The Journal of Open Source Software

SOUPy: Stochastic PDE-constrained optimization under high-dimensional uncertainty in Python

Dingcheng Luo[1], **Peng Chen**[2], **Thomas O'Leary-Roseberry**[1], **Umberto Villa**[1], **and Omar Ghattas**[1,3]

1 Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, USA **2** School of Computational Science and Engineering, Georgia Institute of Technology, USA **3** Walker Department of Mechanical Engineering, The University of Texas at Austin, USA

DOI: 10.21105/joss.06101

Software

- Review
- Repository
- Archive

Editor: Jed Brown

Reviewers:

- @Himscipy
- @IgorBaratta

Submitted: 26 September 2023
Published: 28 July 2024

Summary

Computational models governed by partial differential equations (PDEs) are frequently used by engineers to optimize the performance of various physical systems through decisions relating to their configuration (optimal design) and operation (optimal control). However, the ability to make optimal choices is often hindered by uncertainty, such as uncertainty in model parameters (e.g., material properties) and operating conditions (e.g., forces on a structure). The need to account for these uncertainties in order to arrive at robust and risk-informed decisions thus gives rise to problems of optimization under uncertainty (OUU) (D. P. Kouri & Shapiro, 2018).

SOUPy is a Python library for solving PDE-constrained optimization problems with uncertain parameters, where we use the term *parameters* broadly to refer to sources and initial/boundary conditions in addition to PDE coefficients. The optimization problem is defined by a risk measure over a given quantity of interest (QoI), which is present as either an optimization objective or constraint (as in chance-constrained optimization). Specific attention is given to the case where the uncertain parameters are formally infinite dimensional (e.g., Gaussian random fields). The software allows users to supply the underlying PDE model, quantity of interest, and penalty terms, and provides implementations for commonly used risk measures, including expectation, variance, and superquantile/conditional value-at-risk (CVaR) (Rockafellar & Uryasev, 2000), as well as derivative-based optimizers. SOUPy leverages FEniCS (Logg et al., 2012) for the formulation, discretization, and solution of PDEs, and the framework of hIPPYlib (U. Villa et al., 2018; Umberto Villa et al., 2021) for sampling from random fields and automating adjoint-based derivative computation, while also providing interfaces to existing optimization algorithms in SciPy.

Statement of need

Problems of PDE-constrained optimization under uncertainty arise due to the ubiquity of uncertainty in natural and engineered systems. In deterministic PDE-constrained optimization, the goal is typically to optimize a quantity of interest (QoI) that is a function of the system's state and quantifies its performance, where the optimization and state variables are related through the underlying PDE model. Compared to this deterministic counterpart, PDE-constrained OUU involves an added layer of complexity, since the QoI becomes a random variable due to its dependence on the uncertain model parameters. In OUU, the cost functional and/or constraints are instead given in terms of risk measures, which are statistical quantities summarizing the QoI's distribution. A canonical example of such a risk measure is the expected value of the QoI, though other measures that account for the tail behavior of the distribution such as the variance, or superquantile/CVaR are common choices. Computation of risk measures typically requires sampling or quadrature methods to approximate the integral over the distribution of the uncertain parameter. This results in a complex optimization problem in which each evaluation of the optimization objective requires numerous solutions of the underlying PDE.

Several open-source software packages such as dolfin-adjoint (Mitusch et al., 2019) and hIPPYlib provide the capabilities for solving PDE-constrained optimization problems with generic PDEs using derivatives computed by adjoint sensitivity methods. However, these packages largely focus on the deterministic setting. On the other hand, the Rapid Optimization Library (ROL) (D. Kouri et al., 2017), released as a part of Trilinos (Trilinos Project Team, 2024), provides advanced algorithms for both deterministic and stochastic (risk measure) optimization, where support for PDE-constrained OUU is enabled by its interfaces with user-supplied state and adjoint PDE solvers.

SOUPy aims to provide a unified platform to formulate and solve PDE-constrained OUU problems using efficient derivative-based optimization methods. Users can supply the definitions for the PDE constraint, QoI, and additional penalty terms that account for the cost of design/control, and are then given the option to choose from a suite of risk measures to optimize. To this end, SOUPy makes use of FEniCS, an open source finite element library, to create and solve the underlying PDEs. The unified form language (UFL) (Alnæs et al., 2014) used by FEniCS allows users to conveniently define the PDE, QoI, and penalty terms in their variational forms. SOUPy is also integrated with hIPPYlib, an open source library for large-scale inverse problems, adopting its framework for automating adjoint-based derivative computation by leveraging the symbolic differentiation capabilities of UFL and algorithms for efficient sampling of random fields. As a core functionality, SOUPy implements sample-based evaluation of risk measures as well as their gradients and Hessians, where parallel-in-sample computation is supported through MPI. The resulting cost functionals can then be minimized using SOUPy's implementations of large-scale optimization algorithms, such as L-BFGS (Liu & Nocedal, 1989) and Inexact Newton-CG (Eisenstat & Walker, 1996; Steihaug, 1983), or through algorithms available in SciPy (Virtanen et al., 2020) using the provided interface.

