A FRAMEWORK FOR ADAPTIVE MESHES IN QUASI-STATIC MODELING

W. Bangerth and L. L. Lavier
University of Texas at Austin

The problem of resolution of different scales is a question that recurs in many fields of geodynamical modeling. The main reason for this is that the processes affecting large scale motion are expressed through physical processes at much smaller scales. Examples of these phenomena are the localization of deformation on faults, diking during melt migration and land-sliding or slumping related to surface processes such as erosion and sedimentation. One prime example is the motion of entire tectonic plates and their interaction at faults: While plate motions are driven by thermal and chemical processes occurring over 1000s of kilometers in the lithosphere and astenosphere, a large part of their dynamics is determined by what happens at plate boundaries which themselves can be as small as a few meters or kilometers.

Understanding geodynamic processes requires resolving these different scales numerically as best as possible. However, for a variety of reasons, present-day computer codes are mostly not capable of doing so. Even in simpler cases, small scales are usually 2-3 orders of magnitude smaller than the large scales. If small scales should be resolved with at least 10 grid points, this requires 1.e9 to 1.e12 grid points when uniformly refined meshes are used. For simulations over realistic time spans this is many orders of magnitude away from what is possible with today's computing facilities. Thus, realistic simulations necessarily have to use meshes that are adapted to local features, i.e. are small in the vicinity of regions with high dynamics while they are coarse in regions where the solution is smoother.

Codes that allow the use such adaptive meshes have been developed in the last decade in a number of fields, but have not found widespread use yet. The main reason is that this new generation of programs is more complex than the codes that have traditionally been used in the past, since they have to depart from the relatively simple paradigm of using ?IJK? meshes and explicit time stepping schemes. Programs implementing adaptive 3d meshes with implicit time stepping schemes therefore often have several 100,000 lines of code and use complex data structures. In addition, providing parallelization support is significantly more complex since implicit solvers require communication within each time step, and local refinement can only be made efficient if cells can be moved from processors with larger blocks to ones with smaller blocks.

Writing such codes therefore requires a much more concerted effort, and should be focused to general purpose libraries that support different applications built on top of them. Based on our past and present experience in building such libraries and geodynamics applications, we therefore propose to work in the following directions:

A framework for quasi-static modeling

We are presently working on an application simulating long-term plate deformation and motion by describing plate interaction using an elastoplastic and viscoplastic model. The purpose of this code is to study the growth and the interaction of tectonic faults. This code uses adaptive meshes [3, 4, 6] to resolve the small-scale features of faults while retaining coarse meshes away from them, and implicit solvers to avoid the stability problems of explicit methods when used with locally refined meshes. It contains a very general description of plasticity models that allows to quickly replace one model by another one in order to investigate the effects of different parameterizations on the output.

In its present version, our code is solving the quasi-static equations of elastoplasticity using an adaptive finite element scheme with implicit time stepping solvers [8, 9, 10, 11]. We propose to extend the application to a framework that is capable of solving different but similar equations or combinations of equations of quasi-static type. This, for example, includes solving for slow magma motion or coupling tectonics with equations describing erosion.

In all these cases, the overarching framework driving the time stepping scheme, when, where and how to refine the mesh [2, 5, 12, 13], when to generate graphical output, and most of the technology for linear and nonlinear solvers as well as the infrastructure for parallel computing can remain the same. Only the actual description of the equation to be solved, i.e. an implementation of the operators used in each linearized step need to change. By using a carefully written framework, the time needed to switch or extend an application by another equation can be significantly reduced.

Support libraries for geodynamics applications

The deal.II library is a general purpose library supporting the use of adaptive finite element methods for 1d, 2d, and 3d applications [1]. It has been used for a wide variety of problems, including glacier modeling, ground penetrating radar simulations, cancer imaging, wave propagation, computational fluid dynamics, and elasto-plastic rheology for simulations of continental scale deformations. It comes with interfaces to the PETSc linear algebra library [7] supporting distributed parallel computations on clusters of computers. Implicit solvers with up to 10,000,000 unknowns have been demonstrated using this coupling.

We propose to further integrate deal.II with other software packages that are used within the Computational Infrastructure in Geodynamics (CIG) initiative. As with all previous versions of deal.II, the resulting codes will be made available as Open Source to the general public and the members of CIG in particular.

In addition, we propose to offer the code implementing abovementioned quasi-static equation framework to the CIG initiative for use in other projects.

Achieving these goals

In order to achieve the goals laid out in this project, the following steps are required: Complete the existing program for elasto-plasticity and extend it into a general framework for quasi-static equations, including using automatically adapted hierarchical meshes using simple error indicators, and implicit time stepping schemes. This includes validating the solutions using analytically known solutions as well as typically used numerical benchmarks. It is expected that this step requires 6-8 months of full-time work.

Implementation of individual models within the framework, such as viscosity or magma migration. Given that the framework is already designed to allow adding more equations, this is intended to be a relatively straightforward step that we expect to require 2-3 months of full time work, including validation of results.

Based on the ability to solve general equations, we intend to develop realistic models of geologic dynamics, such as realistic rheologies for rock behavior in or close to faults. This also includes systematic testing of material property ranges to identify values of parameters matching observable features of the earth. Implementing these features, and systematic studies are expected to take 6-8 months of full-time work.

In parallel to all of these efforts, we intend to further improve the capabilities of the framework as well as of the deal.II library to handle large-scale computations. While the library is able to support distributed computations on parallel machines, it can be further optimized in this direction by parallelizing more operations and tuning linear solvers for less communication between processors. We anticipate that this concurrent effort will take 6 months of full-time work.

References

[1] W. Bangerth, R. Hartmann, and G. Kanschat. `deal.II` *Differential Equations Analysis Library, Technical Reference*, 2004. `http://www.dealii.org/`.

[2] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations.* Birkhäuser Verlag, 2003.

[3] G. F. Carey. *Computational Grids: Generation, Adaptation and Solution Strategies.* Taylor & Francis, 1997.

[4] G. F. Carey and J. T. Oden. *Finite Elements: Computational Aspects.* Prentice-Hall, 1984.

[5] P. A. Cundall. Numerical experiments on localization in frictional materials. *Ing. Arch.*, 58:148–159, 1989.

[6] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Adaptive Finite Element Methods.* North-Holland, 1996.

[7] PETSc: Portable, Extensible Toolkit for Scientific Computation. http://www-unix.mcs.anl.gov/petsc/petsc-2.

[8] R. Rannacher and F.-T. Suttmeier. A posteriori error control in finite element methods via duality techniques: Application to perfect plasticity. *Comp. Mech.*, 21(2):123–133, 1998.

[9] R. Rannacher and F.-T. Suttmeier. A posteriori error estimation and mesh adaptation in elasticity and elasto-plasticity. In P. Ladevèze and J. T. Oden, editors, *Advances in Adaptive Computational Methods in Mechanics*, volume 176, pages 275–292, Amsterdam, 1998. Elsevier.

[10] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity.* Springer, 1998.

[11] J. C. Simo and R. L. Taylor. Consistent tangent operators for rate-independent elastoplasticity. *Comp. Math. Appl. Mech. Engrg.*, 48:101–118, 1985.

[12] P. Wriggers and O. Scherf. An adaptive finite element algorithm for contact problems in plasticity. *Comp. Mech.*, 17:88–97, 1995.

[13] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Numer. Meth. Eng.*, 24:337–357, 1987.