The Journal of Open Source Software

Snek5000: a new Python framework for Nek5000

Ashwin Vishnu Mohanan [1*], **Arman Khoubani** [2*], **and Pierre Augier** [2*]

1 Swedish Meteorological and Hydrological Institute, Norrköping, Sweden **2** Laboratoire des Écoulements Géophysiques et Industriels, Université Grenoble Alpes, CNRS, Grenoble INP, 38000 Grenoble, France ***** These authors contributed equally.

DOI: 10.21105/joss.05586

Software

- Review
- Repository
- Archive

Editor: Philip Cardiff

Reviewers:

- @joneuhauser
- @maxhutch

Submitted: 28 April 2023
Published: 24 August 2023

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

Summary

Computational fluid dynamics (CFD) simulations are essential tools in various scientific and engineering disciplines. Nek5000 is a CFD Fortran code based on spectral element methods with a proven track record in numerous applications. In this article, we present Snek5000, a Python package designed to streamline the management and visualization of fluid dynamics simulations based on Nek5000. The package builds upon the functionality of Nek5000 by providing a user-friendly interface for launching and restarting simulations, loading simulation data, and generating figures and movies. This paper introduces Snek5000, discusses its design principles, and highlights its impact on the scientific community.

Statement of need

State of the art

The CFD framework Nek5000 (*NEK5000 Version 19.0*, 2019) is the culmination of several decades of development. Nek5000 solvers can produce high-fidelity simulations owing to the spectral-element method and can scale up to several thousands of cores (Offermans et al., 2016). Development of Nek5000 is primarily driven by performance optimization, incorporating new numerical methods whilst following a keep-it-simple approach to ensure portability across various high-performance computing machines.

Development of Nek5000 continues to this day with efforts underway to use GPUs (Vincent et al., 2022) and to rewrite it in C++ (Fischer et al., 2022) and modern Fortran (Jansson et al., 2021).

To the best of the authors' knowledge no other actively maintained and reusable approaches have been made to wrap Nek5000. A project called nekpy was the only known prior work resembling Snek5000, where it uses template source files to fill in parameters.

Better user-experience with Snek5000

Snek5000 enhances the user-experience by addressing the following downsides of using a typical Nek5000 solver:

1. Only a limited set of utilities come packaged with Nek5000 and those focus on compilation and mesh-generation. As a result, usability of Nek5000 takes a hit and a practitioner is left to construct a home-brewed solution to conduct exploratory research and parametric studies. Snek5000 functions as a workflow manager for assisting packaging, setup, compilation and post-processing aspects of a simulation.
2. The simulation parameters are spread across at least three different files (`*.box`, `*.par`, and `SIZE`). Some parameters have short and cryptic names (for example, `lx1`, `lxd`) and

are dependent on each other. Snek5000 tries to provide good defaults, uses more legible parameter names when necessary and dynamically set some of these parameters when possible. This allows a user to get started without the need to master the whole manual.

Snek5000: design principles, features and capabilities

Powered by Python Packages

Snek5000 leverages a variety of Python packages, including Snakemake (Mölder et al., 2021), FluidSim (Mohanan et al., 2019), Pymech (Mohanan et al., 2022), Matplotlib (Hunter, 2007), Jinja, Pytest, and Xarray (Hoyer & Hamman, 2017), to deliver a robust and user-friendly workflow management tool for Nek5000. These packages provide a powerful foundation for Snek5000, enabling its seamless integration with existing Python-based workflows and enhancing its overall usability.

A FluidSim extension

Snek5000 is based on the CFD framework FluidSim (Mohanan et al., 2019), which introduces the concept of "FluidSim solvers". A FluidSim solver consists of few files describing a set of potential and similar simulations. A concrete simulation can be created via a simple and generic Python API. For example, for the `snek5000-cbox` solver,

```python
from snek5000_cbox import Simul

params = Simul.create_default_params()

# set simulation parameters
...

sim = Simul(params)
```

During the instantiation of the `Simul` object, all the directories and files necessary to run the simulation have been created. We see that Snek5000 can be seen as an advanced template system. Then, one can launch the Nek5000 simulation with `sim.make.exec("run_fg")`. Further details of these two stages can be found in the Jinja templates and Snakemake rules, which are responsible for source-code generation and execution of Nek5000 commands respectively. These files are organized under the sub-package `snek5000.resources` and are both re-usable and extendable.

Since the simulations generated by a solver share some similarities (for example, some aspects of the geometry and the equations), the solver can contain code to create, plot and post-process output data, which is accessible through objects contained in `sim.output`.

Streamlined simulation management

With a Snek5000-FluidSim solver, users can efficiently launch and restart simulations using Python scripts and a terminal command (`snek-restart`). Snek5000 handles all file operations, such as directory creation and file copying. This streamlines the process of managing simulations, freeing up time and resources for data analysis and understanding the underlying physics.

Loading simulations for data visualization, post-processing and data analysis

It is very easy to "load" an existing simulation, i.e., to recreate a Python object `sim` similar to the one used to create the simulation. This can be done with the function `snek5000.load` or with the command `snek-ipy-load`, which opens a IPython session with a `sim` variable. Snek5000 simplifies the process of reading associated parameters (in `sim.params`) and data, and generating visualizations, such as figures (sim.output.phys_fields.plot_hexa) and movies (sim.output.phys_fields.animate). By utilizing popular Python packages, such as Matplotlib (Hunter, 2007) and Xarray (Hoyer & Hamman, 2017), Snek5000 facilitates the creation of high-quality visualizations that can be easily customized to meet individual needs. This powerful visualization capability aids researchers in understanding complex fluid dynamics phenomena and effectively presenting their findings. Beyond visualization, Snek5000 also provides tools for post-processing and data analysis. Users can easily load simulation data into Python for further processing, statistical analysis, and comparison between different simulations. This streamlined approach to data analysis enables researchers to gain valuable insights into their simulations and focus on the underlying physical processes.

