




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

Jakob Hærvig  ^{1*}

¹ Aalborg University, Denmark * These authors contributed equally.

DOI: [10.21105/joss.06411](https://doi.org/10.21105/joss.06411)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

OpenTerrace

Editor: [Adam R. Jensen](#)  

Reviewers:

- [@baperry2](#)
- [@AdamRJensen](#)

Submitted: 08 December 2023

Published: 03 June 2024

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

OpenTerrace is a simulation framework for the prediction of transient temperature responses in thermal energy storage systems. Being able to store energy for extended periods of time is important for modern societies where increasing amounts of energy stem 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 solution because it is cheap and can be scaled up easily. While the total energy content of thermal energy storage systems can be estimated easily, the transient response requires solving partial differential equations in space and time. Using OpenTerrace the transient response of a wide range of thermal energy storage systems can be simulated easily. The storage system contains a storage tank filled with a carrier fluid and an optional bed phase. OpenTerrace comes with a wide range of predefined substances to be used as either fluid or bed material. Also, OpenTerrace ships with a set of primitive, predefined shapes that act as either storage tank or bed material. Besides that a set of predefined boundary conditions and source terms cover many different thermal energy storage applications.

OpenTerrace is built to be:

- Fast by making use of modern compilers and optimised tri-diagonal matrix solvers.
- Flexible for easy integration in system models and optimisation loops.
- Extendable by allowing new modules for new materials such as non-spherical rocks or exotic Phase Change Materials (PCM) to easily be plugged into the OpenTerrace framework.

More information about how to get started, along with a user guide, can be found in the [OpenTerrace documentation](#). Users may send [pull requests](#) to have their contributions with new functionality added to the official [OpenTerrace GitHub repository](#). A set of tutorials is also provided within the framework to highlight its current functionality 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.

Statement of need

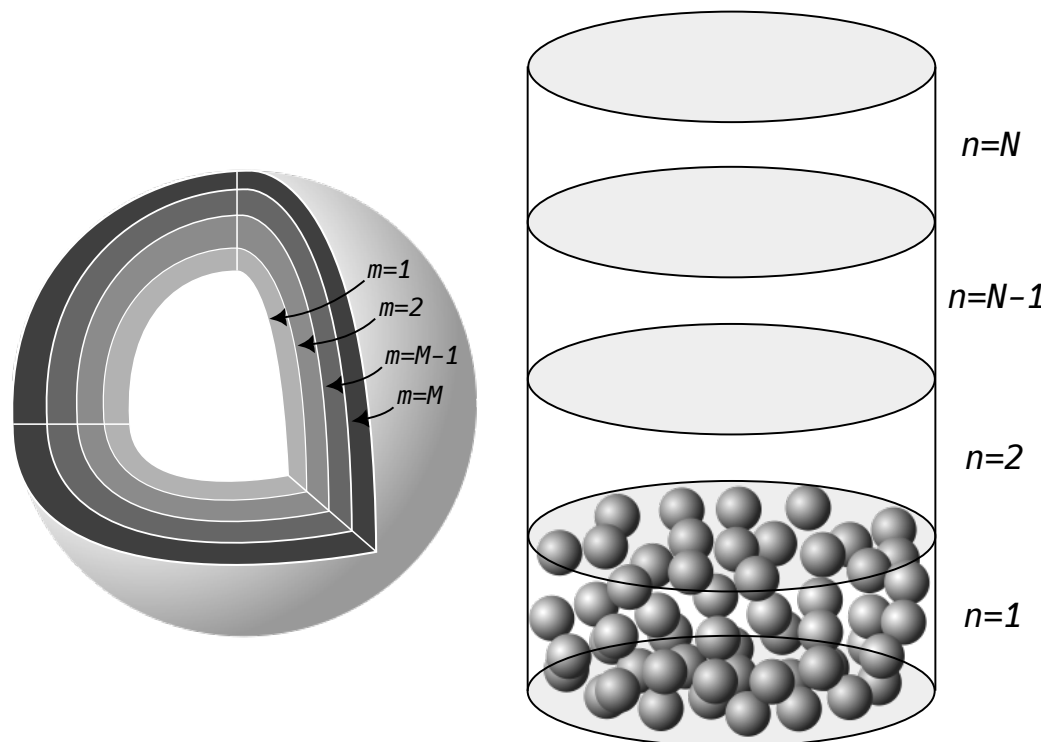
Research on thermal energy storage systems commonly relies on in-house developed code, which makes the results presented in research papers less transparent and not reproducible. Various models of varying complexity have been proposed in the literature. For a packed bed storage system, Schumann (Schumann, 1929) proposed a model that neglects heat conduction in the fluid, between the bed material, and inside the bed material. Later research by (Littman et al., 1968) included conduction in the fluid and between bed particles, which was shown to be important for low Reynolds number flows. Depending on the Biot number for the bed material, temperature gradients inside the particles may also be important, as shown by (Handley & Heggs, 1969). This is especially important when the bed material consists of Phase Change Material (PCM).

OpenTerrace is an open-source simulation framework, which aims to provide researchers and decision makers in industry with a common, transparent, open-source framework for simulating thermal energy storage packed bed systems. It includes the possibility to easily include or exclude the above-mentioned effects. It assumes 1D heat transfer in both the fluid and bed phases. Also, 1D heat transfer is assumed inside the bed material.

The core part of the computations relies on calls to 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 rely on Matplotlib (Hunter, 2007) and are provided with the framework to allow easy access to visualisations.

Features and functionality

OpenTerrace solves for a fluid phase, which is discretised in N nodes. Within each fluid volume element, the bed material is assumed to have the same temperature distribution. The bed material is discretised in M nodes, which results in a total of $N \times M$ equations to be solved at each time step, see the figure below.



Right side figure shows a cylindrical storage tank with 4 nodes. The bed material in each volume element is assumed to have similar temperature distribution. Left side figure shows discretisation of the bed material, which in this case is assumed to be a hollow sphere.

Each phase is described by a non-linear, partial differential equation. The phases are coupled using source terms to account for various physical mechanisms such as heat loss and convective heat transfer between the phases. 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 be added easily.

Target Audience

OpenTerrace serves different purposes. First and foremost it is a research tool to be used by researchers and industry for design decisions. Also, it has now proven useful in an educational context in which engineering students become familiar with thermal storage systems, test out new ideas or extend the functionality of OpenTerrace.

Acknowledgements

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

References

- Handley, D., & Heggs, P. J. (1969). The effect of thermal conductivity of the packing material on transient heat transfer in a fixed bed. *International Journal of Heat and Mass Transfer*, 12(5), 549–570. [https://doi.org/10.1016/0017-9310\(69\)90038-6](https://doi.org/10.1016/0017-9310(69)90038-6)
- 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>
- Littman, H., Barile, R. G., & Pulsifer, A. H. (1968). Gas-Particle Heat Transfer Coefficients in Packed Beds at Low Reynolds Numbers. *Ind. Eng. Chem. Fundamen.*, 7(4). <https://doi.org/10.1021/i160028a005>
- Schumann, T. E. W. (1929). Heat transfer: A liquid flowing through a porous prism. *Journal of the Franklin Institute*, 208(3), 405–416. [https://doi.org/10.1016/S0016-0032\(29\)91186-8](https://doi.org/10.1016/S0016-0032(29)91186-8)
- 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>