2005 SCEC Progress Report: Deployment of a parallelized three-dimensional finite element code for modeling deformation in southern California

Charles A. Williams, Rensselaer Polytechnic Institute

Introduction

The goal of this project was the ongoing development of a quasi-static finite element code to model deformation in southern California. The first phase of this project (2003) was devoted primarily to an initial integration of the PI's version of the TECTON finite element code into the GeoFramework [www.geoframework.org], an extension of the Pyre simulation framework being developed at Caltech. During this first phase, there was also a considerable amount of code restructuring, including the removal of outdated f77 constructs, the transition to BLAS routines to handle linear algebra tasks, and modularization of the code. In the second phase of the project, many new code capabilities were added, and an alpha version of the code was made available for use by the SCEC modeling community. The main improvements were the addition of 9 new element types (including linear and quadratic tetrahedral elements), a new modular method of defining material models, and a complete reworking of code logic to increase efficiency and prepare for the task of code parallelization. In addition, the code was more thoroughly integrated into the Pyre framework, allowing the code to make use of additional features such as automatic unit conversion. The code also went through a couple of name changes during the second phase. The Pyre version of the code was originally named PyCrust to indicate a completely new code branch from the original TECTON code. Once it was discovered that this name was already in use by another software package, the name was changed to LithoMop, which is the name presently in use.

The third phase of this work is ongoing, and has been devoted primarily to improving the usability of the code by the SCEC modeling community. Several additional features have been added to the code, including a new method of dealing with gravity-induced stresses and improved output options for visualization. Additional work has involved an improved build procedure and an online installation guide and tutorial based on one of the CFEM benchmarks.

This SCEC project continues to leverage resources from other sources. An NSF ITR project continues to support much of the work of code parallelization as well as adding additional capabilities to the code. This project also continues to make use of resources from the Computational Infrastructure for Geodynamics (CIG) [www.geodynamics.org]. CIG has devoted significant resources to this project, including the services of Matt Knepley (one of the authors of the PETSc parallel computation library), who is assisting with code parallelization. CIG has also assisted with improving the build procedure for LithoMop and for the Pythia/Pyre simulation framework. As a result of discussions at the CFEM workshop in Los Alamos in July of 2005, I submitted two separate proposals to CIG that will directly benefit this project.

At present, an initial parallel version of the code is being tested, and an efficient implementation should be publicly available by early 2006. This implementation should have most of the features needed by the SCEC modeling community. Several modeling efforts are currently underway, and these will serve to test the code as well as highlight any additional improvements that need to be made. Finally, an effort is underway to merge LithoMop with Brad Aagaard's EqSim dynamic rupture propagation code. The resulting code, to be named PyLith, will be able to simulate earth behavior over a wide range of temporal and spatial scales.

Recent Developments and Ongoing Work

One problem that plagues numerical models of the earth is the sudden 'switching on' of gravity at the beginning of the solution (Figure 1). In a typical model this results in unrealistically large strains at the beginning of computations. The difficulty is in providing a reasonable initial state of stress without these large deformations, while ensuring that gravitational forces function normally for subsequent time steps. I have developed a method that accomplishes this by precomputing the gravitational stresses using user-specified values for Poisson's ratio. The resulting self-equilibrated set of stresses is used as an initial stress in all subsequent computations, while allowing body forces to be used as usual. Because the solution method uses a total strain formulation, all material models have been modified to account for possible initial stresses. This feature is particularly important for many of the problems that will be addressed by the SCEC modeling community. Many of the models of interest involve viscoelastic relaxation, and gravity must be included to obtain the correct response. This will also be important for models incorporating plastic behavior (used to represent brittle deformation on a macro scale), since the yield and flow functions are pressure-dependent. The new method of addressing this problem has been tested and appears to work well for geologic problems.

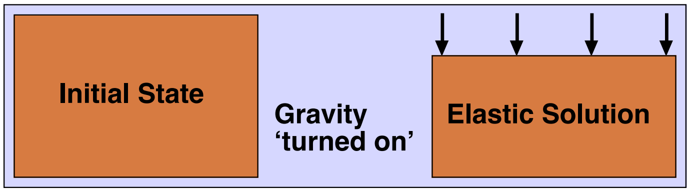

Figure 1. Difficulty encountered when simulating an earth model using gravity. In the real earth, lithospheric material is emplaced in the presence of gravity so that the system is already in equilibrium. In a numerical model, gravity is suddenly applied to the system, resulting in unrealistic strains.

