Software infrastructure for solving non-linear partial differential equations and its application to modeling crustal fault systems

*L. Gross [1,3], J. Gerschwitz [1,3], P. Hornby [2], P. Mora [1,3], E. Saez [1,3], C. Wang [1,3], D. Weatherley [1,3] and H. Xing [1,3]

[1] The University of Queensland, Brisbane, Australia
[2] CSIRO Exploration and Mining, Perth, Australia
[3] Australian Computational Earth Systems Simulator MNRF, Brisbane, Australia

The design of modern simulation codes has three conceptional layers: the application layer, the mathematical layer, and the numerical algorithm layer. The latter is provided in the form of C/C++ libraries which functions are solving computational intensive, typically linear problems such as linear partial differential equations (PDEs). The mathematical layer allows the modeler to define models and to implement high level solution algorithms (e.g. Newton-Raphson scheme, Crank-Nicholson scheme) or import tested algorithms from available libraries. The kernels of the models are solvers provided through the numerical algorithm layer. An interactive programming language such as Python is the ideal environment for the mathematical layer. Finally, to provide an easy-to-use application environment and to link the simulation code with databases or other simulation codes, a web interface is (semi-automatically) built to edit the XML input file for the modeling code.

In the talk, we will discuss the advantages and disadvantages of this concept in the context of solving non-linear PDE. We will present the modeling environment *escript* which is a prototype implementation toward such a software system within Python. Key components of *escript* are the *Data* class and the *PDE* class. Objects of the *Data* class allow generating, holding, accessing, and manipulating data in such a way that the actual, in the particular context best, representation of the data is transparent to the user. They are also the key to establish connections with external data sources such as data bases and other simulations. *PDE* class objects are defining (linear) PDEs to be solved by a numerical library, in this case the parallel finite element code *Finley*.

As a case study we will demonstrate how this environment is applied to the simulation of crustal fault systems which is a key for the understanding and prediction of earthquakes. The model, which bases on non-linear, time-dependent PDEs, is considering two different time scales: quasi-static loading , which gradually increases stress in the system, and dynamic rupture, which rapidly redistributes stress. The latter is noticeable as an earthquake. On the fault, the approach uses a frictional constitutive relationships that is slip and slip-rate dependent. We will present simulations of the fault system in Flinders Ranges of South Australia.