A comparison of three open-source numerical optimization python libraries for full-waveform inversion.

Oscar F. Mojica, Senai Cimatec Supercomputing Center & National Institute of Petroleum Geophysics (INCT-GP)

Abstract

Geoscientists favor Python for its user-friendly interface and scientific packages that support application implementation. Python's capabilities make it particularly useful for seismic full waveform inversion (FWI), which can be implemented quickly using available packages. We compare three open-source gradient-based optimization Python packages - scipy.optimize, sotb-wrapper, and PyROL - for solving the FWI optimization problem. The comparison is based on the packages' core features, documentation, and learning curves evaluated through the implementation of a 2D time-domain FWI application, built using the Devito modeling engine along with the aforementioned optimization packages.

All three packages provide direct Python interfaces to compiled-language optimizers and support box-constraints. The scipy.optimize module contains various useful methods for optimizing different kinds of objective functions. Its minimize() function can be used to solve FWI. Sotb-wrapper allows the use of Seiscope optimization toolbox from Python and include four optimization algorithms. Diferently to scipy.optimize, where FWI can be solved with a straightforward function call, the user is responsible for programming the optimization workflow. PyROL serves as an interface between Python and Rapid Optimization Library (ROL), which provides many optimization algorithms, line-search methods, and Wolf-type conditions. Implementing FWI with PyROL is more complex and requires several steps, but ultimately FWI is solved by calling the solve() method from the `OptimizationSolver` class. We ran a low frequency FWI using the Marmousi2 model, which was discretized into 88x426 grid points with a 40-m grid interval. We smoothed both the true and the initial models since it was a low-frequency experiment. Seismic modeling was performed to produce experimental synthetic data using a Ricker wavelet with a peak frequency of 4 Hz. We used 16 shots and 426 receivers evenly distributed across surface of the model. FWI was executed separately for each package with the l-BFGS algorithm, available in all three. Convergence to the true solution demonstrated each package's effectiveness.

In terms of documentation, Scipy.optimize has extensive and detailed resources with simple examples. Sotb-wrapper has basic tutorials on its GitHub repository that are well-suited for novice users. PyROL's documentation website is not comprehensive, but basic examples on its Bitbucket repository can assist users in gaining a better understanding of the package's features. When it comes to the learning process, Scipy.optimize's documentation proves to be a valuable resource, enabling a swift and efficient start. Sotb-wrapper's examples comprehensively encompass the necessary elements for solving optimization problems with the package, making it remarkably user-friendly. On the other hand, while PyROL boasts an impressive array of features, its lack of abundant examples can prove to be a hurdle for beginners, demanding extra perseverance to fully comprehend and harness its capabilities.

Although comparative studies like ours are important, they are rare. FWI practitioners who wish to utilize python optimization packages for their research or applications are confronted with a challenging decision due to the various available options and the limited scientific evaluation conducted by the research community to compare them. Our study aimed to simplify this decision-making process and provide guidance to practitioners in selecting the package that best suits their needs.