A 2-D Numerical Method for Tracking a Moving Water Table

Emily K. Clearwater, Michael J. O'Sullivan, and Rosalind A. Archer

Department of Engineering Science, University of Auckland, Auckland, New Zealand
emily.clearwater@auckand.ac.nz

Keywords
Water table, TOUGH2, moving boundary, numerical method, modelling

ABSTRACT

A new numerical method has been incorporated into the existing TOUGH2 framework to allow for more accurate tracking of a moving water table in a non-isothermal system. The method is based on the calculation of the mass flow into or out of the top surface of the model (the water table surface). The correct position of the water table is one where fluid is no longer moving into or out of the top surface of the water table block. This paper describes a numerical scheme for accurately tracking the movement of the water table. It is implemented through python scripting without interfering with the TOUGH2 code.

To demonstrate the method a 2-D non-isothermal model representing a shallow reservoir with constant fluid production is considered. For comparison, an equivalent air/water EOS model was run using standard TOUGH2. The results demonstrate the improvement the new method holds over current approaches of either using a fixed water table or using an air/water model and moving the water table block by block.

Further development and extension of the method will be explored in order to extend water table tracking to boiling flow and to include the vadose zone.

1. Introduction

Geothermal reservoir modelling is an important tool for monitoring shallow reservoir dynamics and gaining understanding of the connection between the deep reservoir and shallow regions. TOUGH2 is the industry standard tool used to simulate movement of mass and heat throughout a geothermal system.

State-of-the-art software used for numerical modelling of non-isothermal, two-phase flow in geothermal systems does not allow for the accurate tracking of the water table over time. The usual approaches for modelling the shallow zones are: (i) to use a fixed top for the model thus excluding the vadose zone and holding the water table in a fixed position, or (ii) to use an air/water model where the location of the water table is inferred from the air mass fraction in shallow blocks – an approach which often suffers from convergence issues and whose accuracy is dependent on grid resolution.

Previous work by the authors described the current approaches used in geothermal and groundwater contexts, and discussed two new methods for tracking the movement of the water table, based around the TOUGH2 code. The alternative methods use a fully saturated, wateronly, one dimensional model where the top surface of the grid moves at each time step as required from a mass balance calculation (derived either through iteration or a flux approximation). The current work extends these methods, demonstrating their applicability in two dimensions.

Clearwater *et al.* (2014) gives further discussion on how approaches (i) and (ii) are applied and why they are not satisfactory. There remains a need for an improved approach which accurately tracks the water table while remaining numerically stable.

2. Tracking a Moving Water Table

The scope for this work is limited to incorporating new numerical methods into the existing TOUGH2 framework. Thus, the new approach needs to employ the integrated finite difference technique utilized by TOUGH2. The region of interest is discretized in space (into blocks or elements where the *i*th block has a volume V_i and connection area α_{ij} to the adjacent *j*th block), and as time is incremented mass and energy flux are evaluated at the new time step. Our aim is to achieve an all-encompassing (saturated zone, vadose zone and boiling effects) water table tracking method and the relatively simple models and algorithms discussed here are a step in this process and will be built upon.

The two dimensional approach utilized in this work is an extension of the one dimensional isothermal iterative moving boundary method described in Clearwater *et al.* (2014). Energy conservation has now been included but the water table surface is still set at the top of the model. Thus, no vadose zone or capillarity effects are taken into account. The method employed aligns with the algorithm of Crowe *et al.* (1999), however due to the difference in the governing equations, the numerical strategy is different.

3. Numerical Derivation

Appendix A of Pruess *et al.* (1999) describes the general form of the mass and energy balances utilized by TOUGH2 to simulate mass and heat flow through porous media. The mass balance equation can be written as:

$$V_i\left(A_{mi}^{n+1} - A_{mi}^{n}\right) = \sum_j a_{ij} F_{mij}^{n+1}\,\Delta t_n + Q_{mi}^{n+1}\,\Delta t_n \tag{1}$$

The LHS of equation (1) is mass accumulation, F_{mij}^{n+1} is the mass flux, and Q_{mi}^{n+1} denotes mass extracted or injected from sinks and sources.

