GALE: CIG's Arbitrary Lagrangian Eulerian Code

4th November 2005

This is a proposal for what the Arbitrary Lagrangian Eulerian (ALE) code developed by CIG for the long term crustal dynamics community will look like. This effort is in direct response to the recommendations coming from the NSF workshop on Tectonic Modelling[1]. The preliminary name for the code is **GALE** (**G**eodynamics **ALE**).

The goal is to develop an open source code that is at least as useful as SOPALE and MicroFEM, with the addition of 3D capability. In particular, GALE should be able to run all of the benchmarks described in the attached GeoMOD 2004 poster, as well as a traditional subduction model as shown in Fig 1.

We do not have a copy of either SOPALE or MicroFEM, so we have been driven to infer what the codes do through related publications. Our inferences may be wrong, so we invite corrections. Our primary source is a paper by Fullsack (Geophysics J. Int., 1995, **120**, 1-23) which describes MicroFEM. It has a flowchart describing the code, which is reproduced on the left side of Fig 2.

We have to decided to base our efforts on St Germain, StgFEM, PiCellerator, and Underworld. These are proven, capable, open source FEM libraries written by the Victorian Partnership for Advanced Computing (VPAC) and Louis Moresi's group at Monash University. CIG has a good working relationship with VPAC and Louis Moresi, and members of CIG have collaborated with them in the past.

In Fig 2, the arrows point to where those parts are already mostly or completely implemented. Everything else is thrown into GALE, even though some of the work will certainly involve improving some of the underlying libraries. Going through the steps in the algorithm, we can see where most of the parts already exist, and where parts have yet to be implemented.

1. Compute and solve finite element system on Eulerian grid

   This requires a basic FEM setup (St Germain) but also the ability to interpolate between an Eulerian and Lagrangian mesh (PiCellerator). The

