Gamr: A Free, Parallel Adaptive Code for Tectonics and Mantle Convection

Walter Landry
Computational Infrastructure for Geodynamics

Summary

Computational Infrastructure for Geodynamics (CIG) is an NSF funded, community governed center that develops, maintains, and distributes high performance parallel software for geophysics. In response to requests from the community, CIG is now starting development of Gamr, a new Adaptive Mesh Refinement (AMR) code for Tectonics and Mantle Convection. This is a tentative plan on how we will accomplish this.

Basic Equations

Many processes in Tectonics and Mantle Convection are well modeled by the equations for Stokes flow and advective-diffusive heat flow.

$$\tau_{ij,j} + p_{,i} = f_i$$

$$\nabla \cdot v = 0$$

$$\frac{\partial T}{\partial t} + v \cdot \nabla T = \kappa \nabla^2 T + Q$$

where, for a simple linear Newtonian fluid

$$\tau_{ij} \equiv \eta (v_{i,j} + v_{j,i})$$

Realistic materials are usually non-linear, incorporating plasticity, elasticity, and temperature dependent viscosity. In addition, there may just be different materials, leading to strong variations in viscosity and strain rate across small regions. This can lead to very fine structures such as faults and plumes seen in Figure 1.

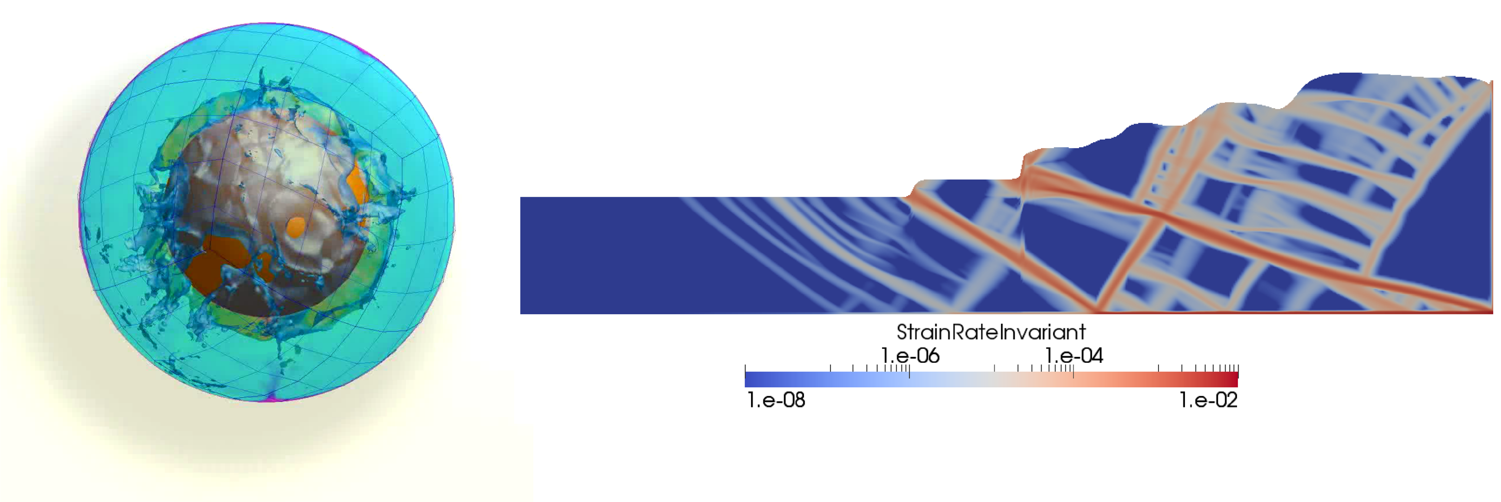

Figure 1: Fine structures are evident in these mantle convection and sandbox simulations. These simulations were produced by CitcomS and Gale, two codes available from CIG.

AMR Fundamentals

The idea of AMR is to have multiple grids with different resolutions. There is a coarse grid which covers the whole simulation, and then there are finer grids which only cover select regions as in Figure 2.

