

TBA

Felix Herrmann

Earth and Ocean Sciences, University of British Columbia
fherrmann@eos.ubc.ca

Abstract

Inverse problems in (exploration) seismology are known for their large to very large scale. For instance, certain sparsity-promoting inversion techniques involve vectors that easily exceed 2^{30} unknowns while seismic imaging involves the construction and application of matrix-free discretized operators where single matrix-vector evaluations may require hours, days or even weeks on large compute clusters. For these reasons, software development in this field has remained the domain of highly technical codes programmed in low-level languages with little eye for easy development, code reuse and integration with (nonlinear) programs that solve inverse problems.

Following ideas from the Symes' Rice Vector Library and Bartlett's C++ object-oriented interface, Thyra, and Reduction/Transformation operators (both part of the Trilinos software package), we developed a software-development environment based on overloading. This environment provides a pathway from in-core prototype development to out-of-core and MPI 'production' code with a high level of code reuse. This code reuse is accomplished by integrating the out-of-core and MPI functionality into the dynamic object-oriented programming language Python. This integration is implemented through operator overloading and allows for the development of a coordinate-free solver framework that (i) promotes code reuse; (ii) analyses the statements in an abstract syntax tree and (iii) generates executable statements.

In the current implementation, we developed an interface to generate executable statements for the out-of-core unix-pipe based (seismic)

processing package RSF-Madagascar (`rsf.sf.net`). The modular design allows for interfaces to other seismic processing packages and to in-core Python packages such as `numpy`.

So far, the implementation overloads linear operators and element-wise reduction/transformation operators. We are planning extensions towards nonlinear operators and integration with existing (parallel) solver frameworks such as Trilinos.