[1] `http://www.geodynamics.org:8080/cig/workinggroups/crust/workarea/report`

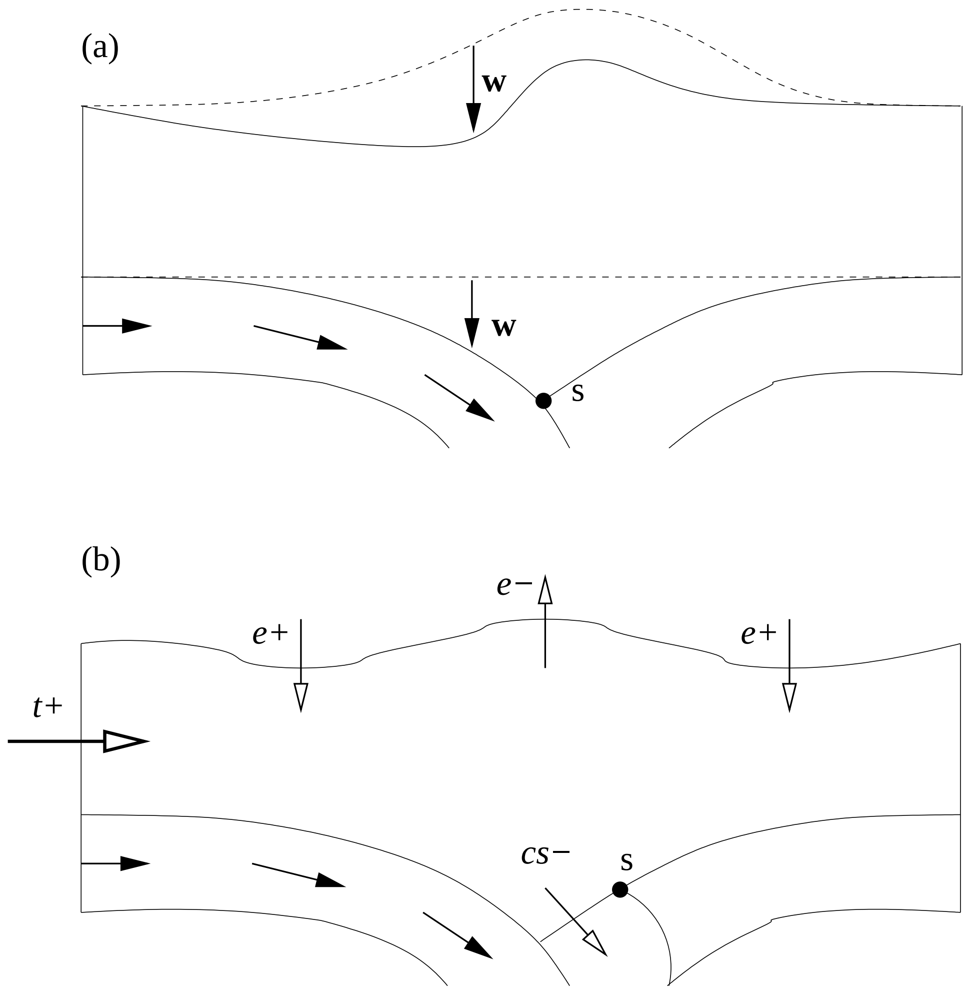

Figure 1: General subduction model from Fullsack (1995). The basic subduction model can be modified by introducing isostasy and/or boundary mass fluxes. (a) Flexural or local isostasy. Every Eulerian column is displaced by the corresponding local or flexural deflection $\mathbf{w}$ of the base. (b) Mass fluxes crossing the boundaries may be: (1) the tectonic flux t^+ into the domain, due to tectonic convergence; (2) the deposition flux e^+ onto the domain due to surface processes; (3) the deposition flux e^- from the domain, due to surface processes; (4) the crustal subduction flux cs^- from the domain, due to the entrainment of crustal material from the subducting plate.

Figure 2: Mapping between MicroFEM and GALE

Eulerian grid is logically regular, but it can distort to match the upper boundary as it changes (see Fig 1). Moving the grids is relatively complicated (covered in step 3), but solving the finite element system with a stretched grid is not. Implementing the boundary conditions for solving the finite element system in Fig 1 is straightforward.

We will also use rheologies from Underworld as a basis for realistic computations. Implementing this part does not involve significant work.

2. Apply surface processes to advect topography

   Because surface processes are still an area of research, GALE will not provide a complete solution for this process. Rather, GALE will provide a simple example solution and supply an API to enable users to provide their own routines.

   This will require tracking the surface and moving particles. This is new functionality and will require some work.

3. Compute net vertical Eulerian motion of Eulerian boundaries

   The ability to conform the domain of computation to the surface is the distinguishing aspect of the MicroFEM and SOPALE codes. It is also the trickiest to implement. It has two components:

   (a) Compute and track the movement of the boundary

       The boundary is defined by the Lagrangian particles, so we have to compute contours to even find the boundary. Once the boundary is known, computing the movement is the subject of later steps in the algorithm.

   (b) Perform all of the needed interpolations to scale the boundary

       When moving the Eulerian grid, MicroFEM only computes the movement of the top of the grid and scales the rest of the grid linearly. Emulating this is mostly making sure that all of the quantities are correctly interpolated. This is new work that will likely go in as an enhancement of St Germain itself.

4. Compute vertical flexural/local isostasy

   This part is just calculating the vertical displacement $\mathbf{w}$ from vertical flexing of the crust due to changes in the weight on the crust. This is essentially a part of the boundary condition for the bottom of the simulation. In a 2D code for the problem shown in Fig 1, this requires solving a pair of 1D finite element beam problems. In the 3D code, this requires solving a 2D finite element beam problem. This is a specialized boundary condition, so it will have to be implemented.

5. **v** is corrected to **v**=**v**+**w**

   This is a trivial step.

6. Interpolate **v** to get motion of Eulerian grid

   This simply requires the appropriate interpolation functions, many of which are already present in Underworld and PiCellerator.

7. Interpolate **v** on Lagrangian nodes

8. Regrid Lagrangian fields on Eulerian grid

   The necessary functionality for these two steps is already implemented in PiCellerator.

With all that in mind, we have developed the following schedule.

Nov 20

- Write Rough Proposal, integrating the goals in the context of Fullsack (1995) (WL, MG, SQ)
- Develop XML Specs for GeoMOD plastic bench marks
- Set up a Plone page
- Establish repository with empty parts (WL, SQ)
- Demonstrate an extension model
- Complete the initial implementation of a moving upper boundary of the Eulerian mesh

Dec 15

- Iterate on proposal by seeking input from community (WL)

Jan 15

- Demonstrate SOPALE workflow using GALE primarily based on StGermain components (WL, LH)
- Parts will not be fully working

Apr 15

- Demonstrate complete simple benchmarks

Oct 15

- Run the complete SOPALE integration benchmark
- Release GALE 1.0

WL: Walter Landry; LH: Luke Hodkinson; SQ: Steve Quennette; MG: Mike Gurnis