The Journal of Open Source Software

piglot: an Open-source Package for Derivative-free Optimisation of Numerical Responses

R. P. Cardoso Coelho [1,2¶], **A. Francisca Carvalho Alves** [1,2], **T. M. Nogueira Pires** [1], **and F. M. Andrade Pires** [1,2]

1 Faculty of Engineering, University of Porto, Porto, Portugal **2** Institute of Science and Innovation in Mechanical and Industrial Engineering, Porto, Portugal ¶ Corresponding author

DOI: 10.21105/joss.06652

Software

- Review
- Repository
- Archive

Editor: Hugo Ledoux

Reviewers:

- @Extraweich
- @pprachas

Submitted: 22 February 2024
Published: 23 July 2024

Summary

piglot is an open-source Python tool tailored for the automated optimisation of responses stemming from numerical solvers. This tool aims to provide a simple and user-friendly interface that seamlessly integrates with a spectrum of community solvers, ensuring effortless extension. piglot emerges as a versatile toolkit for tackling inverse problems spanning diverse research domains, such as structural analysis, material modelling, fluid dynamics, control systems, and astrophysics - fields where methodologies such as finite element analysis, spectral methods, and Monte Carlo simulations are often adopted. The primary emphasis is on derivative-free optimisation, ensuring compatibility with black-box solvers in scenarios lacking gradient information and when function evaluations may be noisy.

Statement of need

The increasingly growing interest in computational analysis for engineering problems has driven the development of more accurate, robust, and efficient methods and models. With the advent of this technology, the application of the so-called inverse problems, where one seeks optimised parameters, geometries, configurations or models for numerical problems arising in engineering, has been gaining traction over the last years. In this context, in the past years, some packages have been developed to automate the identification of parameters (Akiba et al., 2019; Rapin & Teytaud, 2018), which have been widely applied in many areas. However, for many applications, the upfront cost of implementing interfaces for these tools is prohibitive, and specific-purpose tools are preferred to these highly flexible frameworks. Particularly in the scope of structural analysis, quickly identifying parameters of numerical models from experimental data is of utmost importance. While commercial tools are available for this task (Skacel, 2011), to the authors' best knowledge, an open-source package to this end is still lacking.

In this work, we present pilot, an open-source Python package for the automated optimisation of numerical responses, such as responses stemming from finite element simulations. In particular, the focus is placed on derivative-free optimisation, to allow compatibility with black-solvers where gradient information may be unavailable. In this context, an extensible interface for coupling with physics solvers is provided, encouraging contributions from the community to the package. As long as the solver can return a time-response for the fields of interest, it is possible to optimise it with piglot. Currently, interfaces for several solvers are included in the package, namely a solver for fitting analytical functions and interfaces for our in-house finite element code Links (derived from HYPLAS), for the commercial finite element software Abaqus, and the open-source clustering-based reduced-order model CRATE package (Ferreira et al., 2023).

For the optimisation itself, several methods are implemented and available, such as DIRECT, LIPO, and Bayesian optimisation, among others. Particularly, a significant effort has been employed in Bayesian optimisation algorithms, backed with an open-source implementation (Balandat et al., 2020) and allowing for single- and (scalarised) multi-objective optimisation of both noise-free and stochastic objectives. Furthermore, a novel composite Bayesian optimisation strategy is available for curve-fitting problems, which, in our tests, severely outperforms classical optimisation approaches (Cardoso Coelho et al., 2024).

The package also provides a built-in tool, `piglot-plot`, to visualise the results of the optimisation. There are native plotting utilities for the optimised responses, the parameter history, objective history and, for supported solvers, live plotting of the currently running case. The package also includes complete documentation for a straightforward installation and usage, supporting a simple framework for new developments. With this in mind, thorough automated testing is incorporated, ensuring the compliance of new developments.

With this package, we aim to provide a simple and effective tool for the general optimisation of numerical responses, which can be easily extended for other solvers in the community. The combination of `piglot` with `piglot-plot` provides an integrated framework that allows for easily solving inverse problems and quickly post-processing the results.

Methodology and use cases

In Figure 1, a scheme of the workflow of `piglot` is illustrated. There are two modes of initialisation available: using `.yaml` configuration files, or building the optimisation problem in a Python script. Configuration files are the simplest and recommended approach to using `piglot`, and its usage is extensively documented. During the optimisation, there is a continuous exchange of information between the physics solvers, `piglot`, and the optimisers. Whereas the optimisers are responsible for providing a candidate solution for the parameters, θ, based on the objective function value, $J(\theta)$, the physics solvers receive the parameters, θ, and compute the numerical response, here denoted as σ, accordingly. The results of the optimisation can then be visualised using the `piglot-plot` tool.

