Geophysical Research Abstracts
Vol. 14, EGU2012-14116, 2012
EGU General Assembly 2012

Recent progress in modelling 3D lithospheric deformation

B.J.P. Kaus (1), A Popov (1), and D.A. May (2)

(1) Institute of Geosciences, University of Mainz, Germany, (2) Institute of Geophysics, ETH Zurich, Switzerland

Modelling 3D lithospheric deformation remains a challenging task, predominantly because the variations in rock types, as well as nonlinearities due to for example plastic deformation result in sharp and very large jumps in effective viscosity contrast. As a result, there are only a limited number of 3D codes available, most of which are using direct solvers which are computationally and memory-wise very demanding. As a result, the resolutions for typical model runs are quite modest, despite the use of hundreds of processors (and using much larger computers is unlikely to bring much improvement in this situation).

For this reason we recently developed a new 3D deformation code,called LaMEM: Lithosphere and Mantle Evolution Model. LaMEM is written on top of PETSc, and as a result it runs on massive parallel machines and we have a large number of iterative solvers available (including geometric and algebraic multigrid methods). As it remains unclear which solver combinations work best under which conditions, we have implemented most currently suggested methods (such as schur complement reduction or Fully coupled iterations). In addition, we can use either a finite element discretization (with Q_1P_0, stabilized Q_1Q_1 or Q_2P_{-1} elements) or a staggered finite difference discretization for the same input geometry, which is based on a marker and cell technique). This gives us he flexibility to test various solver methodologies on the same model setup, in terms of accuracy, speed, memory usage etc.

Here, we will report on some features of LaMEM, on recent code additions, as well as on some lessons we learned which are important for modelling 3D lithospheric deformation. Specifically we will discuss:

1) How we combine a particle-and-cell method to make it work with both a finite difference and a (lagrangian, eulerian or ALE) finite element formulation, with only minor code modifications code

2) How finite difference and finite element discretizations compare in terms of accuracy, speed, and robustness for iterative solvers.

3) How the code scales on massive parallel machines (we have performed simulations on over 4096 cores with over a billion degrees of freedom).

4) We will give an overview of some recent scientific applications of the code.