Construction and Comparison of Uzawa Solvers

Dave May + Louis Moresi,
Monash University, Australia.

Overview

- Compressible formulation
- Key component of constructing Uzawa solvers

Some preliminary results of different Uzawa

The Continuum Problem

Momentum equation. (1)

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

eg. Stokes Flow (isotropic viscosity), but we will not restrict ourselves to one rheological model.

Conservation of mass. (2)

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0$$

Energy equation...

We only consider the flow equations here.

Compressible Formulation

(Jarvis & McKenzie, 1980)

Horizontal density gradients negligible.

$$\frac{\partial u_i}{\partial x_i} + \beta w = 0$$

where $$\beta = \frac{1}{\rho}\frac{\partial \rho}{\partial z}$$

The Discrete Problem

Std. Mixed Finite Element formulation converts (1) & (2) into

$$\begin{pmatrix} \boldsymbol{K} & \boldsymbol{G} \\ \boldsymbol{D'} & \boldsymbol{M} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{q} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{h} \end{pmatrix}$$

Let $\boldsymbol{G} = (\boldsymbol{G}_x, \boldsymbol{G}_y, \boldsymbol{G}_z)^T$

Incompressible: $\boldsymbol{D'} = \boldsymbol{G}^T$

Compressible: $\boldsymbol{D'} = \boldsymbol{G}^T + \boldsymbol{C}$

Solver Requirements

- Largely insensitive to rheology.
- Amenable to parallel computing.
- Robust + Efficient + Scalable + Fast.

Solution Strategies

a) De-couple {u} and {q}. Solve the pressure Schur complment system.

$$\left(D'K^{-1}G - M\right)q = D'K^{-1}f - h$$
$$\widehat{K}q = \widehat{f}.$$

b) Solve the momentum equation.

Proven to be robust for wide variety of rheologies.

PCG-Uzawa (I)

Preconditioned Conjugate Gradient

Two loop algorithm. Outer loop appears like a standard cg algorithm. Inner loop involves solving a system like;

$$Kv^* = Gs_k$$

PCG-Uzawa (II)

Demonstrations of the scalability, efficiency an
rheological flexibility include the convection
codes of Moresi, Zhong, Baumgardner.

Only applicable for symmetric systems. Thus
the compressible formulation presents a probler

Non-symmetric Uzawa

In practise we can construct a suitable Uzawa solver (ie the outer loop) for the compressible model from any iterative solver which handles non-symmetric matrices. (ie. cgs, bicg, etc)

The key ingredient is that all these solvers require the computation of a matrix-vector product.

So each new Uzawa solver requires us to replace

$$Ax \qquad \text{with} \qquad \widehat{K}q$$

which we evaluate via the following procedure

1) Solve $Kv^* = Gq$

2) $\widehat{K}q \sim D'v^* - Mq$

In general this simple substitution motivates developers to re-write the ENTIRE iterative solv into an Uzawa solver.

A better approach is to note that the fundamen operation of all iterative solvers is the mat-ve product - and the write software which ABSTRA this operation.

PETSc (i)

(Portable Extensible Toolkit for Scientific computation - see M

- The concept of an abstract Mat-Vec product is provided in the PETSc library via the matrix type (or class) called MatShell
- MatShell is an abstract matrix class. Thus, in itself the MatShell does not know how to perform any matrix operations - but it provides the application programmer with the ability to define "their own matrix operations"

PETSc (ii)

- What does this provide?
- Access to a vast number of existing iterative iterative solvers which can now (through the abstract Mat class) form as outer iterations for an Uzawa solver. This is all without having to re-writing any iterative solvers.
- Enables users at the command line to try different Uzawa solvers. Thereby enabling developers to choose the most efficient solver for their purposes.

Preconditioners

- We need to define our own preconditioners.
- Here used the simple pc

$$\widehat{\boldsymbol{K}} \sim \boldsymbol{Q}_{\hat{K}} = \mathrm{diag}\left(\boldsymbol{D}'\mathrm{diag}\left(\boldsymbol{K}\right)^{-1}\boldsymbol{G} - \boldsymbol{M}\right)$$

