IMPROVED CONVERGENCE FOR AIR-WATER AND CO_2-WATER TOUGH2 SIMULATIONS

John O'Sullivan[1], Adrian Croucher[1], Angus Yeh[1] and Michael O'Sullivan[1]

[1]Department of Engineering Science, University of Auckland, Private Bag 92019, Auckland 1142, New Zealand

jp.osullivan@auckland.ac.nz

Keywords: *TOUGH2, convergence, inverse modelling.*

ABSTRACT

Numerical modelling has become an important tool in managing geothermal systems and planning their exploitation. The TOUGH2 simulator is the industry standard tool for developing numerical models. It includes several different equation-of-state modules and thus can be used for modelling many different kinds of geothermal fields as well as other complex sub-surface flow problems. Under certain conditions TOUGH2 has difficulty in running a model up to the very large times required for a natural state simulation as it stalls at a relatively small time step size. This problem leads to slow model development and also poses a significant obstacle to inverse modelling using iTOUGH2 or PEST as forward simulations are more computationally expensive and may not finish. In this paper two conditions leading to stalled simulations are identified and analysed. A correction to the air-water and CO_2-water equation-of-state modules eliminates the first problem. The second is eliminated by an adjustment to the saturation temperature calculation. Results are presented showing a dramatic improvement in the simulation time required to achieve a steady state for large models of real geothermal systems.

1. INTRODUCTION

Numerical simulation is an important tool for planning and managing the development of geothermal systems (Burnell *et al.*, 2012). Since its development in the 1980s TOUGH2 (Pruess *et al*., 1999) has become the industry standard simulator and is now widely used. At the same time TOUGH2 has been extended to include several different equation-of-state modules allowing it to be used for modelling many different kinds of geothermal fields, as well as other complex sub-surface flow problems.

Many TOUGH2 simulations must be carried out in order to obtain a well-calibrated model of a geothermal system (O'Sullivan *et al*., 2001). During the calibration process, model parameters such as permeabilities and heat and mass inputs are adjusted in order to match measured observations. Under certain circumstances these changes can cause the TOUGH2 simulator to stall at a relatively small time step size, which then makes it difficult or impossible to reach the desired large simulation time for a natural state model. This behaviour is well known to geothermal reservoir modellers and has also been reported by Noy *et al.* (2012) in CO_2 sequestration simulations using TOUGH2. Previously O'Sullivan *et al*. (2013) proposed a scripting approach to automate interventions into stalled simulations, which allowed some simulations to continue but did not address the underlying problem.

The objective of the present work was to identify stalled simulations, study the cause of the stalling, determine fixes for the problem and apply them directly to the AUTOUGH2 code, the University of Auckland version of TOUGH2 (Yeh *et al*., 2012). Examples of stalled simulations of real geothermal systems were collected over several months. From these, simple test simulations that replicated the problem were created. The analysis technique developed for studying the problem is described below, after a brief discussion of the TOUGH2 solution algorithm. It is important to understand how the TOUGH2 solution algorithm works to be able to understand why the problem occurs and how it can be corrected. Three examples of stalled simulations of real geothermal systems have been selected and the results of the analysis are presented for each model. The results for each simulation using a corrected version of the AUTOUGH2 code are presented and the improvement is discussed.

2. TOUGH2 SIMULATOR

The TOUGH2 simulator solves mass and energy-balance equations to determine the properties of non-isothermal flows of multiphase, multicomponent fluids in porous and fractured media (Pruess *et al*., 1999).

2.1 Governing Equations

The mass-balance and energy-balance equations solved by TOUGH2 can be written in the following form (O'Sullivan, 2012):

$$\frac{d}{dt}\int_V A_\kappa dV = -\int_A \mathbf{n}.\mathbf{F}_\kappa dA + \int_V q_\kappa dV \qquad (1)$$

where V is the volume of integration, A_κ is the amount of each quantity κ within the volume, A is the surface of the volume, $\mathbf{F}_\kappa$ is the flux of quantity κ across the surface A, $\mathbf{n}$ is the normal vector to the surface A and q_κ represents any sources or sinks in the volume. For this work we considered two common types of systems each requiring governing equations for three quantities. The first system uses conservation equations for mass of water, mass of CO_2 and for energy. The equation of state for this type of simulation is referred to as EOS2. The second system uses conservation equations for mass of water, mass of air and for energy. The equation of state for this type of simulation is referred to as EOS3.

The amount of each component per unit volume is calculated as the sum of the contributions from each phase as shown in Equation (2):

$$A_\kappa = \varphi(\rho_l S_l X_{\kappa l} + \rho_g S_g X_{\kappa g}) \qquad (2)$$

Here φ is the porosity and for each phase, β, the density is given by ρ_β, the saturation by S_β and the mass fraction by $X_{\kappa\beta}$. The liquid phase is indicated by the subscript l and the gas phase by the subscript g. For the amount of energy in the volume the definition includes an additional term for the contribution of the rock:

$$A_e = (1-\varphi)\rho_r c_r T + \varphi(\rho_l u_l S_l + \rho_g u_g S_g) \quad (3)$$

T is the temperature, ρ_r the density of the rock, c_r its heat capacity and u_β the internal energy of phase β. The flux of each component $\mathbf{F}_\kappa$ in Equation (1) is calculated using the contribution of each phase $\mathbf{F}_\beta$ weighted by the mass fraction:

$$\mathbf{F}_\kappa = X_{\kappa l}\mathbf{F}_l + X_{\kappa g}\mathbf{F}_g \quad (4)$$

In some equations of state for TOUGH2 a dispersion term can be added to (4) but in most geothermal systems the effects of diffusion and hydrodynamic dispersion are small. For the energy flux a conductive term is also included where K is the thermal conductivity and the enthalpy of each phase β must be taken into account as shown in Equation (5).

$$\mathbf{F}_e = h_l\mathbf{F}_l + h_g\mathbf{F}_g - K\nabla T \quad (5)$$

The flux of each phase is given by the two-phase form of Darcy's Law:

$$\mathbf{F}_\beta = -\frac{\mathbf{k}\,k_{r\beta}}{\nu_\beta}(\nabla p + \rho_\beta \mathbf{g}) \quad (6)$$

Here $\mathbf{k}$ is the permeability tensor (usually assumed to be diagonal), $k_{r\beta}$ the relative permeability of the phase β, ν_β its viscosity, p the pressure and $\mathbf{g}$ is gravity. Note that for this work the effect of capillary pressure was not considered.

