Experiences using multigrid for geothermal simulation

D.P. Bullivant, M.J. O'Sullivan
Department of Engineering Science
University of Auckland
New Zealand

Z. Yang
Centre for Petroleum Engineering
University of New South Wales
Australia

Abstract

Experiences of applying multigrid to the calculation of natural states for geothermal simulations are discussed. The modelling of natural states was chosen for this study because they can take a long time to compute and the computation is often dominated by the development of phase change boundaries that take up a small region in the simulation. For the first part of this work a modified version of TOUGH was used for 2-D vertical problems. A "test-bed" program is now being used to investigate some of the problems encountered with implementing multigrid. This is ongoing work. To date, there have been some encouraging but not startling results.

1 Introduction

This work is motivated by the desire to speed up the calculation of natural states in geothermal reservoir simulation. It was noticed that during these calculations, the time step size is limited by phase changes in just a few blocks. This observation suggests that improvements could be made if the computational effort was directed towards these few blocks. Yang [5] looked at a number of ways for doing this, one of which was multigrid.

A brief introduction to multigrid is given in the following section. In the third section the feasibility study of multigrid for geothermal simulation, carried out by Yang [5], is summarized. The fourth section describes the work in progress to solve some of the problems identified by the feasibility study. The final section is a summary.

2 What is multigrid ?

Multigrid is a numerical method for solving differential equations, which uses two or more levels of discretization. It can reduce the time needed to solve the problem because a full solution is only required on the coarsest level. At the other levels, "smoothing" is carried out before the error is transferred to the next coarsest level or the correction is transferred to the next finest level. The sequence of levels the method goes through is called a cycle. Briggs [2] gives a good introduction to the multigrid method.

For a linear differential equation

$$\mathcal{L}(\mathbf{u}) = \mathbf{g} \tag{1}$$

the discretization at level h can be written as

$$\mathcal{L}^h(\mathbf{u^h}) = \mathbf{r^h} \tag{2}$$

where at the finest level $\mathbf{r^h} = \mathbf{g^h}$. Then the multigrid V-cycle algorithm can be written as

1. While not at the coarsest level do

   (a) "Smooth" the solution at the current level (h)

   (b) Transfer the error to the next coarsest level ($2h$) using the restriction operator I_h^{2h},

   $$\mathbf{r^{2h}} = I_h^{2h}(\mathbf{r^h} - \mathcal{L}^h(\mathbf{u^h})) \tag{3}$$

   (c) Move to the next coarsest level

   $$\mathcal{L}^{2h}(\mathbf{u^{2h}}) = \mathbf{r^{2h}}, \text{ with initial estimate } \mathbf{u^{2h}} = \mathbf{0} \tag{4}$$

2. Solve the differential equation at the coarsest level

3. While not at the finest level do

   (a) Transfer the solution to the next finest level ($h/2$) using the interpolation operator $I_h^{h/2}$,

   $$\mathbf{u^{h/2}} = \mathbf{u^{h/2}} + I_h^{h/2}(\mathbf{u^{h/2}}) \tag{5}$$

   (b) Move to the next finest level

   $$\mathcal{L}^{h/2}(\mathbf{u^{h/2}}) = \mathbf{r^{h/2}} \tag{6}$$

   (c) "Smooth" the solution at the current level ($h/2$)

This is called a V-cycle because of the pattern it makes moving through the levels (see Figure 1). To converge to a solution this is repeated several times. In our latest work, we are using a full multigrid V-cycle (see Figure 1).

