PyAspect: *Workflow and automation tools for SPECFEM3D*

Authors: Thomas Cullison[1], Iban Masfara[2], Rhys Hawkins[1], Jeannot Trampert[1]

[1]Utrecht University
[2]TU Delft

Background and motivation

When trying to maintain reproducible results for FWI and earthquake source inversion, there are several things that need to be considered. These inversion workflows typically require multiple iterations where each iterations contains multiple steps. To be reproducible, any parameters used (e.g. priors, posteriors, models, smoothing, etc.) need to be tracked, as well as the results of each of these iterations and their internal steps. This type of tacking can be complex in its own right; however, some of these inversion steps are often computationally expensive and thus require the use of HPC clusters (especially for 3D inversions). Depending on the HPC environment, it's possible that whole inversion workflows could be run exclusively from within such an environment, but this may not be practical. Connecting to a HPC environment requires internet access and often its own special software tools, and this can limit research that could otherwise be performed offline (e.g. from a laptop). For example, analyzing data results and graphs are task that could be examined offline. Furthermore, tracking software and environment differences between at-home/office computing environments has its own challenges. To mitigate reproducibility issues related to complex research workflows, including research that makes use of multiple computing environments (i.e. laptops and HPC clusters), Beg et al. propose and demonstrate the use of Jupyter related software infrastructure for conducting reproducible research that require complex workflows.

Although we do not discuss in detail this Jupyter infrastructure, it suffices to say that we developed our research code with respect to a subset of actions describe within this paradigm. To completely comply with the paradigm would require significant changes to the HPC environment we used for our research, and it is likely that many other HPC environment would also have to make significant changes to completely comply, as well. Therefore, we focused on a development strategy that could be integrated into most HPC environments, and that could also be extended to the full Jupyter infrastructure should that become necessary. This strategy involved writing new code/libraries using Python packages as much as possible and conducting the inversion workflow using Jupyter Notebooks as much as possible.

As previously stated, most of our research code regarding the inversion workflow was written in Python and is executed using Jupyter Notebooks (although some of our code is contained within shell scripts). Two separate python packages were created to assist with our inversion workflow. The first package, gnam, was written to automatically acquire seismic data for events that were recorded by the G-stations that cover the Groningen field area. Note, further detail of this package lies outside the scope of this presentation. The second package, PyAspect, was written to help automate and track inversion workflows for

SPECFEM3D Cartesian (SPEC). The SPEC software package implements a spectral-element method for modeling seismic sources, including moment-tensors, and for conducting FWI and source inversions. Unfortunately, the SPEC codes were designed to be executed in a step-by-step manner from either a command-line or from a script. The foundation (forward modeling and adjoint wavefield parts) of SPEC was developed in FORTRAN-90, but there is a patchwork of codes/packages built around the foundation to assist with the inversion workflows. These codes/packages were written using various languages including C/C++, CUDA, Pearl, FORTRAN, shell-scripts, etc. They are also not fully integrated with SPEC, which means they may have different OS library requirements and might need to be compiled separately. All of these issues can lead to a condition referred to as "software collapse" or "software rot" (Hinsen, 2019). Furthermore, SPECFEM3D does not have a robust method tracking metadata relating sources, receivers, components, and seismograms. Therefore, to mitigate reproducibility issues and to help consolidate these inversion workflows, we developed workflow tools (PyAspect) to help streamline these workflows and incentivize reproducibility. It should be noted that before beginning the development PyAspect, there was Python package, SeisFlows, which was developed to help streamline the inversion workflows for SPEC. However, this package became out of sync with the SPEC codes; it was not compatible with Python 3.x; and we were not successful in being able to actively use the package. Integrating the use of SeisFlows into PyAspect, is desirable, but this is a future work, and it may require significant modification of SeisFlows.

References:

- M. Beg et al., "Using Jupyter for Reproducible Scientific Workflows," in Computing in Science & Engineering, vol. 23, no. 2, pp. 36-46, 1 March-April 2021, doi: 10.1109/MCSE.2021.3052101.
- Konrad Hinsen. Dealing With Software Collapse. Computing in Science and Engineering, Institute of Electrical and Electronics Engineers, 2019, 21 (3), pp.104-108. 10.1109/MCSE.2019.2900945. hal-02117588
- Ryan T. Modrak, Dmitry Borisov, Matthieu Lefebvre, Jeroen Tromp, SeisFlows -- Flexible waveform inversion software, Computers & Geosciences, Volume 115, 2018, Pages 88-95.