The new method utilized for this work is based on the calculation of the mass flow into or out of the top surface of the model (the water table surface). The correct position of the water table is one where fluid is no longer moving into or out of the top surface of the water table block. For a given time step this position is found when the sum of fluxes across the top surface, F_{m0i}^{n+1}, is zero (see Figure 1) . The top surface of the grid (the water table elevation for the previous time step) is then moved to lie at the correct water table position for the current time step. The following derivation describes how one can calculate the correct volume required for a time step without interfering with the TOUGH2 code.

The mass balance equation we wish to solve can be written as:

$$V_i^{n+1,k} A_{mi}^{n+1,k} - V_i^{n} A_{mi}^{n} = a_{0i}\; F_{m0i}^{n+1,k}\,\Delta t_n - a_{ij} F_{mij}^{n+1,k}\,\Delta t_n + Q_{mi}^{n+1,k}\,\Delta t_n \tag{2}$$

In this method, we iteratively adjust $V_i^{n+1,k}$ and re-solve equation (2) until $F_{m0i}^{n+1,k} = 0$ (i.e. the bottom Figure 1). Because the new volume is fed back into TOUGH2 to be re-solved for iteration k+1 of time step Δt, TOUGH2 is actually solving the following:

$$V_i^{n+1,k} A_{mi}^{n+1,k} - V_i^{n+1,k} A_{mi}^{n} = a_{oi} F_{m0i}^{n+1,\,k}\,\Delta t_n - a_{ij} F_{mij}^{n+1,k}\,\Delta t_n + Q_{mi}^{n+1,k}\,\Delta t_n + a_{pi}^{n+1,k} F_{mpi}^{n+1,k}\,\Delta t_n - a_{iq}^{n+1,k} F_{miq}^{n+1,k}\,\Delta t_n \tag{3}$$

The iterative process starts with a standard solve of the TOUGH2 time step, with

$$V_i^{n+1,0} = V_i^{n} \tag{4}$$

Thus, equation (3) becomes:

$$V_i^{n} A_{mi}^{n+1,1} - V_i^{n} A_{mi}^{n} = a_{oi} F_{m0i}^{n+1,1}\,\Delta t_n - a_{ij} F_{mij}^{n+1,1}\,\Delta t_n + Q_{mi}^{n+1,1}\,\Delta t_n + a_{pi}^{n} F_{mpi}^{n+1,k}\,\Delta t_n - a_{iq}^{n} F_{miq}^{n+1,k}\,\Delta t_n \tag{5}$$

As described earlier, we want to solve for $F_{m0i}^{n+1,k} = 0$. Hence, we really wish to solve:

$$V_i^{n+1,1} A_{mi}^{n+1,1} - V_i^{n} A_{mi}^{n} = 0 - a_{ij} F_{mij}^{n+1,1}\,\Delta t_n + Q_{mi}^{n+1,1}\,\Delta t_n + a_{pi}^{n+1,k} F_{mpi}^{n+1,k}\,\Delta t_n - a_{iq}^{n+1,k} F_{miq}^{n+1,k}\,\Delta t_n \tag{6}$$

In which

$$V_i^{n+1,1} = V_i^{n} + a_{0i}\,\Delta h^{(1)} \tag{7}$$

And

$$\Delta h^{(1)} = h_i^{n+1,1} - h_i^{n} \tag{8}$$

As shown in Figure 1.