Another difficulty encountered in numerical modeling is translating the output from a mesh generation package into something that the finite element code can use. This problem was discussed at the CFEM workshop in July, 2005, and is considered to be a priority for SCEC modeling. This difficulty is being addressed in two ways. First of all, this task will be simplified through the use of the new Sieve methodology presently being developed as part of PETSc [*Knepley and Karpeev, submitted*, 2005]. Sieve is a new method of representing computational meshes and is being used in the present version of LithoMop as a collaborative effort between Matt Knepley and myself. In addition to simplifying the task of code parallelization, Sieve may also be used as a convenient mechanism for coupling different mesh generation packages directly to LithoMop (or other finite element codes). Secondly, another CIG proposal recommends the development of just such a facility using Sieve (Figure 2). This package will be a generic facility that accomplishes the tasks of translation, partitioning, and uniform refinement, and should be usable by any finite element code, including LithoMop. The present version of the code uses Sieve to translate serial LithoMop input into parallel LithoMop input, but the code is now being redesigned to use Sieve data structures directly. Thus, all that will be needed is a set of modules to translate mesh generation format into Sieve format.

A problem related to getting information into the finite element code is the task of getting output useful for visualization. I have put in options for creating either ASCII or binary output in AVS UCD format. This is a fairly common format used for finite element visualization, and can be used directly by packages such as AVS, Iris Explorer, and ParaView. The output sections have also been redesigned to pave the way for output modules that are likely to be designed as general facilities for CIG. A particularly useful example would be a module using HDF5, which would have the advantages of being platform-neutral and self-describing.

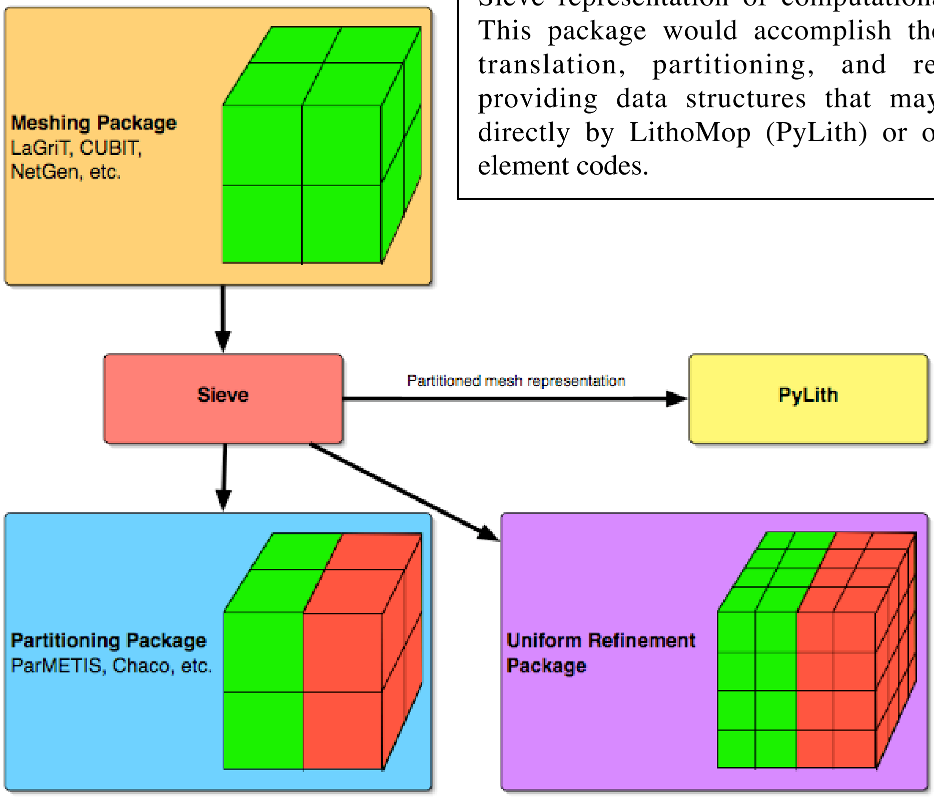

Figure 2. Proposed package making use of the Sieve representation of computational meshes. This package would accomplish the tasks of translation, partitioning, and refinement, providing data structures that may be used directly by LithoMop (PyLith) or other finite element codes.