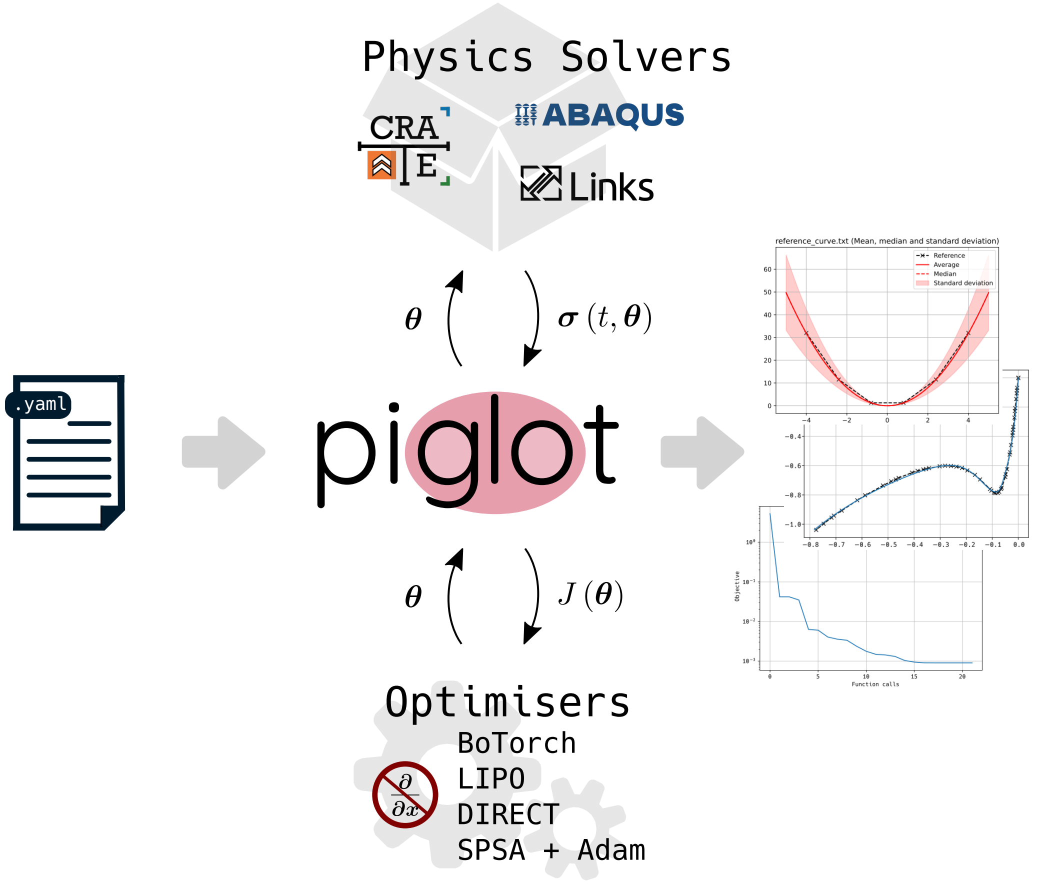

Figure 1: Schematic illustration of `piglot`.

The `piglot` package has been successfully used for the identification of constitutive parameters for classical elasto-plastic models from multi-scale simulations, crystal plasticity models with mechanically-induced martensitic transformations (Cardoso Coelho et al., 2023) and models for amorphous polymers (Carvalho Alves et al., 2023). Moreover, this tool has also demonstrated its potential in the material design of different microstructures, such as particulate PC/ABS polymer blends.

Acknowledgements

R. P. Cardoso Coelho and A. Francisca Carvalho Alves gratefully acknowledge the support provided by Fundação para a Ciência e a Tecnologia (FCT) through the scholarships with references 2020.07159.BD and 2020.07279.BD, respectively. This research has also been supported by Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI).

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.48550/arXiv.1907.10902

Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., & Bakshy, E. (2020). BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *arXiv:1910.06403 [Cs, Math, Stat]*. https://doi.org/10.48550/arXiv.1910.06403

Cardoso Coelho, R. P., Carvalho Alves, A. F., & Andrade Pires, F. M. (2024). Efficient constitutive parameter identification through optimisation-based techniques: A comparative analysis and novel composite Bayesian optimisation strategy. *Computer Methods in Applied Mechanics and Engineering*, *427*, 117039. https://doi.org/10.1016/j.cma.2024.117039

Cardoso Coelho, R. P., Vieira de Carvalho, M., & Andrade Pires, F. M. (2023). A multi-scale model combining Martensitic transformations with multi-phase crystallographic slip. *Computers & Structures*, *289*, 107174. https://doi.org/10.1016/j.compstruc.2023.107174

Carvalho Alves, A. F., Ferreira, B. P., & Pires, F. M. A. (2023). On the modeling of cavitation and yielding in rubber-toughened amorphous polymers: Fully implicit implementation and optimization-based calibration. *International Journal of Solids and Structures*, 112488. https://doi.org/10.1016/j.ijsolstr.2023.112488

Ferreira, B. P., Pires, F. M. A., & Bessa, M. A. (2023). CRATE: A Python package to perform fast material simulations. *Journal of Open Source Software*, *8*(87), 5594. https://doi.org/10.21105/joss.05594

Rapin, J., & Teytaud, O. (2018). Nevergrad — A gradient-free optimization platform. In *GitHub repository*. https://github.com/facebookresearch/nevergrad; GitHub.

Skacel, P. (2011). *HYPERFIT*. http://hyperfit.wz.cz/home.php.