Discretising in space and applying implicit time stepping reduces Equation (1) to a set of coupled non-linear equations which can be written as:

$$\frac{V_i}{\Delta t^{n+1}}\left(A_{\kappa i}^{n+1} - A_{\kappa i}^{n}\right) = -\sum_j a_{ij}F_{\kappa ij}^{n+1} + \sum_p q_{\kappa ip}^{n+1} \quad (7)$$

In Equation (7) the superscript refers to the time step at which the quantity is calculated. The term a_{ij} is the area of the interface between block i and block j and $F_{\kappa ij}^{n+1}$ is the flux of each quantity κ across the same interface. The term $q_{\kappa ip}^{n+1}$ represents p separate source terms of quantity κ in block i and finally the time step size is given by Δt^{n+1}.

The discrete form of Equation (6) is used to calculate the fluxes of each quantity:

$$F_{\beta ij}^{n+1} = -\left(\frac{k\,k_{r\beta}}{\nu_\beta}\right)_{ij}^{n+1}\left[\frac{p_j^{n+1} - p_i^{n+1}}{d_{ij}} - \rho_{\beta ij}^{n+1} g_{ij}\right] \quad (8)$$

To calculate the relative permeability and viscosity term in Equation (8) upwind differencing is used:

$$\left(\frac{k_{r\beta}}{\nu_\beta}\right)_{ij}^{n+1} = \begin{cases} \left(\frac{k_{r\beta}}{\nu_\beta}\right)_i^{n+1} & \text{phase } \beta \text{ flows } i \text{ to } j \\ \left(\frac{k_{r\beta}}{\nu_\beta}\right)_j^{n+1} & \text{phase } \beta \text{ flows } j \text{ to } i \end{cases} \quad (9)$$

The direction in which each phase flows is determined by the pressure gradient and the effects of gravity as defined by the following equation:

$$G_{\beta ij}^{n+1} = \frac{p_j^{n+1} - p_i^{n+1}}{d_{ij}} - \rho_{\beta ij}^{n+1} g_{ij} \quad (10)$$

Now $G_{\beta ij}^{n+1} > 0$ means that phase β flows from block j to block i and $G_{\beta ij}^{n+1} < 0$ means the opposite.

In both Equation (8) and Equation (10) g_{ij} is the component of gravity acting normally to the interface between block i and block j and the density of the phase at the interface $\rho_{\beta ij}^{n+1}$ is calculated simply from the average for the two blocks:

$$\rho_{\beta ij}^{n+1} = \frac{\rho_{\beta i}^{n+1} + \rho_{\beta j}^{n+1}}{2} \quad (11)$$

In Equation (8) the permeability and block distance terms are combined and calculated as a harmonic weight:

$$\frac{d_{ij}}{k_{ij}} = \frac{d_i}{k_i} + \frac{d_j}{k_j} \quad (12)$$

In Equation (10) the block distance term is simply the sum of the distance from each block centre to their shared interface:

$$d_{ij} = d_i + d_j \quad (13)$$

The values of the secondary parameters h_β, u_β, ρ_β and ν_β are calculated using the primary variables and steam table data supplied through the EOS modules.

2.2 Solution Process

Gathering all the terms in Equation (7) to the left hand side gives a coupled system of nonlinear equations, which must be solved at each time step to calculate the new values of the primary variables in each block:

$$\mathbf{r}(\mathbf{x}) = \mathbf{0} \quad (14)$$

This system of equations is solved iteratively using the Newton-Raphson (NR) method which can be written in the following form (Pruess, 1999):

$$-\mathbf{J}_k(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathbf{r}(\mathbf{x}_k) \quad (15)$$

where k represents the NR iteration number and $\mathbf{J}$ is the Jacobian matrix of the system of equations defined as:

$$\mathbf{J} = \left[\frac{\partial \mathbf{r}}{\partial \mathbf{x}}\right] \quad (16)$$

In TOUGH2 the Jacobian is calculated numerically by evaluating the change in $\mathbf{r}$ corresponding to a small change in $\mathbf{x}$.

Once $\mathbf{r}(\mathbf{x}_k)$ is reduced to the defined convergence limit, the NR method is complete and the values of the primary variables at the new time step have been calculated. TOUGH2 uses adaptive time stepping which means that if the NR method successfully converges within a specified number of iterations then the time step size is increased. If it successfully converges, but only after more than the specified number of iterations, the time step remains the same. When the NR does not converge within a specified iteration number limit then the time step is reduced and

another attempt is made at solving for the primary variables.

The simulation is complete for a natural state model once the total simulated time reaches the required target, usually set at ~1.0x10^{15}s. This large target time does not reflect the real time that the system took to develop over geological time but rather is a computational technique for ensuring that the system is at equilibrium.

2.3 Stalled simulations

Due to the highly nonlinear nature of the equations governing flow in geothermal systems, the adaptive time stepping often reduces and increases the time step size many times during the solution process. However, in some cases the solution process stalls with the time step unable to increase above a relatively small value. This prevents the simulation from reaching the specified total time and hence it does not complete. This behaviour is often manifested by a cycle of time step reductions followed by a single time step increase. Figure (1), below, shows this type of behaviour for two different models: one of the Lihir Island geothermal system and the other of the Wayang Windu geothermal system.