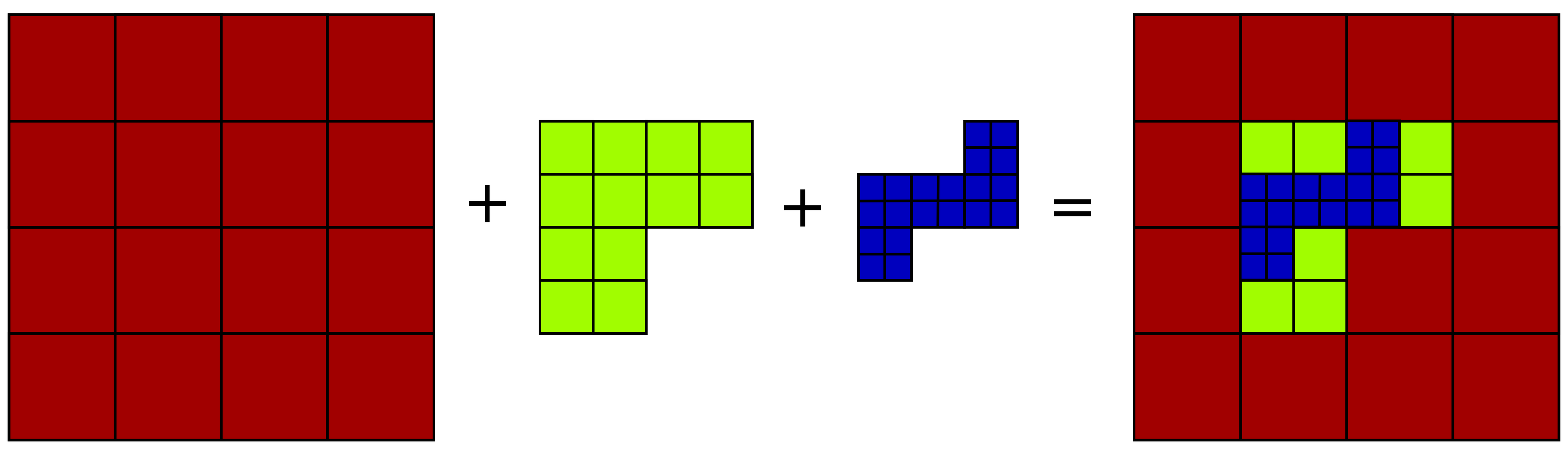

Figure 2: Multiple grids in an AMR simulation.

Solving Stokes in AMR

At the beginning, we use Chombo to solve for the Stokes flow on all of the grids simultaneously using a staggered finite difference multigrid solver. Traditional multigrid has each grid cover the whole domain, but we use the existing fine and coarse grids. This means that we need boundary conditions for these finer levels. Most AMR solvers for other equations apply Dirichlet conditions for all variables, but then the Stokes equations will become singular. So we apply Dirichlet conditions for the normal velocities, and Neumann conditions for the tangential velocities.

Interpolating boundary conditions from the coarse level requires some care, since a naive choice will ruin the accurracy. In order to maintain second order accuracy, boundary conditions interpolated from the coarse levels need to be of quadratic accuracy or better. Figure 3 shows stencils for v_x for the x boundary (Dirichlet) and y boundary (Neumann). Of note is that the interpolation can use points from the fine grid.

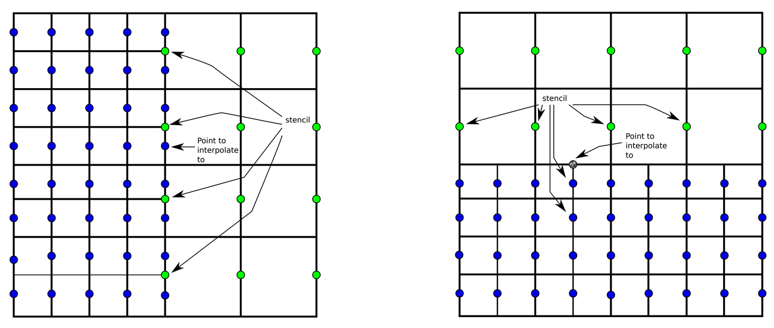

Figure 3: Interpolation stencils for v_x at the x and y boundaries.

Subcycling in Time

In general, to get a stable solution on the fine grid you have to take small time steps. However, the coarse grids do not require such small time steps. An important component of the improved efficiency of traditional AMR is that the coarse grid does not need to take as many steps. However, all of the existing efforts to apply AMR to solid earth geophysics (ALOES, DOUAR, RHEA) use a uniform time step for both fine and coarse grids.

We implement subcycling by first computing a solution on all of the grids at once. Then we march the coarse grids forward one, big coarse step. We compute a solution with just the coarse grid. We can then interpolate between the two coarse timesteps to give boundary conditions for the fine time steps.

