Geophysical Research Abstracts
Vol. 14, EGU2012-3979, 2012
EGU General Assembly 2012

Combining Numerical and Symbolic Computing in *Python*

L. Gross, C. Altinay, J. Fenwick, and L. Gao

The University of Queensland, School of Earth Sciences, Earth Systems Science Computational Center, St Lucia, Australia (l.gross@uq.edu.au)

Escript is a Python-based programming tool for modeling physical processes with a focus on Earth Sciences, see https://launchpad.net/Escript-finley. Users can easily implement complex models represented through strongly or loosely coupled partial differential equations (PDEs). Model scripts can be run on desktop computers as well as large scale parallel computers without any modifications. The key components in *Escript* are general linear PDEs and PDE coefficients which are both represented through appropriate Python classes. Users can combine instances of these classes to build and couple mathematical models without the need to access the data structures used for the numerical representation on the C/C++ level. *Escript* has been successfully applied to a broad variety of problems including mantel convection, melting processes, volcanic flow, earthquakes, mineralization, plate subduction, and tsunamis.

In many situations, e.g. when solving nonlinear PDEs using the Newton-Raphson scheme, solving inversion problems, or for sensitivity analysis, a linearization of the PDE is required. Due to the complexity of this operation it is highly desirable that the calculation of the linearization is (completely or partially) automated. Computer algebra provides an appropriate framework to perform this task. The focus of recent development work on *Escript* has been the introduction of symbolic representations of (nonlinear) PDEs and PDE coefficients allowing for automated calculations of PDE linearizations with respect to the solution as well as PDE coefficients. The actual solution process (e.g. the Newton-Raphson iteration scheme to solve a nonlinear PDE) still requires the solution of a series of linear PDE which are solved numerically (e.g. using finite elements).

In the paper we will present a concept of symbolical representation of PDEs for *Escript* outline its implementation for *Python* and discuss some aspects of efficiency at the interface between symbolic and numerical calculations. We will illustrate the usage of this concept to solve an inversion problem for spatial variable PDE parameters.

Acknowledgment: This work is supported by the Australian Geophysical Observing System EIF (AGOS), the Queensland State Government and The University of Queensland.