Tutorials and documentation

Snek5000 provides comprehensive documentation and tutorials to guide users through its features and capabilities. These resources help new users quickly become familiar with Snek5000 and enable experienced users to explore advanced features and customization options. By providing thorough documentation, Snek5000 promotes its widespread adoption and fosters a community of users and developers.

Open-source solvers, such as snek5000-phill, snek5000-cbox, and snek5000-tgv, are available, and users can easily develop custom solvers tailored to their specific Nek5000 cases. This flexibility allows researchers to adapt Snek5000 to a wide range of fluid dynamics problems and simulation requirements. For example, Snek5000-cbox has been used for a study on linear stability of vertical convection (Khoubani et al., 2023).

Future developments and enhancements

The Snek5000 development team is committed to continuously improving the package and incorporating user feedback to address evolving needs within the scientific community.

Snek5000 has been thoughtfully designed with modularity and code reuse principles in mind. By leveraging inheritance and object-oriented programming, Snek5000 is well-positioned to accommodate the adoption of the next-generation NekRS (Fischer et al., 2022) code, developed by the Nek5000 team, while maintaining its existing structure and functionality. This adaptability ensures that the framework stays up-to-date with the latest advancements in the field. In the future, Snek5000 can also function as a compatibility layer to migrate to upcoming rewrites of Nek5000 which require some extra input files (Fischer et al., 2022; Jansson et al., 2021).

The design principles of Snek5000 has inspired FluidsimFoam, currently under development, a promising FluidSim extension to bridge the gap between FluidSim and OpenFOAM (Weller et al., 1998). This extension allows users to create custom FluidSim solvers specifically tailored for simulations on the widely-used open-source CFD software package, OpenFOAM. By harnessing the strengths of Python and OpenFOAM, FluidsimFoam aims to provide a versatile and user-friendly environment for conducting computational fluid dynamics simulations, broadening the scope of potential applications.

Acknowledgements

The financial support of the SeRC ExABL project which made this project possible is duly acknowledged. The authors also acknowledge the numerical support provided by Olivier De-Marchi, Gabriel Moreau and Cyrille Bonamy of the LEGI informatics team. This project was funded by the project LEFE/IMAGO-2019 contract COSTRIO. AK acknowledges the finance of his PhD thesis from the school STEP of the University Grenoble Alpes. Part of this work was performed using resources provided under GENCI allocation number A0120107567.
A CC-BY public copyright license has been applied by the authors to the present document

The Journal of Open Source Software

and will be applied to all subsequent versions up to the Author Accepted Manuscript arising from this submission, in accordance with the grant's open access conditions.

References

Fischer, P., Kerkemeier, S., Min, M., Lan, Y.-H., Phillips, M., Rathnayake, T., Merzari, E., Tomboulides, A., Karakus, A., Chalmers, N., & others. (2022). NekRS, a GPU-accelerated spectral element Navier–Stokes solver. *Parallel Computing*, *114*, 102982. https://doi.org/10.1016/j.parco.2022.102982

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

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

Jansson, N., Karp, M., Podobas, A., Markidis, S., & Schlatter, P. (2021). *Neko: A modern, portable, and scalable framework for high-fidelity computational fluid dynamics*. https://arxiv.org/abs/2107.01243

Khoubani, A., Mohanan, A. V., Augier, P., & Flór, J.-B. (2023). Vertical convection regimes in a rectangular cavity: Prandtl and aspect ratio dependance. *Submitted to the Journal of Fluid Mechanics*. https://arxiv.org/abs/2304.12657

Mohanan, A. V., Bonamy, C., Linares, M. C., & Augier, P. (2019). FluidSim: Modular, object-oriented Python package for high-performance CFD simulations. *Journal of Open Research Software*, *7*. https://doi.org/10.5334/jors.239

Mohanan, A. V., Chauvat, G., Kleine, V. G., Fabbiane, N., & Canton, J. (2022). *Pymech: A Python software suite for Nek5000 and SIMSON*. https://doi.org/10.5281/zenodo.7358961

Mölder, F., Jablonski, K., Letcher, B., Hall, M., Tomkins-Tinch, C., Sochat, V., Forster, J., Lee, S., Twardziok, S., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., & Köster, J. (2021). Sustainable data analysis with Snakemake [version 1; peer review: 1 approved, 1 approved with reservations]. *F1000Research*, *10*(33). https://doi.org/10.12688/f1000research.29032.1

NEK5000 version 19.0. (2019). Argonne National Laboratory, Illinois. https://nek5000.mcs.anl.gov/

Offermans, N., Marin, O., Schanen, M., Gong, J., Fischer, P., Schlatter, P., Obabko, A., Peplinski, A., Hutchinson, M., & Merzari, E. (2016). On the strong scaling of the spectral element solver Nek5000 on petascale systems. *Proceedings of the Exascale Applications and Software Conference 2016*. https://doi.org/10.1145/2938615.2938617

Vincent, J., Gong, J., Karp, M., Peplinski, A., Jansson, N., Podobas, A., Jocksch, A., Yao, J., Hussain, F., Markidis, S., Karlsson, M., Pleiter, D., Laure, E., & Schlatter, P. (2022). Strong scaling of OpenACC enabled Nek5000 on several GPU based HPC systems. *International Conference on High Performance Computing in Asia-Pacific Region*, 94–102. https://doi.org/10.1145/3492805.3492818

Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, *12*(6), 620–631. https://doi.org/10.1063/1.168744