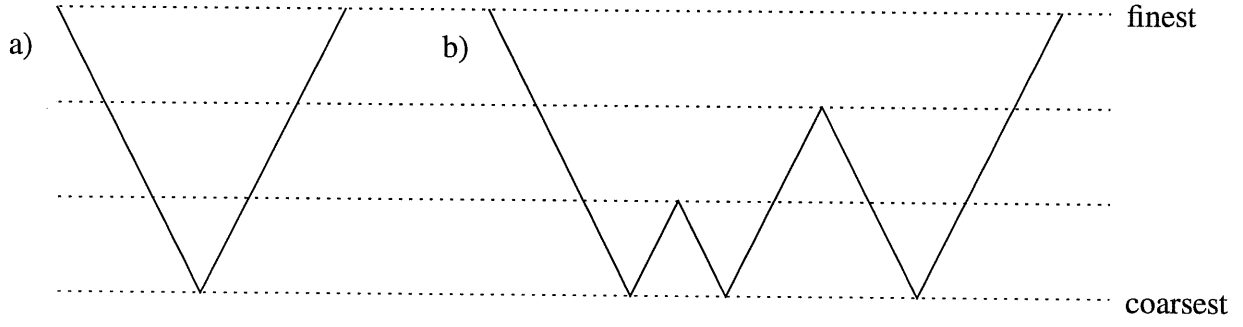

Figure 1: Multigrid cycles. a) V-cycle, b) full multigrid V-cycle

A "smoothing" sweep involves stepping through the solution vector $\mathbf{v^h}$ and solving for each value with the other values fixed. For a linear system of equations, this is the Gauss-Seidel method. A common strategy is to carry out two smoothing sweeps before transferring from fine to coarse and one smoothing sweep after transferring from coarse to fine.

For nonlinear differential differential equations, such as those solved by TOUGH, the multigrid method needs to be modified. We are using the full approximation storage method McCormick [3]. For this method the transfer of the error from fine to coarse is modified as follows:

$$\mathbf{r^{2h}} = I_h^{2h}(\mathbf{r^h}) + \mathcal{L}^{2h}(I_h^{2h}(\mathbf{u^h})) - I_h^{2h}(\mathcal{L}^h(\mathbf{u^h})) \tag{7}$$

The initial estimate on the coarse grid is the restriction of the fine grid solution

$$\mathbf{u^{2h}} = I_h^{2h}(\mathbf{u^h}) \tag{8}$$

and the transfer of the correction from coarse to fine is also modified:

$$\mathbf{v^{h/2}} = \mathbf{v^{h/2}} + I_h^{h/2}(\mathbf{v^h} - I_{h/2}^h(\mathbf{v^{h/2}})) \tag{9}$$

In the following sections, the interpolation and restriction operators used will be discussed.

3 Feasibility study

Part of the work by Yang [5] is concerned with improving the computational efficiency of steady state simulations of geothermal reservoirs. One method that was considered was multigrid. This section summarizes Yang's work on multigrid.

Multigrid was implemented by modifying MULKOM. The test problem for this work was a 2-D vertical square (1km side) with heat input over a fifth of the base, atmospheric conditions at the top and no flow through the rest of the boundary (see Figure 2). Two grids were used - a coarse 5 by 5 grid and a fine 10 by 10 grid. The normal MULKOM/TOUGH solution routine was used to solve on the coarse grid.

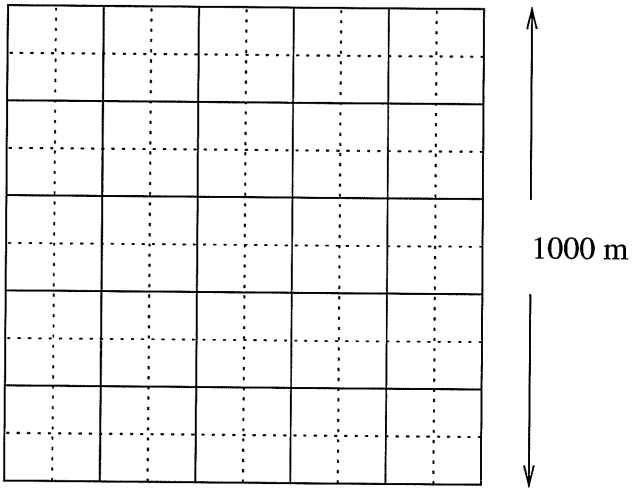

Figure 2: Test problems

When implementing multigrid, choices have to be made for the restriction operator, the interpolation operator and the smoothing method. For this feasibility study the best results were obtained by:

- Transferring the right hand side from fine to coarse, I_f^c, by averaging over the four fine blocks that make up the coarse block and multiplying by a damping factor ω ($0 <= \omega <= 1$)

$$\mathbf{r}^{\mathbf{c}}_{\mathbf{i,j}} = \omega * (\mathbf{r}^{\mathbf{f}}_{\mathbf{2i-1,2j-1}} + \mathbf{r}^{\mathbf{f}}_{\mathbf{2i-1,2j}} + \mathbf{r}^{\mathbf{f}}_{\mathbf{2i,2j-1}} + \mathbf{r}^{\mathbf{f}}_{\mathbf{2i,2j}})/4 \tag{10}$$

  For strongly two-phase flow, values of ω less than 0.5 were found to be necessary to ensure stability.

