Geophysical data processing and modeling by open source resources in Python

Vahid E. Ardestani [*,**]

** Institute of Geophysics, University of Tehran, Center of Excellence in Survey Engineering and Disaster Management, Tehran, Iran, E-mail: ebrahimz@ut.ac.ir*

Abstract

Geophysical data processing and modeling by applying open source resources are very quickly improving. Theses open sources are accessible through several websites. These open sources include many aspects of data processing and modeling from data corrections and processing to forward and inversion modeling. Theses open sources are developed and provided based on newest and The most popular methods by utilizing Python programming. These Python scripts cover a vast majority of different methods in geophysics. I try to express firstly the advantages of Python and then introducing some of the most appropriate websites which can be used for accessing to open source resources.

Key words: open source resources, Python, data processing, modeling.

1- Python

In this section, the advantages and disadvantages of Python are expressed:

1-1- The advantages of Python coding are as follows:

- **Extensive Libraries**

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don't have to write the complete code for that manually.

- **Extensible**

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

- **Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

- **Improved Productivity**

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

- **IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

- **Simple and Easy**

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn**, **understand**, and **code**. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

- **Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory**. This further aids the readability of the code.

1-2- Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

- **Speed Limitations**

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

- **Weak in Mobile Computing and Browsers**

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**. The reason it is not so famous despite the existence of Brython is that it isn't that secure.

- **Underdeveloped Database Access Layers**

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

2- SOME OF THE LIBRARIES AND OPEN SOURCES IN Python :

- **Numpy:** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **Scipy:** SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.
- **Pandas:** In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.
- **Matplotlib:** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.
- **Scikit-learn:** Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, ...
- **IPython:** IPython is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history. IPython provides the following features: Interactive shells.

3- SimPEG: Simulation and Parameter Estimation in Geophysics

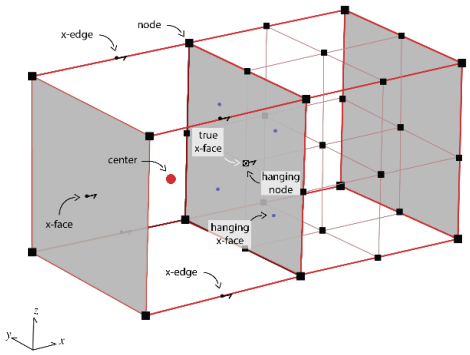
An open source python package for simulation and gradient based parameter estimation in geophysical applications. Contribute to a growing community of geoscientists building an open foundation for geophysics. SimPEG provides a collection of geophysical simulation and inversion tools that are built in a consistent framework including: Gravity, Magnetics, Viscous remanent magnetization, Direct current resistivity, Induced polarization, Electromagnetics, Richards Equation, Seismic.

3-1- Discretizing

The most powerful aspects of Simpeg are its capability in dicretizing the model space, forward modeling and inversion for different problems in geophysics.
For discretizing five advanced methods and scripts are available including:

Tensor, octree, quadtree, cylindrical and Logicaly rectangular mesh meshes. In Fig. (1) different aspects of a cubic mesh are presented.

Fig. (1)



4- Inversion Components

Another great aspect of SimPEG is its ability in modeling.
The SimPEG inversion framework is modular, allowing you to explore, experiment and iterate over a variety of approaches to the inverse problem such as: Optimization, Regularization, Data misfits, Parametric models, Joint inversions
In following fellow chart, The different aspects of modeling through Simpeg are demonstrated:

Flowchart (1