2D Test Problems

1) Isoviscous, boundary driven flow, 2x1 box,

$$u = \sin(x), \quad y = 1$$

2) Laterally varying viscosity, 1x1 box, (Zhong, 1996)

$$\eta = \exp(\alpha x)$$

Incompressible (symmetric) problems we use as outer Uzawa iterations;

cg, cgs, bicg, bcgs, bcgsl, minres, gmres

Compressible (non-symmetric) problems we use as outer Uzawa iterations;

cgs, bicg, bcgs, bcgsl, gmres

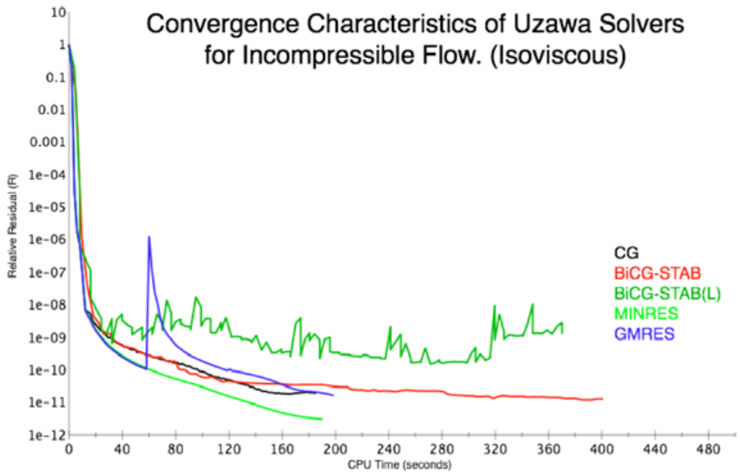
Notes

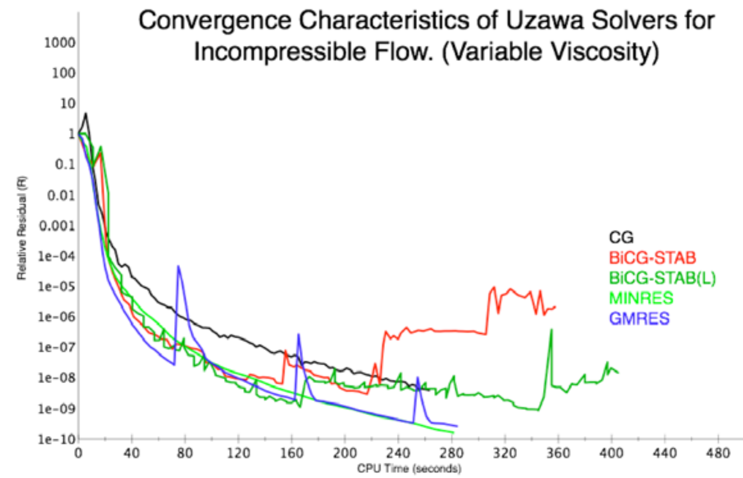
- The residual is given by; $$R = \frac{\|\widehat{\boldsymbol{f}} - \widehat{\boldsymbol{K}}\boldsymbol{q}\|}{\|\widehat{\boldsymbol{f}}\|}$$
- Test resolution 256 x 128 (isoviscous), 128x128 (variable viscosity)
- Inner solves performed using cg with ilu. Used standard PETSc stopping conditions.

