Computers & Geosciences 85 (2015) 142–154

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Review article

SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications

Rowan Cockett [a,*], Seogi Kang [a], Lindsey J. Heagy [a], Adam Pidlisecky [b], Douglas W. Oldenburg [a]

[a] *Geophysical Inversion Facility, University of British Columbia, Canada*
[b] *University of Calgary, Canada*

ARTICLE INFO

Article history:
Received 14 January 2015
Received in revised form
11 August 2015
Accepted 23 September 2015
Available online 26 September 2015

Keywords:
Geophysics
Numerical modeling
Inversion
Electromagnetics
Sensitivities
Object-oriented programming

ABSTRACT

Inverse modeling is a powerful tool for extracting information about the subsurface from geophysical data. Geophysical inverse problems are inherently multidisciplinary, requiring elements from the relevant physics, numerical simulation, and optimization, as well as knowledge of the geologic setting, and a comprehension of the interplay between all of these elements. The development and advancement of inversion methodologies can be enabled by a framework that supports experimentation, is flexible and extensible, and allows the knowledge generated to be captured and shared. The goal of this paper is to propose a framework that supports many different types of geophysical forward simulations and deterministic inverse problems. Additionally, we provide an open source implementation of this framework in Python called SimPEG (Simulation and Parameter Estimation in Geophysics, http://simpeg.xyz). Included in SimPEG are staggered grid, mimetic finite volume discretizations on a number of structured and semi-structured meshes, convex optimization programs, inversion routines, model parameterizations, useful utility codes, and interfaces to standard numerical solver packages. The framework and implementation are modular, allowing the user to explore, experiment with, and iterate over a variety of approaches to the inverse problem. SimPEG provides an extensible, documented, and well-tested framework for inverting many types of geophysical data and thereby helping to answer questions in geoscience applications. Throughout the paper we use a generic direct current resistivity problem to illustrate the framework and functionality of SimPEG.

Contents

* Corresponding author.
E-mail address: rcockett@eos.ubc.ca (R. Cockett).

http://dx.doi.org/10.1016/j.cageo.2015.09.015

1. Introduction and motivation

Geophysical surveys can be used to obtain information about the subsurface as the responses that are measured depend on the physical properties and contrasts in the earth. Inversions provide a mathematical framework for constructing physical property models consistent with the data collected by these surveys. The data collected are finite in number while the physical property distribution of the earth is continuous. Thus, inverting for a physical property model from geophysical data is an ill-posed problem, meaning that no unique solution explains the data. Furthermore, the data may be contaminated with noise. Therefore, the goal of a deterministic inversion is not only to find a model consistent with the data, but must be to find the 'best' model that is consistent with the data.[1] The definition of 'best' requires the incorporation of assumptions and *a priori* information, often in the form of an understanding of the particular geologic setting or structures (Constable et al., 1987; Oldenburg and Li, 2005; Lelièvre et al., 2009). Solving the inverse problem involves many moving pieces that must work together, including physical simulations, optimization, linear algebra, and incorporation of geology. Deterministic geophysical inversions have been extensively studied, and many components and methodologies have become standard practice. With increases in computational power and instrumentation quality, there is a greater drive to extract more information from the geophysical data. Additionally, geophysical surveys are being applied in progressively more challenging environments. As a result, the geosciences are moving towards the integration of geological, geophysical, and hydrological information to better characterize the subsurface (e.g. Haber and Oldenburg, 1997; Doetsch et al., 2010; Gao et al., 2012). This is a scientifically and practically challenging task (Li and Oldenburg, 2000b; Lelièvre et al., 2009). These challenges, compounded with inconsistencies between different data sets, often makes the integration and implementation complicated and/or non-reproducible. The development of new methodologies to address these challenges will build upon, as well as augment, standard practices; this presupposes that researchers have access to consistent, well-tested tools that can be extended, adapted and combined.

There are many proprietary codes available that focus on efficient algorithms and are optimized for a specific geophysical application (e.g. Kelbert et al., 2014; Li and Key, 2007; Li and Oldenburg, 1996b, 1998). These packages are effective for their intended application, for example, a domain specific large-scale geophysical inversion or a tailored industry workflow. However, many of these packages are 'black-box' algorithms, that is, they cannot easily be interrogated or extended. As researchers, we require the ability to interrogate and extend ideas; this must be afforded by the tools that we use. Accessibility and extensibility are the primary motivators for this work. Other disciplines have approached the development of these tools through open source initiatives using interpreted languages, such as Python, for example, Astropy in astronomy (Astropy Collaboration et al., 2013) and SciPy in numerical computing (Jones et al., 2001). Interpreted languages facilitate interactive development using scripting, visualization, testing, and interoperability with code in compiled languages and existing libraries. Furthermore, many open source initiatives have led to communities with hundreds of researchers contributing and collaborating using social coding platforms, such as GitHub (https://github.com). There are also initiatives in the geophysical forward and inverse modeling community targeting specific geophysical applications (cf. Hansen et al., 2013; Hewett and Demanet, 2013; Uieda et al., 2014; Kelbert et al., 2014; Harbaugh, 2005). We are interested in creating a community around geophysical simulations and gradient based inversions. To create a foundation on which to build a community, we require a comprehensive framework that is applicable across domains and upon which researchers can readily develop their own tools and methodologies. To support these goals, this framework must be modular and its implementation must be easily extensible by researchers.

The goal of this paper is to present a comprehensive framework for simulation and gradient based parameter estimation in geophysics. The core ideas from a variety of geophysical inverse problems have been distilled to create this framework. We also provide an open source library written in Python called SimPEG (Simulation and Parameter Estimation in Geophysics, http://github.com/simpeg/simpeg). Our implementation has core dependencies on SciPy, NumPy, and Matplotlib, which are standard scientific computing packages in Python (Jones et al., 2001; Van Rossum and Drake, 1995; Oliphant, 2007; Hunter, 2007). SimPEG includes staggered grid, mimetic finite volume discretizations on structured and semi-structured meshes. It interfaces to standard numerical solver packages, convex optimization algorithms, model parameterizations, and visualization routines. We make use of Python's object-oriented paradigm leading to modular code that is extensible through inheritance and subtype polymorphism. SimPEG follows a fully open source development paradigm (Feller and Fitzgerald, 2000), and uses the permissive MIT license. Throughout its development, we have focused on modularity, usability, documentation, and extensive unit-testing (Wilson et al., 2014; Holscher et al., 2010; Kalderimis and Meyer, 2011; Merwin et al., 2015). See the website http://simpeg.xyz for up-to-date code,

[1] Alternatively, the inverse problem can be formulated in a probabilistic framework, see for example Tarantola (2005) and Tarantola and Valette (1982). In this paper we will focus our attention on the deterministic approach.