Thus, SOUPy allows researchers to rapidly prototype formulations for PDE-constrained OUU problems. Additionally, SOUPy aims to facilitate the development and testing of novel algorithms. For example, SOUPy has been used in the development of Taylor approximation-based methods for the risk-averse optimization of turbulent flows (Chen et al., 2019), metamaterial design (Chen et al., 2021), and photonic nanojets (Alghamdi et al., 2022), as well as groundwater extraction (Chen & Ghattas, 2021) subject to chance constraints. It has also been used to obtain baselines for the development of machine learning approaches for PDE-constrained OUU (Luo et al., 2023).

Acknowledgements

This project is partially supported by NSF grants 2012453 and 2245674 and DOE grants DE-SC0019303 and DE-SC0023171

References

Alghamdi, A., Chen, P., & Karamehmedović, M. (2022). *Optimal design of photonic nanojets under uncertainty*. https://doi.org/10.48550/arXiv.2209.02454

Alnæs, M. S., Logg, A., Ølgaard, K. B., Rognes, M. E., & Wells, G. N. (2014). Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, *40*(2). https://doi.org/10.1145/2566630

Chen, P., & Ghattas, O. (2021). Taylor approximation for chance constrained optimization problems governed by partial differential equations with high-dimensional random parameters.

SIAM/ASA Journal on Uncertainty Quantification, *9*(4), 1381–1410. https://doi.org/10.1137/20M1381381

Chen, P., Haberman, M., & Ghattas, O. (2021). Optimal design of acoustic metamaterial cloaks under uncertainty. *Journal of Computational Physics*, *431*, 110114. https://doi.org/10.1016/j.jcp.2021.110114

Chen, P., Villa, U., & Ghattas, O. (2019). Taylor approximation and variance reduction for PDE-constrained optimal control under uncertainty. *Journal of Computational Physics*, *385*, 163–186. https://doi.org/10.1016/j.jcp.2019.01.047

Eisenstat, S. C., & Walker, H. F. (1996). Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing*, *17*(1), 16–32. https://doi.org/10.1137/0917003

Kouri, D. P., & Shapiro, A. (2018). Optimization of PDEs with uncertain inputs. In H. Antil, D. P. Kouri, M.-D. Lacasse, & D. Ridzal (Eds.), *Frontiers in PDE-constrained optimization* (pp. 41–81). Springer New York. https://doi.org/10.1007/978-1-4939-8636-1_2

Kouri, D., Ridzal, D., & Winckel, G. von. (2017). *Rapid optimization library*. Sandia National Laboratories. https://trilinos.github.io/pdfs/ROL.pdf

Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS methods for large scale optimization. *Mathematical Programming*, *45*, 503–528. https://doi.org/10.1007/BF01589116

Logg, A., Mardal, K.-A., & Wells, G. N. (Eds.). (2012). *Automated solution of differential equations by the finite element method* (Vol. 84). Springer. https://doi.org/10.1007/978-3-642-23099-8

Luo, D., O'Leary-Roseberry, T., Chen, P., & Ghattas, O. (2023). *Efficient PDE-constrained optimization under high-dimensional uncertainty using derivative-informed neural operators*. https://doi.org/10.48550/arXiv.2305.20053

Mitusch, S. K., Funke, S. W., & Dokken, J. S. (2019). Dolfin-adjoint 2018.1: Automated adjoints for FEniCS and firedrake. *Journal of Open Source Software*, *4*(38), 1292. https://doi.org/10.21105/joss.01292

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *The Journal of Risk*, *2*(3), 21–41. https://doi.org/10.21314/jor.2000.038

Steihaug, T. (1983). Local and superlinear convergence for truncated iterated projections methods. *Mathematical Programming*, *27*, 176–190. https://doi.org/10.1007/BF02591944

Trilinos Project Team. (2024). *Trilinos Project website*. https://web.archive.org/web/20240228185301/https://trilinos.github.io/rol.html

Villa, U., Petra, N., & Ghattas, O. (2018). hIPPYlib: an Extensible Software Framework for Large-scale Deterministic and Bayesian Inverse Problems. *Journal of Open Source Software*, *3*(30). https://doi.org/10.21105/joss.00940

Villa, Umberto, Petra, N., & Ghattas, O. (2021). HIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs: Part I: Deterministic inversion and linearized Bayesian inference. *ACM Transactions on Mathematical Software*, *47*(2). https://doi.org/10.1145/3428447

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (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