FOXES: Farm Optimization and eXtended yield Evaluation Software

Jonas Schmidt [1][¶], **Lukas Vollmer**[1], **Martin Dörenkämper** [1], **and Bernhard Stoevesandt** [1]

1 Fraunhofer IWES, Küpkersweg 70, 26129 Oldenburg, Germany ¶ Corresponding author

Summary

The `Farm Evaluation and eXtended yield Evaluation Software` (`foxes`) is a Python package that calculates wind farm results such as power production or wind turbine wake effects. The meteorological input data can be a timeseries or statistical data like wind speed histograms, wind roses or any other distribution. Uniform inflow fields are supported as well as horizontal profiles and three dimensional flow fields from other sources, for example wind fields from mesoscale weather models or wind atlases (Dörenkämper et al., 2020). Typical applications are wind farm optimization, e.g. layout optimization or wake steering; wind farm pre and post construction analyses; wake model studies, comparison and validation; and wind farm simulations invoking complex model interactions.

Statement of need

The amount of electrical energy that wind farms are able to extract from the kinetic energy of the wind naturally depends on the meteorological conditions at the site. In particular, wind speed and wind direction are decisive quantities, and in general those are time and space dependent fields. Furthermore, for each wind turbine, its location within the wind farm plays an important role, since wake effects stemming from up-stream turbines reduce the wind speed and increase turbulence at the rotor. Additionally, the rotor height, its size as well as specifics related to the wind turbine model are crucial parameters. When it comes to wind farm production analysis, the fast evaluation of the annual energy production for long-term time series of meteorological conditions is required, and such data often contains many ten thousand entries.

In addition, wake effects from neighbouring wind farms and wind farm clusters need to be considered, such that realistic offshore scenarios can include as many as thousands of wind turbines. Hence scalability is an issue of central importance for modern numerical wind farm and wake modelling approaches.

`foxes` is built upon the idea of fast vectorized evaluation of the input states, making use of the dask package (Dask Development Team, 2016) via the `xarray` package (Hoyer & Hamman, 2017). This means that in general, it does not solve differential equations that interconnect the states, but rather relies either on analytical models or on lookup-table based models. Exceptions are the included features of calculations along streamlines of the background flow and integrations of variables along the latter for some models.

`foxes` was designed to be very modular when it comes to modelling aspects of wind farm calculations. Users can combine various model types and easily add new models, such that a realistic representation of the wind farm's behaviour can be achieved. Such models include, among others, rotor discretization models, wake models and turbine models such as wind

sector management, turbine derating or power boost.

Any wind farm calculation variable in `foxes` can be subjected to optimization via the `iwopy` package (Schmidt, 2023), with support for vectorized evaluation of optimization variables. This is crucial for the fast evaluation of genetic or other heuristic algorithms, for example for the purpose of solving the turbine positioning optimization problem or finding optimal wind farm control parameters.

Other related open source Python packages that follow a similar agenda, but with differences in methods, models and architecture, are pywake (Mads M. Pedersen & Réthoré, 2023) and `floris` (Flemming, 2023). The foxes package has no code overlap with these packages and has been developed as an independent software for many years.

Acknowledgements

References

Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. https://dask.org

Dörenkämper, M., Olsen, B. T., Witha, B., Hahmann, A. N., Davis, N. N., Barcons, J., Ezber, Y., García-Bustamante, E., González-Rouco, J. F., Navarro, J., Sastre-Marugán, M., Sīle, T., Trei, W., Žagar, M., Badger, J., Gottschall, J., Sanz Rodrigo, J., & Mann, J. (2020). The making of the new european wind atlas – part 2: Production and evaluation. *Geosci. Model Dev.*, *13*, 5079--5102. https://doi.org/10.5194/gmd-13-5079-2020

Flemming, P. (2023). *FLORIS Wake Modeling and Wind Farm Controls Software*. https://github.com/nrel/floris

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

Mads M. Pedersen, P. van der L., Alexander Meyer Forsting, & Réthoré, P.-E. (2023). *PyWake 2.5.0: An open-source wind farm simulation tool*. https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake

Schmidt, J. (2023). *IWES Optimization Tools in Python*. https://github.com/FraunhoferIWES/iwopy