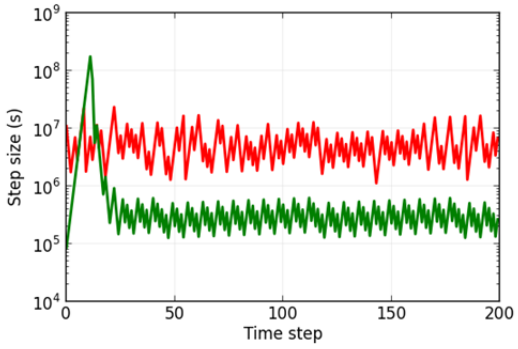

Figure 1: Time step size for two stalled simulations. Results for the Wayang Windu model using EOS2 (-) and the Lihir Island model using EOS3 (-).

This situation occurs when the NR method is able to converge within the maximum number of iterations at one time step size, but cannot converge at a larger time step size. TOUGH2 can be set to output the maximum residual, $\mathbf{r}(\mathbf{x}_k)$, at each iteration throughout the simulation. This reveals that the problem is usually caused by the residual at the larger time step not falling below the convergence criterion for some particular block.

Often the block in question is close to the interface between a saturated and an unsaturated zone. The interface is usually moving and hence two-phase conditions are either evolving or disappearing within blocks in the local area. By examining the conditions in the area, it is possible to determine how the interface is moving and estimate the new primary variables. In the past modellers have attempted to implement this fix by intervening in the simulation either manually or by using scripts (O'Sullivan *et al*., 2012) to adjust the primary variables in the block or in its neighbours.

In some cases this allows the simulation to progress and complete, but in others it does not, in which case it may be necessary to take the more drastic step of adjusting local permeabilities. This is unsatisfactory not only because there may be no physical basis for the permeability change, but also because it is not guaranteed to solve the problem. A much more satisfactory approach is to analyse and understand the problem and develop a solution within the AUTOUGH2 code itself.

2.4 Analysis Technique

A number of models were investigated and examples of the stalled solution behaviour were found. By examining these examples, a number of simple models were created which also displayed the stalling behaviour, and an analysis technique was developed. This technique required editing the AUTOUGH2 code to output full information for both the primary variables and secondary parameters at the problem block and its neighbours. The maximum time step size was then limited to enable the behaviour of each variable and parameter to be recorded and plotted during a stalled simulation.

The results of this investigation are given in the following sections for three models. The first two are examples of a moving interface between a saturated and unsaturated zone. One model uses EOS2 and the other EOS3. The third model is an EOS2 simulation that has stalled at a relatively large time step and the block responsible is not near an unsaturated zone. In all three examples, plots of the key variables and parameters are given and the analysis discussed. A solution for the stalling problem is proposed and the results of the improved simulation are shown.

3. CASE ONE: A MOVING WATER TABLE

The first example of a simulation that has stalled due to a moving unsaturated interface is taken from a model of the Lihir Island geothermal system in Papua New Guinea. The model uses EOS3 to solve the heat and mass transport in both the deep reservoir and the shallow unsaturated zone up to the surface. Thus the model grid follows the surface topography and bathymetry, and air atmosphere blocks are used on land and wet atmosphere blocks of the appropriate pressure are used to represent the sea. The model consists of 9683 blocks and covers a physical area of 103 km^2. For more details of the model refer to O'Sullivan *et al*. (2011).

From Figure (1) it can be seen that the simulation has stalled at a time step of approximately 3x10^5 seconds. Examining the simulation output revealed that the block causing the stall was in the 6th layer of the model, which is only one layer below the surface at that location. The block itself is two-phase, as is the one above it, but the block below contains single phase liquid.

At the point the simulation stalls, the gas phase in the problem block evolves and then disappears again, from one NR iteration to the next. In Figure 3 it can be seen that the flow direction of the gas phase between the problem block and the block above also changes direction with each NR iteration, preventing the NR method from converging. Mathematically this behaviour is caused by a discontinuity in the value of ρ_g calculated using Equation (11) which in turn causes discontinuities in the calculations of $\mathbf{r}(\mathbf{x})$ and $\mathbf{J}$ during the NR solve.

