sotb-wrapper: a Python wrapper of the SEISCOPE optimization toolbox library

Oscar F. Mojica, Senai Cimatec Supercomputing Center

Resumo

Python language has been increasingly adopted by the geosciences community, establishing itself as one of the main choices for developing numerical applications, either due to its features to facilitate data analysis or visualization (observe examples at https://github.com/softwareunderground/awesome-open-geoscience). Python offers its users several open-source frameworks, libraries, and development tools to curtail development time without increasing development cost and this, without a doubt, is one of the main reasons for its popularity. However, despite the available large spectrum of frameworks/libraries, it is often not possible to find a single one that best satisfies user needs. In this regard and in relation to gradient-based numerical optimization packages, there seem to be only a few options (i.e., SciPy.optimize), which contrasts with low-level languages such as Fortran and C/C++. In this last case, a good example of such packages is the SEISCOPE optimization toolbox, which is coded in Fortran. The toolbox implements four different optimization schemes: the steepest-descent, the nonlinear conjugate gradient, the l-BFGS method, and the truncated Newton methods along with a line-search strategy that ensures its robustness. In spite of recent efforts by a community of developers and users to turn Fortran into a better language for scientific computing (i.e., adding interactivity, package management system, etc), it will take some time before it is adopted by new users in certain domains of computational science. Under these circumstances, constructing Python interfaces to Fortran codes becomes particularly relevant, as it not only enables the Python user base to interact with those codes but also fills the gaps in the python tooling ecosystem with an additional advantage in performance. Therefore, in this paper, I introduce sotb-wrapper, an open Python-wrapped version of the SEISCOPE toolbox that makes the functionality of gradient-based optimizers written in Fortran available to Python programmers. Among the various available tools to create a Python interface to Fortran, I chose to use the Python built-in ctypes module to write the wrapper. This choice is motivated by the fact that the toolbox makes use of derived types, a feature not supported by automated wrapping tools as f2py. With this approach, certain modifications were made to the original code without changing the reverse communication protocol proper of the toolbox. In accordance with this protocol, the computation of the solution of the optimization problem is conducted via a minimization loop controlled by the user in which the chosen minimization wrapped routine from the toolbox is called at each iteration. Depending on the return values of a communication flag, the user performs the action required by the solver at a given point, namely, computation of the objective function and its gradient, application of a preconditioner, and computation of a Hessian-vector product. The process continues until the convergence is reached. This protocol provides a high level of flexibility, as code depending on the physics of the problem is separated from that responsible for the minimization and also facilitates the integration with other frameworks as demonstrated in the examples available in Github at https://github.com/ofmla/seiscope_opt_toolbox_w_ctypes.