

Development practices and lessons learned in developing SimPEG

Rowan Cockett, Lindsey Heagy, Seogi Kang & Gudni Rosenkjaer

Inverse modelling provides a mathematical framework for constructing a model of physical property distributions in the subsurface that are consistent with the data collected in geophysical surveys. The geosciences are increasingly moving towards the integration of geological, geophysical, and hydrological information to better characterize the subsurface. This integration must span disciplines and is not only challenging scientifically, but additionally the inconsistencies between conventions often makes implementations complicated, non reproducible, or inefficient. SimPEG is an open-source, multi-university effort aimed at providing a generalized framework for solving forward and inverse problems. SimPEG includes finite volume discretizations on structured and unstructured meshes, interfaces to standard numerical solver packages, convex optimization algorithms, model parameterizations, and visualization routines. The SimPEG package (<http://simpeg.xyz>) supports an ecosystem of forward and inverse modelling applications, including electromagnetics, vadose zone flow, seismic, and potential fields, that are all written with a common interface and toolbox. The goal of SimPEG is to support a community of researchers with well-tested, extensible tools, and encourage transparency and reproducibility both of the SimPEG software and the geoscientific research it is applied to.

In this presentation, we will share some of the lessons we have learned in designing the modular infrastructure, testing and development practices of SimPEG. We will discuss our use of version control, extensive unit-testing, continuous integration, documentation, issue tracking, and resources that facilitate communication between existing team members and allows new researchers to get involved. These practices have enabled the use of SimPEG in research, industry, and education as well as the ability to support a growing number of dependent repositories and applications. We hope that sharing our practices and experiences will help other researchers who are creating communities around their own scientific software. As this session suggests, “software is critical to the success of science,” but, it is the **communities** of researchers that must be supported as we strive to create top quality research tools.