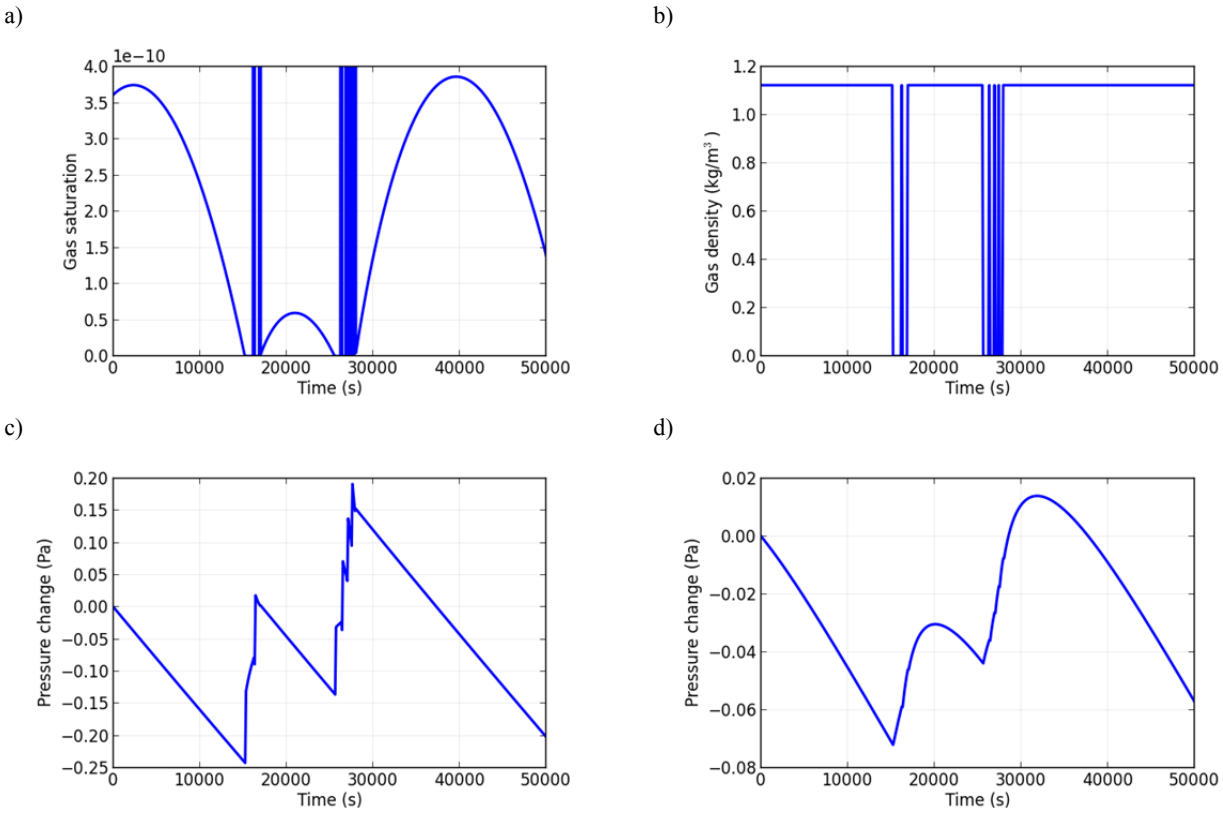

Figure 2: Results for (a) gas saturation, (b) gas density and (c) pressure change in the problem block in a stalled EOS3 simulation. The results for the pressure change in the block beneath the problem block are shown in (d).

The discontinuity occurs in the calculations of $\mathbf{r}(\mathbf{x})$ and $\mathbf{J}$ because of the upwind differencing used by TOUGH2 in Equation (9). A situation can exist (and occurs surprisingly often) where the flow direction for each phase, calculated by Equation (10), can be from a block where the phase is present to one where it is not. From Equation (9) the upwind differencing means that a non-zero relative permeability will be calculated at the interface between the blocks and subsequently a non-zero flux will be calculated for the phase across that interface using Equation (8). Because Equation (8) also contains a contribution from gravity and the phase density, if the value of the density of the phase is discontinuous then the calculated flux is also discontinuous.

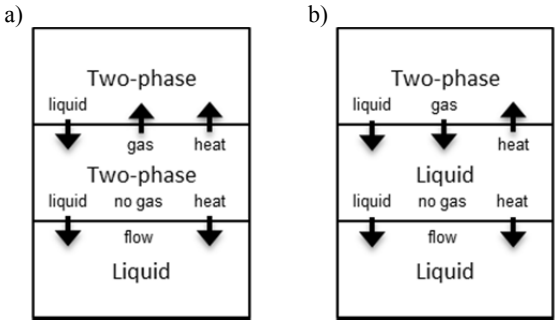

Figure 3: The two states of the problem blocks during a stalled air/water simulation of Lihir Island geothermal system.

For the initial state in plot (a) of Figure 3 both the top block and the problem block contain both phases, so Equation (11) correctly calculates the density of each phase, and Equation (10) calculates the gas phase flow direction correctly. However, as the unsaturated interface moves upwards, eventually the gas phase in the problem block disappears. At this point, in TOUGH2 (and the current version of AUTOUGH2) the EOS3 subroutine calculates the value of the gas density in the problem block as:

$$\rho_g = 0 \tag{17}$$

Plot (b) in Figure 2 show clearly the discontinuity in the gas density calculated in the problem block as a result of using Equation (17) when the gas saturation drops to zero as shown in plot (a). Plots (c) and (d) show the effect of the discontinuities in the subsequent calculations of the pressure in the problem block and the block beneath. It is not difficult to see why the NR method stalls when attempting to calculate derivatives of discontinuous functions such as these.

To ensure the value of a phase density is continuous prior to a phase evolving or after it disappears, a new approach has been developed where it is calculated in the same way as it would be for the two-phase mixture.

