has now proven useful for engineering students working on semester projects where the learning outcome includes both thermal energy storage knowledge and coding experience with Python.

Acknowledgements

This research has received funding from EU Horizon project SERENE under grant agreement no. 130930.

References

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

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

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. https://doi.org/10.1145/2833157.2833162

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2