A problem related to visualization is the ability to store and query simulation results. This will require the ability to store metadata describing the simulation as well as storage space for the results themselves. To be useful, it will also be necessary to query the results for particular temporal and spatial locations, ideally using the same interpolation methods that were used in the original computations. This facility will be useful both in comparing benchmark results as well as providing a means of querying model results to compare with observations. Another proposal has been submitted to CIG to address these needs.

Another issue that was brought to light at the CFEM workshop was the difficulty in building and setting up LithoMop and the supporting codes for different platforms. To address this issue, I have worked with Leif Strand at CIG and we have successfully modified both Pythia and LithoMop to use the Gnu build procedure (autoconf, automake, etc.). This dramatically simplifies the build procedure for users, while at the same time moving the burden of porting issues from the Pythia/LithoMop code developers to the open-source Gnu community.

Code Demonstration and Online Tutorial

At the CFEM workshop in July of this year I gave a presentation on LithoMop. As part of the workshop I also assisted several workshop participants in installing the code on their laptops, and walked the workshop participants through an online tutorial based on SCEC BM 5. At www.geodynamics.org users can find tarballs for code installation, installation instructions, and the online tutorial. The tutorial (Figure 3) walks users through the complete process of generating a mesh using the public-domain NetGen mesh generation package [www.hpfem.jku.at/netgen], setting up and running SCEC BM 5, and visualizing the results using the ParaView public-domain visualization package [www.paraview.org]. This is a useful problem for demonstrations since it involves both faults and relatively complex boundary conditions. I am presently in the process of putting together additional tutorials using different meshing packages such as LaGriT [lagrit.lanl.gov] and other SCEC benchmarks. Eventually, the site will also include methods for users to store and compare benchmark results using the requested archival/querying facility from CIG.

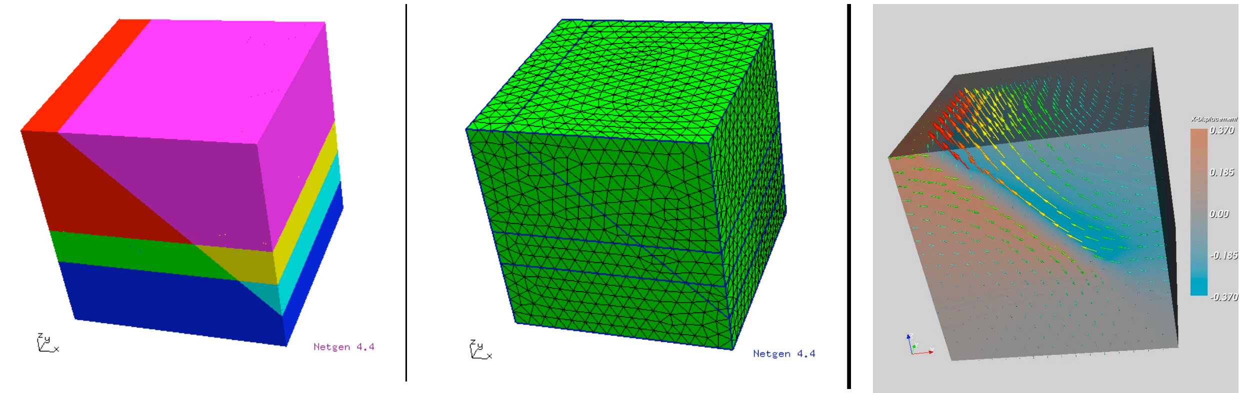

Figure 3. Some of the steps covered in the online tutorial based on SCEC BM 5. Users first define a geometry that may be used for NetGen (left), and create a mesh (center). After setting up and running the problem, the results may be viewed using a visualization package such as ParaView (right).

