This is the reference implementation of Powell's derivative-free optimization solvers, namely COBYLA, UOBYQA, NEWUOA, BOBYQA, and LINCOA.

The goal is to implement these solvers in modern languages — first **modern** Fortran (>= F2003), and then MATLAB, Python, and probably Julia and R. It will be a faithful implementation, in the sense that the new code will be mathematically equivalent to Powell's, except for the bug fixing and improvements that we make intentionally.

The focus is to implement Powell's solvers in a modularized and structured way so that they are readable, maintainable, and extendable. The new code will have no GOTO (of course) and will use matrix-vector procedures instead of loops whenever possible.

This is not a trivial mission due to the delicacy of Powell's algorithms and the unique style of his code. We started The Fortran code by refactoring Powell's code into the free form via a small MATLAB tool written by ourselves. However, such refactored code is far from what we want, because it will inherit completely the structure and style of Powell's code except for the format. Extensive modifications are needed to reorganize the code. To maintain the faithfulness quality of our implementation, intensive tests are conducted each every tiny modification. The tests are automated with the help of GitHub Actions. As of July 2022, more than 20,000 `workflows` have been run by GitHub Actions (see [https://github.com/zequipe/gitpersonal/actions] and [https://github.com/zequipe/pdfo_ref/actions]). Normally, each workflow consists of more than 5 tests that are conducted in parallel, each test taking from tens of minutes to several hours (the maximum is 6 hours, after which the workflow will be canceled automatically). In other words, our implementation has been tested for $\sim 10^5$ hours, or $\sim 10$ years.

This package is part of a research project funded by the Hong Kong Research Grants Council and the Hong Kong Polytechnic University (PolyU). It is still under intensive development, and there is no release yet. If you want to use the above-mentioned solvers, see the website and repository of PDFO instead.

Dedicated to late Professor M. J. D. Powell FRS (1936–2015).