Current Status

Development has just started, but we already have a functional prototype that solves the 2D, isoviscous Stokes equations in parallel. Refinement is implemented (see Figure 4), and, in the coming months, we plan to implement the interpolation operators needed for adaptivity and time stepping. After that, we will add variable viscosity, material tracking, plasticity, elasticity, deformable boundaries, and non-rectangular geometries (e.g. the sphere).

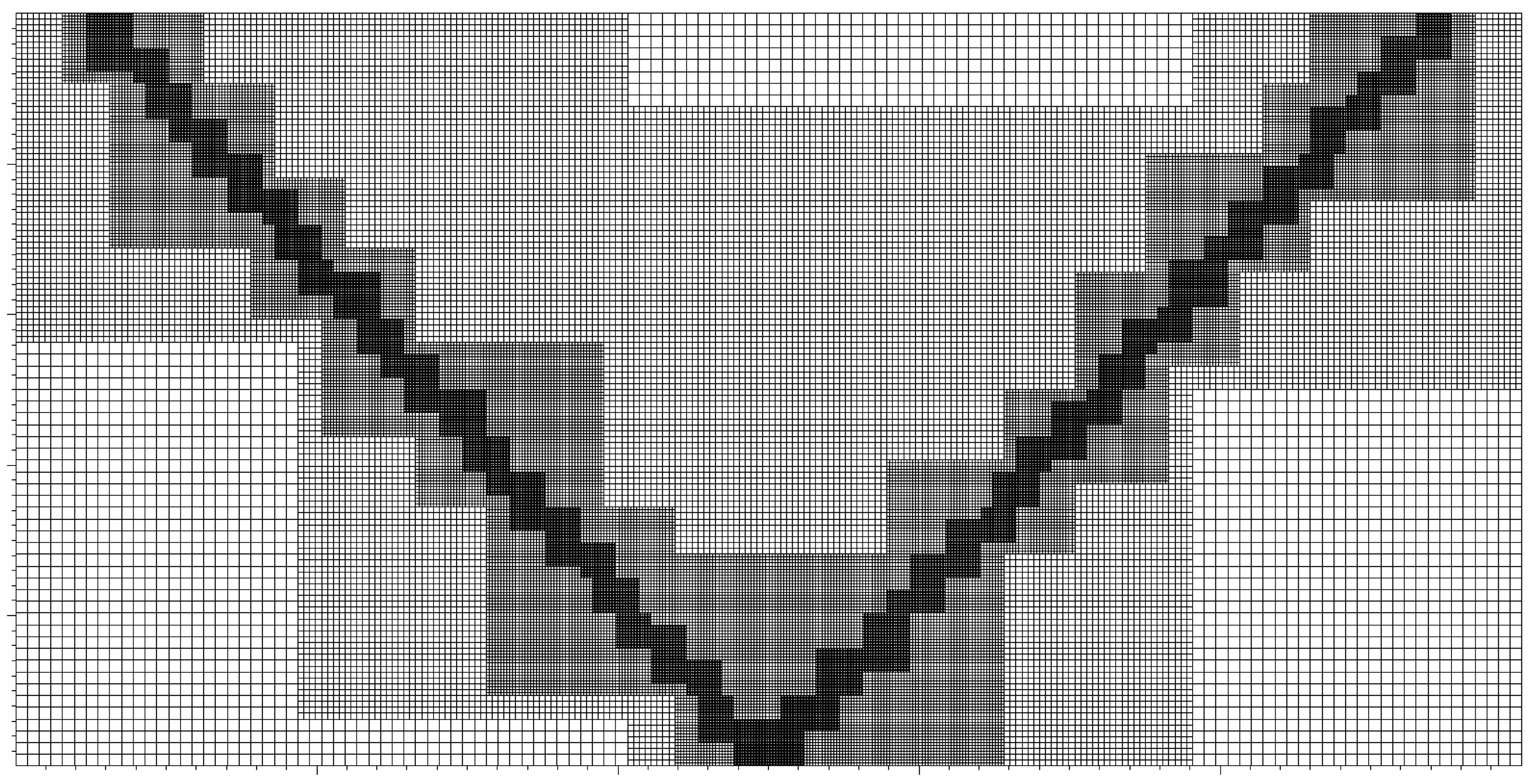

Figure 4: A Gamr model refined along a manually inserted fault