Reproducible Research in SEP

Yang Zhang and Sjoerd de Ridder

The necessity and idea of reproducible scientific computational research occurred to Jon Claerbout and his students of the Stanford Exploration Project (SEP) in the 80's. And beginning 1991 they completed a framework based on Cake. The reproducible electronic document ideally constituted a complete package including Tex source file, software source code files, and documented processing flows, controlled by a framework in *Cake* (and later *GNU Make*). The entire electronic document could be recomputed by the reader, if the reader is equipped with an appropriate computing environment. *GNU Make* is an open-source tool originally devised to maintain the flows for building multi sources software. Instead of just specifying source file dependencies we included data file dependencies in our Makefiles. An appropriate set of generic *Make rules*, were built allowing clear archiving of a given seismic processing flow.

Following this ideology, other reproducible software systems have been proposed and advocated by different research groups, using often python based software construction tools. The SEP reproducibility framework still relies on a set of *Make rules* supplied with SEPLib (our command-line based seismic data processing environment that allows easy user customization). The end product of *Make flows* varies. In the research stage the result is usually an image displayed on the screen often using the graphics package *vplot.* When the research reaches some stage of completion the final result is pdf file. These images are then converted to an appropriate image format and compiled into electronic documents, SEP currently used RSF's SCons framework to compile latex documents.

Various advanced reproducibility technology is currently developed and used in SEP. Modern seismic data requires 3D, interactive, viewing capability. SEPlib contains a program '*Sep_cube'*, which can view several different data formats including SEPlib data and produce lossless postscript files. To maintain the strict command-line reproducibility, *Sep_cube* contains a text-file output option to reproduce the view. In the future similar text-string outputs, will allow creation of a *Sep_cube* customized display/view at lossless quality through conventional 2D *vplot* writing software. Another leap is provided by interactive pdf documents. A python tool catches a query initialized by pushing a button in the pdf file. The python tool runs a specified command (selected from a supplied text file), with almost infinite freedom of what to run. Displaying vplot files through tube, or running a code with different input parameters are just two of the potential applications.

In this presentation, we will briefly introduce the current reproducible research workflow in SEP. An example will be explained to better illustrate how it is done in practice. Last we will comment on the shortcomings/limitation of our current framework.