- Always using the MULKOM solution to the unmodified problem ($\mathbf{r}^{\mathbf{c}}$ not modified by fine grid) on the coarse grid, $\mathbf{u}^{\mathbf{c}}$, as the initial estimate rather than the restricted pressure and temperature or saturation from the fine grid. The restriction that was used was to convert from pressure and temperature or saturation to pressure and enthalpy for each of the four fine blocks, average these pressures and enthalpies to get a pressure and enthalpy for the coarse block, and convert this pressure and enthalpy to a pressure and temperature or saturation. This was found to sometimes give an initial estimate for which MULKOM did not converge.

- Transferring the estimate from the coarse grid to the fine grid by converting the pressure and temperature or saturation estimates on the coarse grid into pressures and enthalpies, computing the changes in pressure and enthalpy from $\mathbf{u}^{\mathbf{c}}$, linearly interpolating these changes multiplied by another damping factor on to the fine grid, and converting the updated pressures and enthalpies on the fine grid into pressures and temperatures or saturations.

- Smoothing using box Gauss-Seidel (see Brandt [1]). Standard Gauss-Seidel involves sweeps through all the blocks in the grid fixing the pressure and temperature or saturation in all but the current block and solving for the pressure and temperature or saturation in the current block using Newton-Raphson. For box Gauss-Seidel, Newton-Raphson is used to solve for the current block and some of its neighbours. As the number of neighbours in the box goes up, the smoothing gets better, but the computational cost goes up.

For the test problem, four different inputs - 1 MW, 2 MW, 4 MW and 6 MW - were used. For 1 MW there is no two-phase zone. For 6 MW there is a large two-phase zone running from the base of the model, above the heat source, to the top of the model and halfway across the top of the model.

The multigrid implementation described above is able to solve the test problem with the four different heat inputs, but is slower than standard MULKOM. As the heat input is increased, multigrid has more difficulty in solving the problem (as does MULKOM).

4 Ironing out the wrinkles

In Yang's preliminary implementation, only two levels were used and changes to the way that standard multigrid transfers between levels were made because of the MULKOM/TOUGH choice of primary variables (pressure and temperature or saturation). To remove these limitations and investigate multigrid further, a "test-bed" program is being written. It is intended to solve the steady-state geothermal equations (as formulated for TOUGH, Pruess [4]) on a square or a cube, with an arbitrary number of multigrid levels, using integrated finite differences (like MULKOM/TOUGH) and with pressure and enthalpy as the primary variables (unlike MULKOM/TOUGH). Our development schedule is:

1. Use integrated finite differences and multigrid to solve Poison's equation. This will check that the outer multigrid loop is working.

2. Develop a single block solver for the steady-state geothermal equations. This will solve for the pressure and enthalpy in a single block with the pressures and enthalpies in the neighbouring blocks fixed. It is the fundamental part of the Gauss-Seidel smoother.

3. Implement standard restriction and interpolation operators. The restriction operator, to transfer values from the fine grid to the coarse grid, will average the values in the fine grid blocks that make up the coarse grid block. The interpolation operator, to transfer values from the coarse grid to the fine grid, will use the value in the coarse grid block in all the fine grid blocks contained in the coarse grid block. These will be easier to implement than in the feasibility study because it will not be necessary to convert between pressure and enthalpy and pressure and temperature or saturation.

4. Investigate the performance of the method on the test problems from the feasibility study.

The implementation of integrated finite differences and multigrid to solve Poison's equation was relatively straight forward and the expected multigrid convergence rate, the ratio of the error at the end of the V-cycle to the error at the beginning of the V-cycle, of 0.1 was achieved. In the implementation, care had to be taken with the boundary conditions.

At present we are working on the single block solver. This is proving difficult. The following subsection discusses the single block solver.

4.1 Single block solver

The single block solver should be able to solve for the pressure and enthalpy in a block when the pressures and enthalpies in the neighbouring four, in 2-D, or six, in 3-D, blocks are specified and fixed. The solver needs to work even when the initial guesses are poor, because it was noticed in the preliminary implementation that poor estimates were sometimes obtained when restricting from the coarse grid to the fine grid, especially on the boundary of two-phase zones.