Another topic covered at the CFEM workshop was the development roadmap for LithoMop (Figure 4). In addition to covering the development history of the code, current and future plans were shown. Many of the plans coincide closely with SCEC priorities. Some of the important priorities (supported primarily by CIG and an ITR grant) are the use of the Pyre Simulation Controller (important for inversions), use of a spatial database for specification of material properties and boundary conditions (greatly simplifying specification of these parameters), and a new and more robust method of specifying faults using cohesive elements. One of the most significant features, though, is the plan to merge LithiMop with Brad Aagaard's EqSim dynamic rupture propagation code. This merged code, which will be named PyLith, will address many of the present and future needs of the SCEC modeling community. This code will be able to address the long-term buildup of stresses on interseismic or multiple earthquake cycle time scales, while simultaneously representing the extremely short time scales involved in seismic rupture. As an example of a current SCEC modeling priority, this merged code would allow us to simulate the slow buildup of stresses leading to the Landers earthquake sequence. The earthquake sequence could then be represented realistically using the dynamic capabilities of the code. The resulting slip distribution could then be used to investigate possible stress triggering of the Hector Mine sequence, allowing us to simultaneously investigate the short-term and long-term properties of the fault and of the crust in this region.

Presentation and Dissemination of Results

The results of this project have been presented at several meetings, in addition to the CFEM workshop mentioned previously. A poster was presented at the AGU Fall meeting last year [*Williams et al.*, 2004], and another will be presented this year [*Williams et al.*, 2005a]. A poster was also presented at the SCEC annual meeting [*Williams et al.*, 2005b]. As mentioned previously, the present version of the code, as well as installation instructions and the online tutorial are available at the CIG website [www.geodynamics.org]. Much more assistance should be forthcoming from CIG, and will also be accessible from this site. This includes archival of simulation results along with the proposed querying facility.