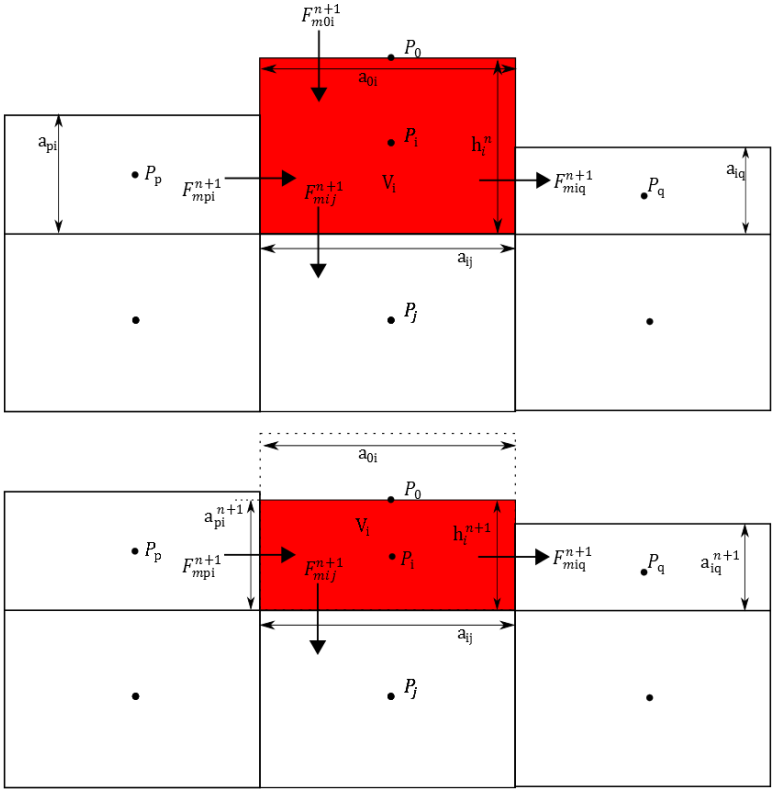

Figure 1. Top, TOUGH2 mass balance on water table block *i* at time t_n. Bottom, mass balance on moving water table block *i* at time t_{n+1}.

Substituting equation (7) into equation (6) results in:

$$\begin{aligned} & V_i^n A_{mi}^{n+1,1} - V_i^n A_{mi}^n = \\ & -A_{mi}^{n+1,1} a_{0i} \Delta h^{(1)} - a_{ij} F_{mij}^{n+1,1} \Delta t_n \\ & + Q_{mi}^{n+1,1} \Delta t_n + a_{pi}^{n+1,k} F_{mpi}^{n+1,k} \Delta t_n \\ & - a_{iq}^{n+1,k} F_{miq}^{n+1,k} \Delta t_n \end{aligned} \tag{9}$$

By comparing equations (5) and (9), we find an equation for evaluating $\Delta h^{(1)}$:

$$\begin{aligned} & A_{mi}^{n+1,1} a_{0i} \Delta h^{(1)} = a_{oi} F_{m0i}^{n+1,1} \Delta t_n \\ & + \left(a_{pi}^{n+1,k} - a_{pi}^n \right) F_{mpi}^{n+1,k} \Delta t_n \\ & - \left(a_{iq}^{n+1,k} - a_{iq}^n \right) F_{miq}^{n+1,k} \Delta t_n \end{aligned} \tag{10}$$

To make progress we neglect the 2nd and 3rd terms on the RHS. This assumes that the change in water table over the time step will be small, so any change in the cross sectional area between the water table block and its neighbours is also small, and thus the associated flux crossing those interfaces is negligible. This assumption reduces (10) to (11), and provides the first estimate for Δh .

$$\Delta h^{(1)} = \frac{F_{m0i}^{n+1,1} \Delta t_n}{A_{mi}^{n+1,1}} \tag{11}$$

To proceed past one iteration (i.e. to go past $k = 1$ and iterate towards a volume $V_i^{n+1,k}$ which results in $F_{m0i}^{n+1,1}$=0), we re-solve equation (1) using TOUGH2. The volume $V_i^{n+1,k}$ is now the new updated volume and the areas of the vertical sides are found from equation (8). Again (as for equation (4)), what we really want to solve rather than equation (1), is:

With

$$V_i^{n+1,k} A_{mi}^{n+1,k} - V_i^n A_{mi}^n = 0 - a_{ij} F_{mij}^{n+1,\,k} \Delta t_n + Q_{mi}^{n+1,k} \Delta t_n + a_{pi}^{n+1,k} F_{mpi}^{n+1,k} \Delta t_n - a_{iq}^{n+1,k} F_{miq}^{n+1,k} \Delta t_n \tag{12}$$

$$V_i^{n+1,k} = V_i^n + a_{0i} \Delta h^{(k)} \tag{13}$$

Substituting equation (13) into (3) we get:

$$\begin{aligned} & V_i^{n+1,k} A_{mi}^{n+1,k} - V_i^n A_{mi}^n = \Delta h^{(k)} a_{0i} A_{mi}^n + a_{0i} F_{m0i}^{n+1,k} \Delta t_n \\ & - a_{ij} F_{mij}^{n+1,\,k} \Delta t_n + Q_{mi}^{n+1,k} \Delta t_n + a_{pi}^{n+1,k} F_{mpi}^{n+1,k} \Delta t_n - a_{iq}^{n+1,k} F_{miq}^{n+1,k} \Delta t_n \end{aligned} \tag{14}$$