Our current solver uses Newton-Raphson with time stepping and care when crossing the phase change boundaries. This is the same method as in TOUGH, but it is using different primary variables.

To test the solver, problems in 2-D with no flow through three sides and the pressure and enthalpy specified for the fourth. With all combinations of boundary orientation (specified pressure and enthalpy either on top or to the side), initial estimate (liquid, two-phase or vapour) and boundary conditions (liquid, two-phase or vapour), there are 18 test problems.

TOUGH converges for all the test problems, but in some problems to the wrong answer -

- For two-phase boundary conditions on the side, the block converges to same pressure and temperature as on the boundary, but has a different saturation (enthalpy).
- For vapour boundary conditions on the top, the block converges to liquid conditions, but for some initial estimates the pressure gradient is liquid-static (correct) and for other initial estimates the pressure gradient is vapour-static (incorrect).

We believe that these errors can be corrected by -

- Using liquid pressures

$$p_l = p + g(d/2)(1 - s_l)(1 - s_l) * (\rho_v - \rho_l) \tag{11}$$

for calculating the liquid flux and vapour pressures

$$p_v = p + g(d/2)s_l s_l(\rho_v - \rho l) \tag{12}$$

for calculating the vapour flux. The concept of liquid and vapour pressures is similar to capillarity pressure, but is independent of rock properties. The pressures at the top and bottom of the block were estimated by assuming that the block is hydrostatic with vapour over liquid and that the block pressure is the average of the pressure with depth. The liquid pressure at the centre of the block, p_l, was calculated by assuming liquid-static from the bottom to the centre. The vapour pressure at the centre of the block, p_v, was calculated by assuming vapour-static from the top to the centre.

- Modifying the gravity correction for the liquid flux from

$$gsin(\beta)(\rho_{l1} + \rho_{l2})/2 \tag{13}$$

and the gravity correction for the vapour flux from

$$gsin(\beta) * (\rho_{v1} + \rho_{v2})/2 \tag{14}$$

to both be

$$gsin(\beta)(d_1(s_{l1}r_{l1} + (1 - s_{l1})\rho_{v1}) + d_2(s_{l2}\rho_{l2} + (1 - s_{l2})\rho_{v2})/(2 * (d_1 + d_2)) \tag{15}$$

Our single block solver with the above modifications does not converge for some of the test problems. We think that the convergence problems are due to the discretized equations not being smooth. There are discontinuities in the slopes of the residuals across the phase change boundaries which both TOUGH and our block solver take care over, but there are also discontinuities in the slopes across any of the lines where a flux changes sign because of the upwinding. TOUGH does not suffer as badly as our single block solver and this may be due to the choice of primary variables. We are now considering making the above modifications to TOUGH to see if it can then solve all the test problems correctly.

5 Summary

A feasibility study of applying multigrid to geothermal reservoir simulation has been carried out and showed that the method could solve the equations but the implementation was slower than standard TOUGH. This study identified a number of problems in implementing multigrid and these are currently being investigated.

A major problem is how to solve for the pressure and enthalpy in a single block with pressures and enthalpies specified on the boundaries. There are difficulties with the formulation of the "capillary" pressure and also because the residuals are not smooth across phase changes and when flows change direction (due to upwinding).

Acknowledgement

This work has benefitted from many discussions with Stephen F. McCormick, University of Colorado, Boulder.

References

[1] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods, Proceedings of Conference at Koln-Porz, November 23-27, 1981*. Springer-Verlag, 1982.

[2] William L. Briggs. *A multigrid tutorial.* Society for Industrial and Applied Mathematics, 1987.

[3] Stephen F. McCormick. *Multilevel adaptive methods for partial differential equations*, volume 6 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1989.

[4] Karsten Pruess. TOUGH2. A general-purpose numerical simulator for multiphase fluid and heat flow. Technical Report LBL-29400, Earth Sciences Division, Lawrence Berkeley Laboratory, 1 Cyclotron Road, Berkeley, CA94720, May 1991.

[5] Zhongke Yang. *Numerical simulations of geothermal systems with an unsaturated zone.* PhD thesis, University of Auckland, 1994.