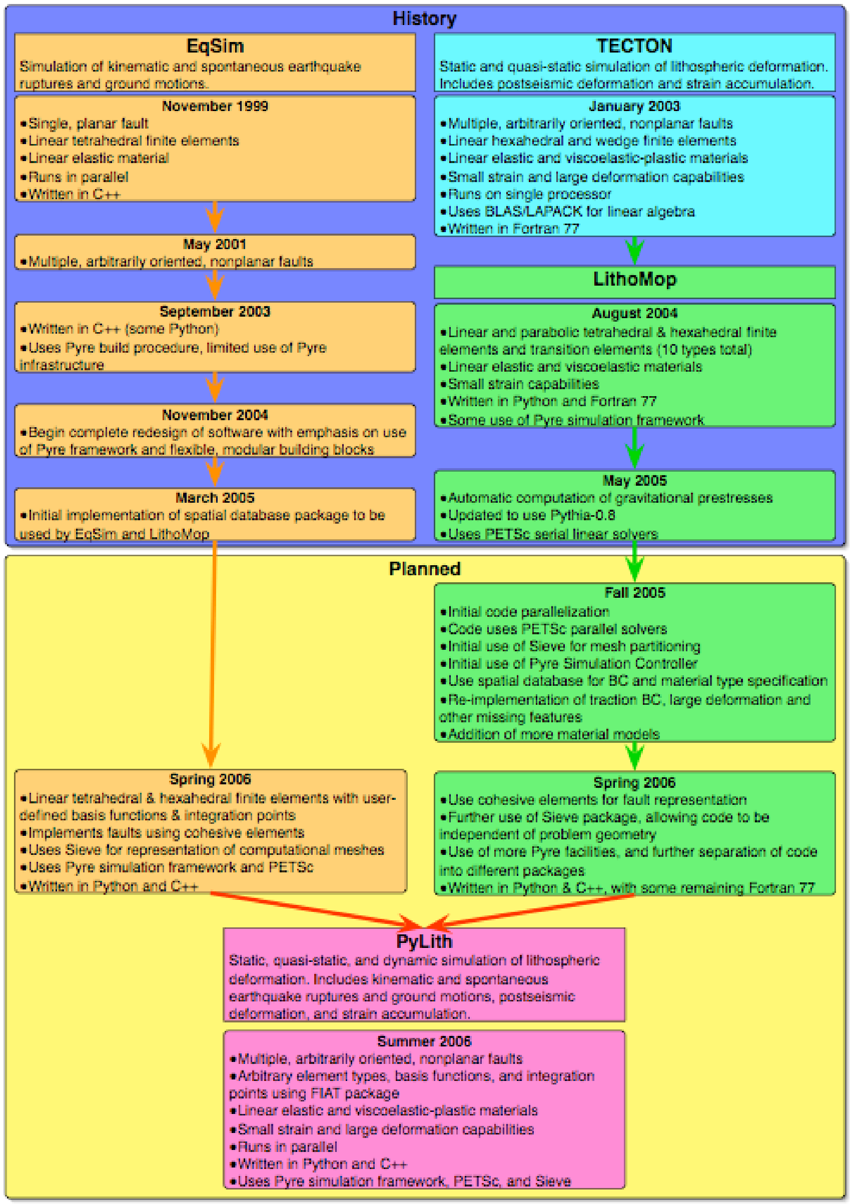
History
EqSim
Simulation of kinematic and spontaneous earthquake ruptures and ground motions.
November 1999
•Single, planar fault
•Linear tetrahedral finite elements
•Linear elastic material
•Runs in parallel
•Written in C++
May 2001
•Multiple, arbitrarily oriented, nonplanar faults
September 2003
•Written in C++ (some Python)
•Uses Pyre build procedure, limited use of Pyre infrastructure
November 2004
•Begin complete redesign of software with emphasis on use of Pyre framework and flexible, modular building blocks
March 2005
•Initial implementation of spatial database package to be used by EqSim and LithoMop
TECTON
Static and quasi-static simulation of lithospheric deformation. Includes postseismic deformation and strain accumulation.
January 2003
•Multiple, arbitrarily oriented, nonplanar faults
•Linear hexahedral and wedge finite elements
•Linear elastic and viscoelastic-plastic materials
•Small strain and large deformation capabilities
•Runs on single processor
•Uses BLAS/LAPACK for linear algebra
•Written in Fortran 77
LithoMop
August 2004
•Linear and parabolic tetrahedral & hexahedral finite elements and transition elements (10 types total)
•Linear elastic and viscoelastic materials
•Small strain capabilities
•Written in Python and Fortran 77
•Some use of Pyre simulation framework
May 2005
•Automatic computation of gravitational prestresses
•Updated to use Pythia-0.8
•Uses PETSc serial linear solvers
Planned
Fall 2005
•Initial code parallelization
•Code uses PETSc parallel solvers
•Initial use of Sieve for mesh partitioning
•Initial use of Pyre Simulation Controller
•Use spatial database for BC and material type specification
•Re-implementation of traction BC, large deformation and other missing features
•Addition of more material models
Spring 2006
•Linear tetrahedral & hexahedral finite elements with user-defined basis functions & integration points
•Implements faults using cohesive elements
•Uses Sieve for representation of computational meshes
•Uses Pyre simulation framework and PETSc
•Written in Python and C++
Spring 2006
•Use cohesive elements for fault representation
•Further use of Sieve package, allowing code to be independent of problem geometry
•Use of more Pyre facilities, and further separation of code into different packages
•Written in Python & C++, with some remaining Fortran 77
PyLith
Static, quasi-static, and dynamic simulation of lithospheric deformation. Includes kinematic and spontaneous earthquake ruptures and ground motions, postseismic deformation, and strain accumulation.
Summer 2006
•Multiple, arbitrarily oriented, nonplanar faults
•Arbitrary element types, basis functions, and integration points using FIAT package
•Linear elastic and viscoelastic-plastic materials
•Small strain and large deformation capabilities
•Runs in parallel
•Written in Python and C++
•Uses Pyre simulation framework, PETSc, and Sieve

Figure 4. Timeline and development roadmap for the EqSim and PyLith finite element codes, which will merge to become PyLith. This graphic and additional information are available from www.geodynamics.org.

References

Knepley, M. G. and D. A. Karpeev, Flexible representation of computational meshes, *ACM Trans. Math. Software*, submitted, 2005.

Williams, C. A., B. Aagaard, M. Aivazis, and M. Simons, Addressing multi-scale problems in geophysics with coupled modeling tools: Current development status of the lithomop deformation modeling code within the Pyre framework, *AGU Fall Meeting*, 2004.

Williams, C. A., B. Aagaard, and M. Knepley, Development of software for studying earthquakes across multiple spatial and temporal scales by coupling quasi-static and dynamic simulations, AGU Fall Meeting, 2005a.

Williams, C. A., B. Aagaard, and M. Knepley, Development roadmap for the PyLith/LithoMop/EqSim finite element code for fault-related modeling in southern California, *SCEC Annual Meeting*, 2005b.