

Student Project:

Solving Structural Optimization and System Identification Problems with SciPy Algorithms

Numerical optimization is the common engine behind three tasks that appear throughout computational structural mechanics: calibrating a model against measurements (System Identification), distributing material within a design domain (Topology Optimization), and moving boundaries to improve structural performance (Shape Optimization). Although the underlying physics differs, all three reduce to the minimization of an objective functional subject to bounds, equality and inequality constraints.

The open-source SciPy library offers a broad collection of algorithms for exactly this purpose — quasi-Newton methods (BFGS, L-BFGS-B), sequential quadratic programming (SLSQP), interior-point methods (trust-constr), derivative-free strategies (Nelder–Mead, Powell, COBYLA) and global searches (differential evolution, basin hopping). Their efficiency and robustness on structural problems, however, vary strongly with the problem size, the constraint formulation and the quality of the supplied gradients.

This student project develops a compact and reproducible benchmark suite in which the three problem classes above are solved with a wide range of `scipy.optimize` algorithms.

Work Packages

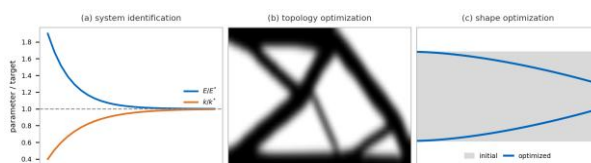
1. **Literature & Tool Review:** Survey the optimization algorithms available in `scipy.optimize` and classify them by gradient requirement, constraint handling and convergence properties.
2. **Benchmark Problem Setup:** Implement three model problems — (i) FE model updating from displacement or eigenfrequency data, (ii) SIMP-based topology optimization of a 2D cantilever under a volume constraint, and (iii) shape optimization of a boundary parameterized by surface points.
3. **Benchmarking & Evaluation:** Compare the algorithms in terms of objective value, number of function and gradient evaluations, runtime, and robustness with respect to the starting point and scaling; summarize the findings in convergence plots and performance profiles.

The work is carried out in **Python** using **NumPy / SciPy** and **Matplotlib**. The structural models may be realized with a lightweight in-house finite element code or, for the larger examples, with **KratosMultiphysics**. All studies are documented in reproducible scripts or Jupyter notebooks.

Applicants should have solid programming skills in **Python**, a good understanding of **numerical methods**, **FEM** and the **basics of mathematical optimization**. Prior experience with structural optimization or system identification is helpful but not required.

References:

- [1] Virtanen, P., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17, 261–272.
- [2] Bendsøe, M. P., & Sigmund, O. (2004). *Topology Optimization: Theory, Methods and Applications*. Springer.
- [3] Ansari, T. S. A., Löhner, R., Wüchner, R., et al. (2025). Adjoint-based recovery of thermal fields from displacement or strain measurements. *Computer Methods in Applied Mechanics and Engineering*.



Three model problem classes to be solved with `scipy.optimize`.

Supervisor: Dr.-Ing. Ihar Antonau (ihar.antonau@tum.de)

Start date: Flexible