Comparing equation (14) and equation (12), the first two terms on the RHS of equation (14) should balance out. When they do, the volume is correct, the water table surface is in the correct position, and simulation can proceed to the next time step. If the terms do not balance out, $\Delta h^{(k+1)}$ may be calculated using:

$$\Delta h^{(k+1)} = -\frac{a_{0i} F_{m0i}^{n+1,k} \Delta t_n}{a_{0i} A_{mi}^{n+1,k}} \tag{15}$$

And

$$\Delta h^{(k+1)} = h_i^{n+1,k+1} - h_i^n \tag{16}$$

The procedure is then iterated by applying $h_i^{n+1,k+1}$ to block volumes and connection areas and distances, and re-solving the time step (Newton Raphson iteration could be used instead) until the first two terms on the RHS of equation (14) balance out.

The iterative procedure is implemented through a python script, utilising the PyTOUGH library (Croucher 2011) for interaction with AUTOUGH2 files. The user runs the script rather than directly running AUTOUGH2. The script loads the input files, identifies water table blocks, calls AUTOUGH2, inspects the results, and updates the geometry and input files accordingly (to re-run the time step with an altered block volume, or to move on to the next time step).

4. Example and Validation

To demonstrate the method a 2-D non-isothermal model representing a shallow reservoir with constant fluid production is considered. For comparison, an equivalent air/water EOS model was run using standard TOUGH2. The numerical simulations were carried out using AUTOUGH2 (Bullivant (1990), Yeh *et al.* (2012)), a version of TOUGH2 (Pruess *et al.* 1999) developed at the University of Auckland. The moving water table simulations use EOS1 (water only) and the air/water model uses EOS3.

The model set up is shown in Figure 2. The grid extends 520m horizontally and 300m vertically. It is discretized into 26 columns (20m wide) and 60 layers (thickness of 5m). A low amount of background heat and mass are applied at the base of the model. After a natural state simulation, fluid is extracted at a constant rate of 20kg/s (distributed evenly across 4 layers from -240m to -260m) from the 13th column (centered at 250m). Results are shown in Figure 3 under the label 'iterative model'.

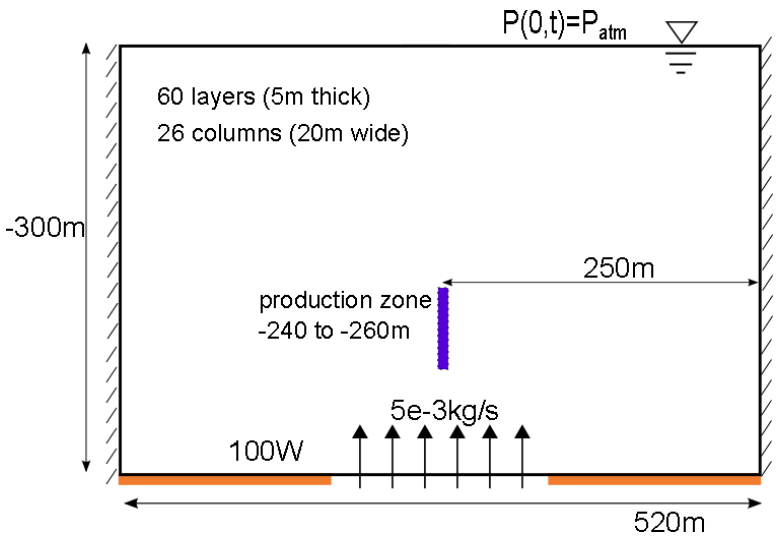

Figure 2. 2D production model setup.