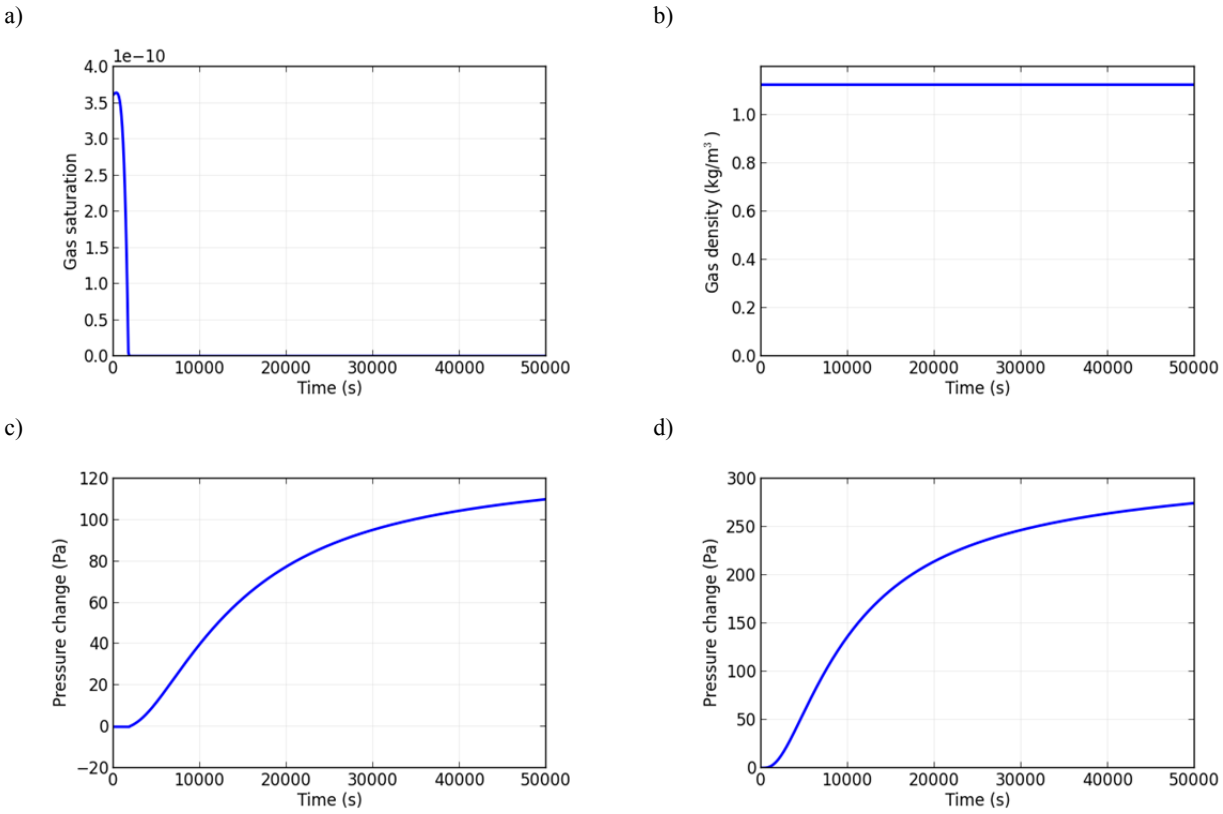

Figure 4: Results for (a) gas saturation, (b) gas density and (c) pressure change in the problem block in a corrected EOS3 simulation. The results for the pressure change in the block beneath the problem block are shown in (d).

For EOS3 the equations are given below:

$$\rho_g = \rho_{vapour} + \rho_{air} \quad (18)$$

$$\rho_l = \rho_{water} \quad (19)$$

The values of ρ_{vapour} and ρ_{water} are calculated from the steam tables using the saturation values for the pressure in the block. This is consistent with the correction already applied in the current version of AUTOUGH2 for the pure water equation of state EOS1.

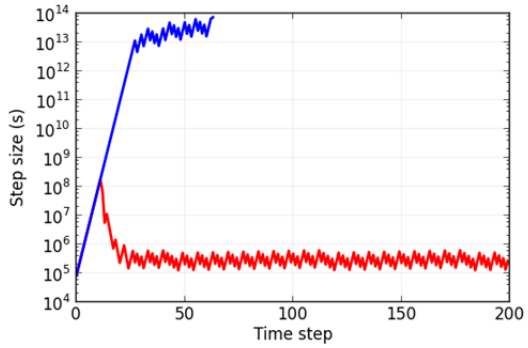

Figure 5: Time step size for two simulations of the Lihir Island system. The stalled simulation results are shown in red (-) and the corrected simulation results in blue (-).

When this approach is applied to the stalled simulation of the Lihir island system, the simulation immediately begins increasing the time step and progresses rapidly to the steady state solution. A comparison of the time step sizes for the stalled simulation and the simulation using the corrected approach is shown in Figure 5.

The results for the gas saturation, gas density and pressure changes for the same blocks in the corrected simulation are shown in Figure 4. Plot (a) shows that the gas saturation in the problem block drops to zero and the gas phase disappears as the unsaturated interface moves upwards. However, for the corrected solution method the gas density, shown in plot (b), remains continuous across the phase change. As a result the block remains single phase liquid, and the pressure in the block itself and the one beneath both increase smoothly as seen in plots (c) and (d).

4. CASE TWO: A MOVING BOILING ZONE

The second example of a simulation that has stalled due to a moving unsaturated interface is taken from a model of the Wayang Windu geothermal system in Indonesia. The model is a CO_2/water model using EOS2. The model grid follows the surface topography and uses CO_2 atmosphere blocks. It consists of 22091 blocks and covers a physical area of 121 km^2.