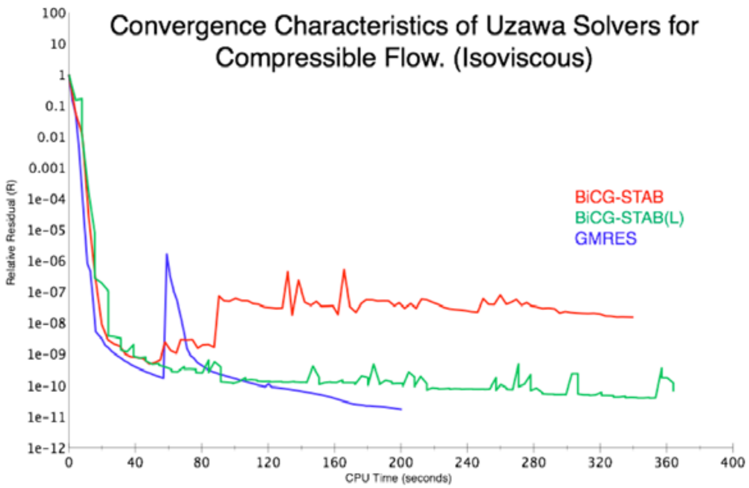
Observations

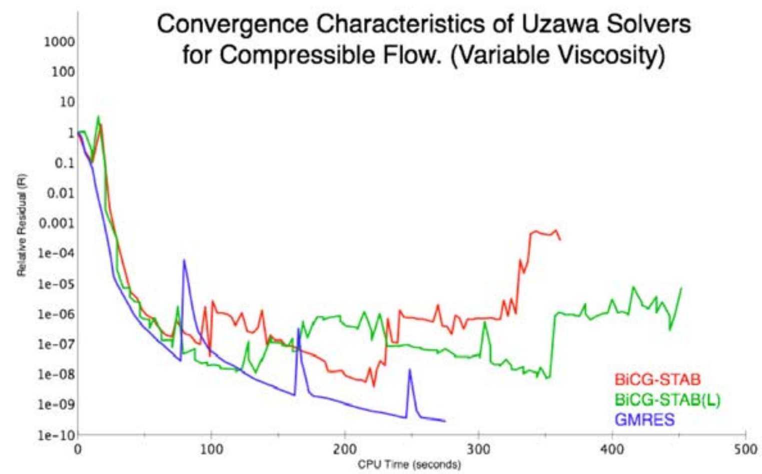
- All Uzawa schemes are insensitive to mesh resolution.
- Preconditioner is effective wrt to viscosity but not wrt the compressibilty term.

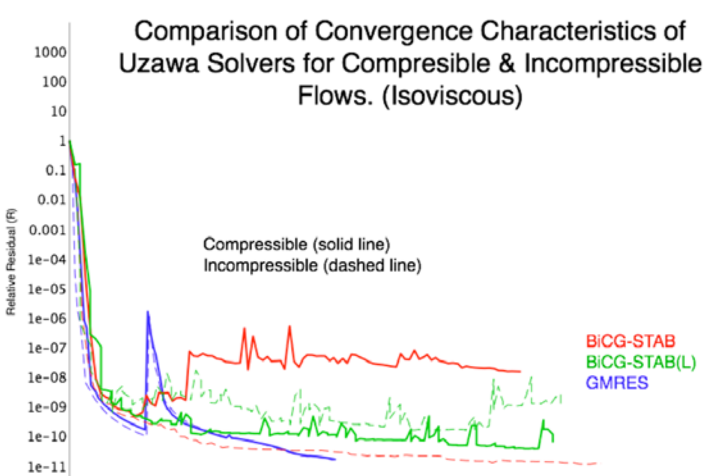
$$10^3 < \eta < 10^7, \qquad 1 < \beta < 100$$

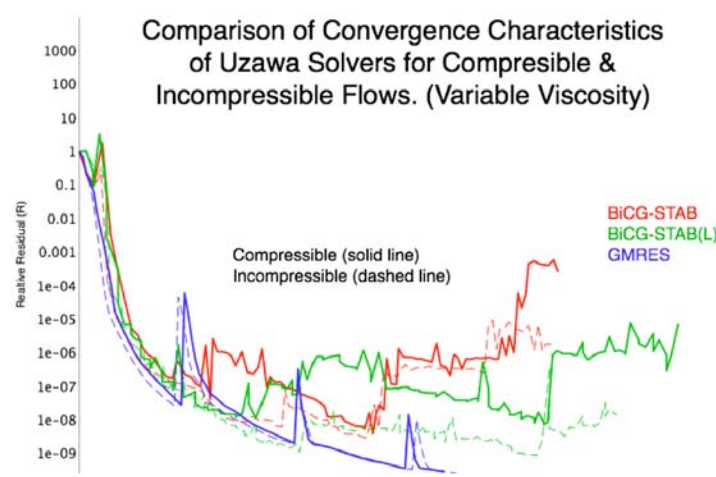

Convergence Characteristics of Uzawa Solvers for Incompressible Flow. (Isoviscous)
CG
BiCG-STAB
BiCG-STAB(L)
MINRES
GMRES
Relative Residual (R)
CPU Time (seconds)


Convergence Characteristics of Uzawa Solvers for Incompressible Flow. (Variable Viscosity)
CG
BiCG-STAB
BiCG-STAB(L)
MINRES
GMRES
Relative Residual (R)
CPU Time (seconds)


Convergence Characteristics of Uzawa Solvers for Compressible Flow. (Isoviscous)
BiCG-STAB
BiCG-STAB(L)
GMRES
Relative Residual (R)
CPU Time (seconds)


Convergence Characteristics of Uzawa Solvers for Compressible Flow. (Variable Viscosity)
BiCG-STAB
BiCG-STAB(L)
GMRES
Relative Residual (R)
CPU Time (seconds)


Comparison of Convergence Characteristics of Uzawa Solvers for Compresible & Incompressible Flows. (Isoviscous)
Compressible (solid line)
Incompressible (dashed line)
BiCG-STAB
BiCG-STAB(L)
GMRES
Relative Residual (R)


Comparison of Convergence Characteristics of Uzawa Solvers for Compresible & Incompressible Flows. (Variable Viscosity)
Compressible (solid line)
Incompressible (dashed line)
BiCG-STAB
BiCG-STAB(L)
GMRES
Reative Residual (R)