Geophysical Research Abstracts
Vol. 12, EGU2010-10224, 2010
EGU General Assembly 2010

Better Strategies for Finite Element Solutions of Variable Viscosity Stokes Flow

Jörg Hasenclever (1), Jason Phipps Morgan (2), and Chao Shi (2)

(1) Institute of Geophysics, Hamburg University, Hamburg, Germany (joerg.hasenclever@zmaw.de), (2) EAS Department, Cornell University, Ithaca, USA (jp369@cornell.edu, cs387@cornell.edu)

Accurate numerical solution of variable viscosity Stokes Flow is one of the most important issues for better geodynamic understanding of mantle convection and mantle melting. While a good Stokes solver is usually an integral part of a good Navier-Stokes solver, typically Navier-Stokes equations are solved for flow of a fluid with uniform viscosity. The lumped-mass-matrix is an excellent and cheap preconditioner for uniform viscosity Stokes flow (cf. Maday and Patera, 1989), therefore for most applications to Navier-Stokes flow the 'Stokes' part of the problem is viewed as well-resolved. Unfortunately, the inverse-viscosity-scaled lumped mass matrix does not work nearly as well to precondition Stokes flow in a fluid with strongly varying viscosity. This issue is already central to accurate numerical studies of convection in Earth's silicate-fluid mantle (May and Moresi, 2008; van Geenen et al., 2009; Burstedde et al., 2009) and may become central for researchers investigating Navier-Stokes problems with lateral variations in viscosity. Here we discuss several known computational hurdles to progress, and suggest strategies that offer promise in overcoming them.

The choices for solving the discrete pressure equation arising from Stokes flow typically involve several tradeoffs between speed and storage requirements. In exact math, the discrete pressure matrix S is symmetric, so that it should be possible to use a symmetric preconditioned conjugate gradient (CG) Krylov algorithm instead of needing an asymmetric GMRES (cf. Saad, 2003) or GCR (Generalized Conjugate Residual, cf. Van der Vorst, 2003) that would require $\sim$10-50 times more storage of past search directions. However, a CG-like method requires that the action of both S and any pressure preconditioner must be almost perfectly symmetric. This means that we must be very careful about the effects of roundoff in any iterative solver-based pressure preconditioner that may introduce numerically asymmetric operators at the level of their solution tolerance. We have found that use of either a Polak-Ribiere (Golub and Ye, 2000) or a Flexible CG (Notay, 2000) formulation for updating the conjugate gradient search directions allows us to use an inexact CG iterative solver with controllable tolerance requirements as an outer preconditioner, whereas the standard Fletcher-Reeves CG formulation typically leads to much slower or even no convergence.

Another ingredient that we have found to be effective is to include a coarse-grid preconditioner. Our preferred recipe currently has the following ingredients:
(1) Use accurate and efficient LBB-stable elements that lead to good convergence in iterative solution schemes (Silvester and Thatcher, 1986; Pelletier et al., 1989). We currently recommend Cornell Macroelements, as they are LBB-stable, are as easy to code as Taylor-Hood elements and have the same convergence properties as them, yet are much more locally incompressible and have no known incompressibility-related artifacts. In contrast, Taylor-Hood elements are 'too squishy' and also have a potential numerical artifact related to capturing an accurate hydrostatic pressure solution (Pelletier et al., 1989).
(2) Use an inexact CG pressure preconditioner to drastically reduce the necessary tolerance for the inverse-velocity calculation within it. This leads to a work-reduction of $\sim$2-3-fold.
(3) Use a coarse-grid preconditioner as part of a Dryja-Widlund Additive Schwartz (DWAS) multilevel preconditioner. The key point is to never compute any additional global fine-grid pressure-matrix operations except for the single operation needed to calculate each new CG search direction (i.e. avoid smoothing operations). We are currently exploring an apparently promising >2 level cascade-type improvement of this general preconditioning strategy.