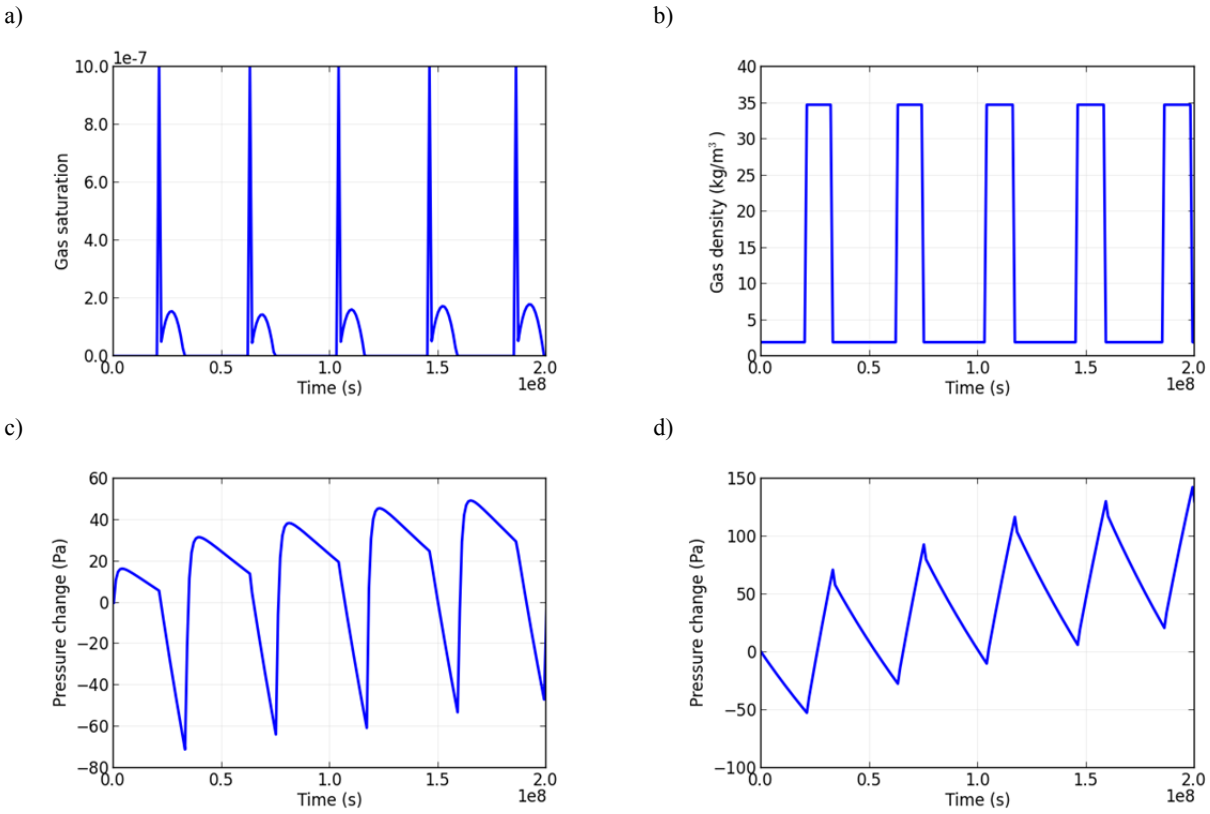

Figure 6: Results for (a) gas saturation, (b) gas density and (c) pressure change in the problem block in a stalled EOS2 simulation. The results for the pressure change in the block beneath the problem block are shown in (d).

Figure 1 shows that this simulation has stalled at a time step of approximately $4x10^6$ seconds. In this case the simulation output revealed that the block causing the stall was in the 4th layer of the model, three layers below the surface. The block is at single phase liquid conditions, as is the block above, whereas in the block below boiling is occurring and two phases are present. Information from the connection table shows that in fact counter-flow is taking place in these blocks, with liquid flowing downwards, and both gas and heat flowing upwards. Figure 7 shows the state of the three blocks and the flow between them once the simulation stalls. Unlike the previous example, none of the flow directions change as a result of the evolution of the gas phase in the problem block.

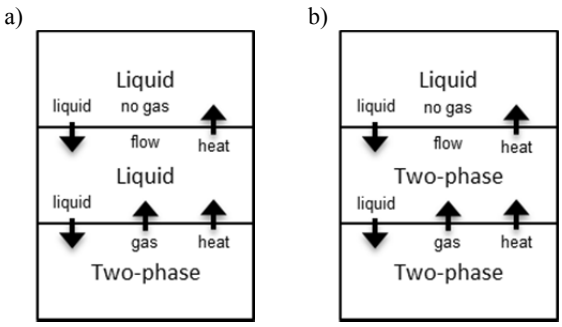

Figure 7: The two states of the problem blocks during a stalled CO_2/water simulation of Wayang Windu geothermal system.

Applying the technique described in Section 2.4 allows the primary variables and secondary parameters to be studied in more detail and the cause of the stall to be identified. The key results are plotted in Figure 6. Plot (a) shows that the gas saturation in the problem block remains very small once the gas phase has evolved before dropping to zero after a short time. After approximately $0.25x10^8$ seconds the gas phase evolves in the problem block again and the stalling cycle continues.

From Figure 7 it can be seen that regardless of the phase state of the problem block, the gas phase flows into it from the block below. This means that in this case, the situation also arises in which the gas phase flows into a block with zero gas phase saturation. However it is clear from plot (b) in Figure 6 that for EOS2 the gas density in a block with zero gas phase saturation is not set to zero, as it is in Equation (17) for EOS3. Inspecting the AUTOUGH2 code reveals that EOS2 has been corrected to include the density of the water vapour (steam) but not the density of the CO_2 gas as shown in Equation (19).

$$\rho_g = \rho_{vapour} \tag{19}$$

Equation (20) gives the equation for the gas density in a two-phase block, and it is this discrepancy between the two that causes the discontinuity shown in plot (b).

