pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems

Neil Wu[1], **Gaetan Kenway**[1], **John Jasa**[1], **and Joaquim R. R. A. Martins**[1]

1 Department of Aerospace Engineering, University of Michigan

DOI: 10.21105/joss.02564

Software

- Review
- Repository
- Archive

Editor: Jack Poulson

Reviewers:

- @jgoldfar
- @vissarion
- @matbesancon

Submitted: 15 July 2020
Published: 11 August 2020

Summary

pyOptSparse is an optimization framework designed for constrained nonlinear optimization of large sparse problems, providing a unified interface for a variety of gradient-free and gradient-based optimizers. By using an object-oriented approach, it maintains independence between the optimization problem formulation and the implementation of the specific optimizers. The code is MPI-wrapped to enable execution of expensive parallel analyses and gradient evaluations, such as when using computational fluid dynamics (CFD) simulations which can require hundreds of processors. During the optimization, a history file can be stored to record the optimization history, which can be used both for post-processing and for performing an optimization hot-start. A graphical user interface application is also provided to interactively plot various quantities over an optimization.

pyOptSparse considers optimization problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{with respect to} \quad & x \\ \text{such that} \quad & l \leq \begin{pmatrix} x \\ Ax \\ g(x) \end{pmatrix} \leq u \end{aligned}$$

where x is the vector of design variables and $f(x)$ is a nonlinear objective function. A is the linear constraint Jacobian, and $g(x)$ is the set of nonlinear constraint functions. pyOptSparse makes a distinction between linear and nonlinear constraints, since some optimizers have special treatments for linear constraints. Sparse linear constraints can be directly supplied in a number of different formats, and pyOptSparse will automatically handle the assembly and conversion for each optimizer. For nonlinear constraints, the sparsity pattern of the constraint Jacobian $\nabla g(x)$ can be specified as well.

Statement of Need

pyOptSparse is a fork of pyOpt (Perez, Jansen, & Martins, 2012), and as the name suggests, its primary motivation is to support the use of sparse linear and nonlinear constraints in the context of gradient-based optimization. This sets itself apart from other optimization frameworks such as SciPy (Virtanen et al., 2020) and NLopt (Johnson, 2020), which do not provide the same level of support for sparse constraints. NLopt does not support sparse Jacobians, either for linear or nonlinear constraints. While SciPy does support both, it does not allow for the Jacobians to be specified separately for each sub-block, since it treats the design vector as a single array. These frameworks also do not offer convenience features such as user-supplied optimization problem scaling, optimization hot-start, or post-processing utilities. Although pyOptSparse is a general optimization framework, it is tailored to gradient-based optimizations of large-scale problems with sparse constraints.

pyOptSparse has been used extensively in engineering applications, particularly in the field of multidisciplinary design optimization (MDO). Researchers have used it to perform aerodynamic shape optimization of aircraft wings (Secco & Martins, 2019) and wind turbines (Madsen, Zahle, Sørensen, & Martins, 2019), and aero-structural optimization of an entire aircraft (Brooks, Kenway, & Martins, 2018). pyOptSparse is also supported by OpenMDAO (Gray, Hwang, Martins, Moore, & Naylor, 2019), a popular Python framework for multidisciplinary analysis and optimization. Through OpenMDAO, pyOptSparse has been applied to problems such as low-fidelity aero-structural wing design (Chauhan & Martins, 2020) and aeropropulsive optimization of a boundary-layer ingestion propulsor (Gray & Martins, 2018).

Acknowledgements

We acknowledge the efforts of the original pyOpt developers, particularly Dr. Ruben E. Perez and Dr. Peter W. Jansen, who laid the foundation of the code. We also acknowledge the numerous pyOptSparse users who have contributed to the code over the years.

References

Brooks, T. R., Kenway, G. K. W., & Martins, J. R. R. A. (2018). Benchmark aerostructural models for the study of transonic aircraft wings. *AIAA Journal*, *56*(7), 2840–2855. doi:10.2514/1.J056603

Chauhan, S. S., & Martins, J. R. R. A. (2020). Tilt-wing eVTOL takeoff trajectory optimization. *Journal of Aircraft*, *57*(1), 93–112. doi:10.2514/1.C035476

Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., & Naylor, B. A. (2019). OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, *59*(4), 1075–1104. doi:10.1007/s00158-019-02211-z

Gray, J. S., & Martins, J. R. R. A. (2018). Coupled aeropropulsive design optimization of a boundary-layer ingestion propulsor. *The Aeronautical Journal*, *123*(1259), 121–137. doi:10.1017/aer.2018.120

Johnson, S. G. (2020). The NLopt nonlinear-optimization package. Retrieved from http://github.com/stevengj/nlopt

Madsen, M. H. A., Zahle, F., Sørensen, N. N., & Martins, J. R. R. A. (2019). Multipoint high-fidelity CFD-based aerodynamic shape optimization of a 10 MW wind turbine. *Wind Energy Science*, *4*, 163–192. doi:10.5194/wes-4-163-2019

Perez, R. E., Jansen, P. W., & Martins, J. R. R. A. (2012). pyOpt: A Python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization*, *45*(1), 101–118. doi:10.1007/s00158-011-0666-3

Secco, N. R., & Martins, J. R. R. A. (2019). RANS-based aerodynamic shape optimization of a strut-braced wing with overset meshes. *Journal of Aircraft*, *56*(1), 217–227. doi:10.2514/1.C034934

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. doi:https://doi.org/10.1038/s41592-019-0686-2