A comparative air/water simulation was performed using the same model setup as described above, but with the air/water module (EOS3) in AUTOUGH2. The 'air/water model' results shown in Figure 3 are an approximate water table location derived from the simulation results. The approximate location is found by finding the bottom most layer with a gas saturation of more than 0.001. A hydrostatic profile was then calculated upwards from the center of this block, and the level where atmospheric pressure was reached was set as the water table location. The plot in Figure 3 shows the limitation of the air water method - pressures are calculated as averages at the block centers, and so the grid refinement has an effect on the resulting water table position.

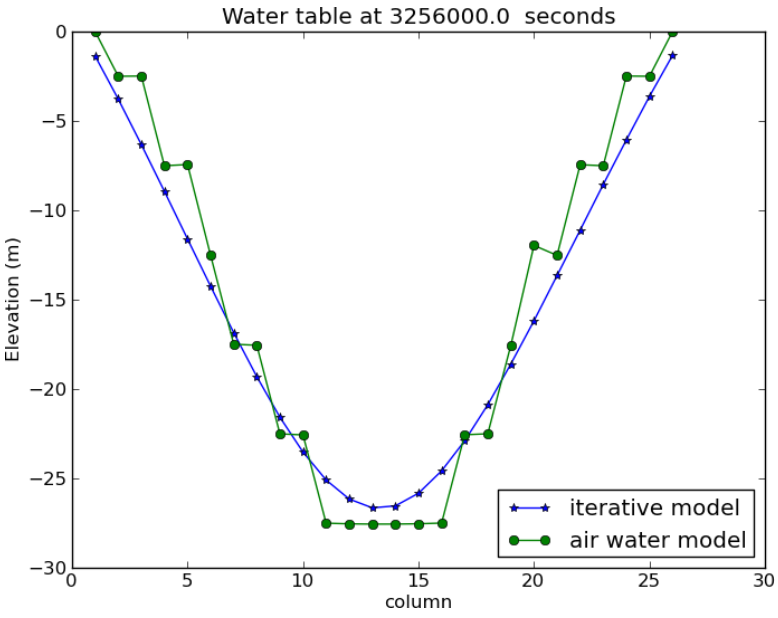

Figure 3. Simulation results for the 2D production model.

The air-water model results continue in a step-wise fashion over time, causing them to be inaccurate at certain times during the simulation. The size of the vertical part of the step varies depending on grid layer resolution, showing that the accuracy of the air/water method is dependent on grid resolution. The flat horizontal part of the step of the air/water solution is due to the pressure remaining constant while air saturation increases as the water table falls through the layer. These inaccuracies are avoided with the moving water table tracking method presented here and the 'it-

erative model' results show a continuous movement of the water table, independent of vertical grid resolution. However, it must be noted that the new method, as it is at present, excludes the vadose zone, and requires more iterations over each time step than the standard AUTOUGH2 air/water approach.

5. Conclusions and Further Work

The results demonstrate the improvement this method holds over current standard approaches (fixed water table or air/water) for modelling a moving water table. Further development and extension of the method, plus other mass balance approximation methods, will be explored in order to extend water table tracking to boiling flow and to include the vadose zone.

The viability of this new approach in terms of computational expense is still under consideration and will be further investigated as larger model are experimented with. Applicability to boiling scenarios is also yet to be confirmed. However, the method is an accurate water table tracking alternative to the air/water approach when applied to 2-D, non-isothermal, TOUGH2 simulations.

References

Bullivant, D. P. (1990), "Making mulkom/tough faster and easier to use". *Proc. TOUGH Workshop 1995*.

Clearwater, E. K., M. J. O. Sullivan and R. A. Archer (2014), "A numerical method for tracking a moving water-table". *Proceedings 36th New Zealand Geothermal Workshop*.

Croucher, A. E. (2011), "Pytough: A python scripting library for automating tough2 simulations". *33rd New Zealand Geothermal Workshop*.

Crowe, A. S., S. G. Shikaze and F. W. Schwartz (1999), "A grid generating algorithm for simulating a fluctuating water table boundary in heterogeneous unconfined aquifers". *Advances in Water Resources*. v. 22, p. 567 - 576.

Pruess, K., C. Oldenburg and G. Moridis (1999), "Tough2 user's guide, version 2.0", Lawrence Berkeley National Laboratory.

Yeh, A., A. E. Croucher and M. J. O'sullivan (2012), "Recent developments in the autough2 simulator". *TOUGH Symposium 2012*.