$$\rho_g = \rho_{vapour} + \rho_{CO2} \tag{20}$$

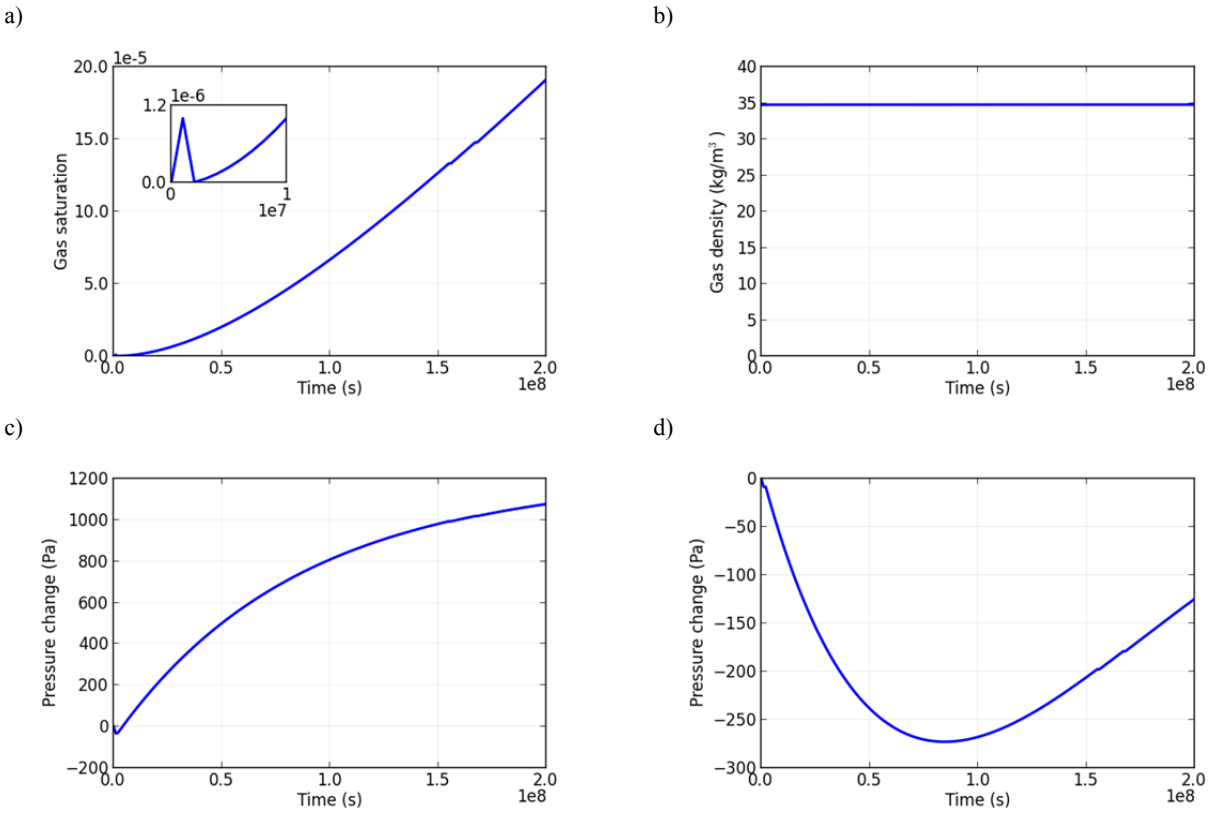

Figure 8: Results for (a) gas saturation, (b) gas density and (c) pressure change in the problem block in a corrected EOS2 simulation. The results for the pressure change in the block beneath the problem block are shown in (d).

For this example the density of the CO_2 gas is more than ten times that of the water vapour, and by disregarding its contribution to the total gas density, the dynamics of the flow are significantly affected. This can be observed in plots (c) and (d) of the pressure change in the problem block and the block beneath it. Sharp discontinuities exist in both and the oscillatory behaviour makes it impossible for the NR method to converge for larger time steps.

By applying Equation (20) to calculate the gas density in blocks where the gas saturation is zero, a corrected version of EOS2 was obtained that ensures the continuity of the gas density. As with the EOS3 correction the saturation pressure is used to calculate the values of ρ_{vapour} and ρ_{CO2}.

Figure (9) shows the results for the same simulation of the Wayang Windu system using the corrected gas density calculation. The stalled behaviour immediately ceases and the time step gradually increases. At several points in the simulation the time step decreases as the flow adjusts towards a final steady state. After 120 time steps the time step begins to increase rapidly and a final time step size of approximately $5x10^{13}$ seconds is achieved before the simulation reaches the desired total time.

Plots of the same parameters for the problem block in the corrected simulation are shown in Figure (8). During the first two time steps the model adjusts to the corrected algorithm and the gas phase evolves then immediately disappears, before evolving again. This corresponds to a short dip in the pressure in the same block before it begins increasing steadily. As expected, plot (b) shows that the gas density is constant across the changes in gas saturation. Once the model has adjusted to the corrected algorithm, the gas saturation in the problem block rises steadily, and plots (c) and (d) show the pressure is well behaved in both blocks.

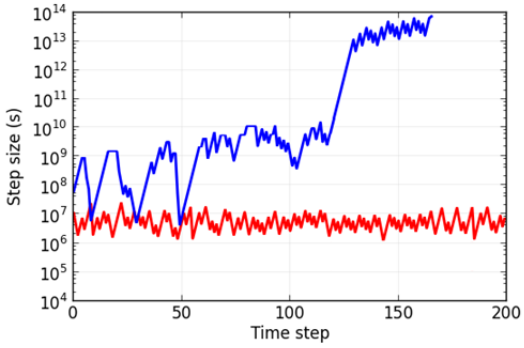

Figure 9: Time step size for two simulations of the Wayang Windu system. The stalled simulation results are shown in red (-) and the corrected simulation results in blue (-).

5. CASE THREE: APPROACHING STEADY STATE

Since corrections to the density calculations in AUTOUGH2's EOS2 and EOS3 have been made, many models have been tested, and no longer exhibit the stalling behaviour during the main part of the simulation. However, as simulations approach steady state they often still display the same stalling behaviour, although at much larger time steps. The results for the corrected simulation in Figure (4) show stalling behaviour developing at time steps of approximately $2x10^{13}$ seconds. For most simulations a steady state is deemed to be obtained once a total time of $1x10^{15}$ seconds is reached, meaning that less than 100 time steps were required for the corrected simulations in Section 3.

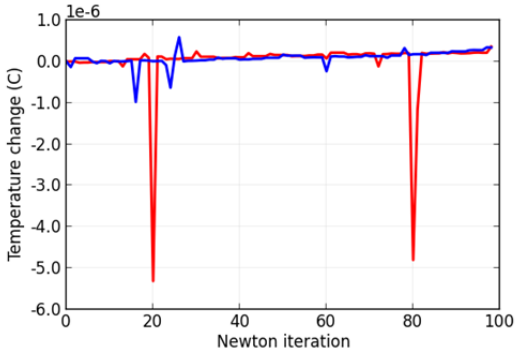

Figure 10: Temperature change in a problem block for two simulations of the Ohaaki system. The stalled simulation results are shown in red (-) and the corrected simulation results in blue (-).

In some models the time step at which the stalling behaviour develops can be much smaller, hence requiring hundreds or thousands of time steps to reach a steady state. An example of such a model is presented in this section. The model is of the Ohaaki geothermal system in New Zealand and is a CO_2/water model with a grid following the surface topography. The model consists of 43012 blocks and covers a physical area of 240 km^2

The output files for the Ohaaki simulation showed that, as in the previous examples, the stalling behaviour was caused by one particular block. However, applying the analysis technique described in Section 2, it was found that no phase changes were occurring in the problem block or any of its neighbours. Also, the block was in layer 26, deep in the model, and so not exposed to a moving water table. Initial investigations showed that the primary variables and secondary parameters appeared to be continuous. Closer inspection revealed that the temperature in the problem block was indeed discontinuous, though on a scale much smaller than its magnitude.

The discontinuities appear as spikes in the plot of temperature change versus NR iteration shown in Figure (10). While the magnitude of these spikes is very small compared to the temperature in the block (which was 224.5°C), they still had a significant effect on the NR solution. This is because from Equation (5) the energy flux includes a temperature gradient term. One of the neighbouring blocks had a temperature of 224.4°C, which meant that the gradient between the blocks was small, and hence could be affected by the discontinuity. The behaviour did not appear until close to the steady state because the error was small until the time step multiplier from Equation (7) was sufficient to cause the NR method to fail to converge.

Once the discontinuous temperature had been identified it was a relatively simple process to debug the AUTOUGH2 code to identify the problem. The problem block was two-phase in the deep boiling zone of the model. This means that at each NR iteration the temperature of the block is set to the saturation temperature, based on the block pressure. The saturation temperature calculation involves a small internal one-dimensional Newton-Raphson solve. This 1D NR method uses a previous estimate of the temperature as its initial guess and in some circumstances it is possible that this initial guess returns a saturation pressure already within the convergence tolerance for the method. These circumstances occur most commonly when the simulation is close to steady state and the temperatures and pressures are changing by very small amounts.

Even though the changes in pressure and temperature are very small, when the 1D NR solve completes without a single iteration it arrives at a temperature not exactly consistent with the temperatures it calculates after a number of iterations. These show up as the spikes in Figure (10) and for this model they occur every 60 iterations of the full NR method, which equates to 12 time steps or four cycles of stalled time step behaviour.

One option for correcting this problem is to reduce the convergence tolerance for the 1D NR method. This was deemed to be somewhat arbitrary, and it was thought that this would simply delay the problem until smaller changes in temperature and pressure were encountered. Instead the 1D NR method was forced to always complete one iteration regardless of the initial convergence calculation. This is consistent with other 1D NR solves that occur throughout AUTOUGH2.

The resulting temperature change for the problem block is shown in blue in Figure (10), and a plot of the time step size for both the stalled and corrected simulations is shown in Figure (11). Obviously this correction also only delays the onset of stalling behaviour rather than completely eliminating it, as the plot for the corrected simulation stalls again at a time step of approximately $4x10^{13}$ seconds compared to $7x10^{12}$ seconds for the stalled simulation. This means that for this example the model reached a steady state 4-5 times faster, which is a significant improvement.

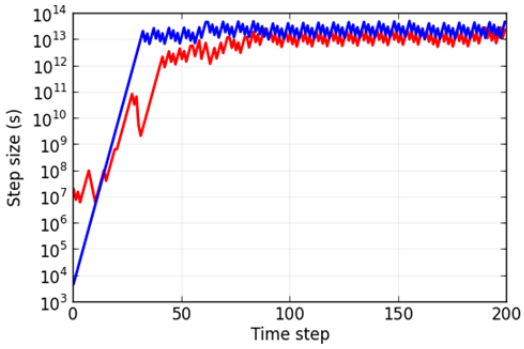

Figure 11: Time step size for two simulations of the Ohaaki system. The stalled simulation results are shown in red (-) and the corrected simulation results in blue (-)

Investigating the stalling behaviour in the corrected simulation revealed that discontinuities were now occurring in several variables and in several blocks. This was found to be due to limitations of the accuracy with which the linear equation solver could solve Equation (15). At large time steps the Jacobian matrix becomes extremely ill-conditioned and the current linear solvers are unable to achieve the desired solution tolerance within a practical computational time frame. Solutions to this problem are an area of current research.

5. CONCLUSION

A method for investigating stalled TOUGH2 simulations has been developed and used to examine three different examples of the problem. An explanation of why each simulation has stalled has been given with reference to the solution algorithm and governing equations. Corrections to the calculations for the phase densities in EOS2 and EOS3 have been suggested as well as a correction to the algorithm for calculating the saturation temperature. As a result of these corrections, simulations no longer stall at the early stages and large time step sizes can be achieved without any manual intervention.

These improvements greatly reduce the computational time for most simulations, remove the need for manual intervention and make inverse modelling of complex geothermal systems more tractable.

ACKNOWLEDGEMENTS

The authors would like to thank George Zyvoloski for the invaluable discussions at various stages of this project.

REFERENCES

Burnell, J., Clearwater, E., Croucher, A., Kissling, W., O'Sullivan, J.P., O'Sullivan, M.J. & Yeh, A. (2012), "Future directions in geothermal modelling," Proceedings (electronic) 34rd New Zealand Geothermal Workshop, University of Auckland, Auckland, New Zealand, 19-21 November, 2012.

Doherty, J. (2010), "PEST: model-independent parameter estimation. User manual, 5th edition." Watermark Numerical Computing, Corinda, Australia, http://www.sspa.com/pest.

Noy, D., Holloway, S., Chadwick, R., Williams, J., Hannis, S. & Lahann, R. (2012), "Modelling large-scale carbon dioxide injection into the Bunter Sandstone in the UK Southern North Sea," International Journal of Greenhouse Gas Control, 9, 220-233.

Omagbon, J.B. & O'Sullivan, M.J. (2011), "Use of an heuristic method and PEST for calibration of geothermal models," Proceedings (electronic) 33rd New Zealand Geothermal Workshop, University of Auckland, Auckland, New Zealand, 21-23 November, 2011.

O'Sullivan, J., Dempsey, D., Croucher, A., Yeh, A., & O'Sullivan, M. (2013), "Controlling complex geothermal simulations using PyTOUGH," Proceedings Thirty-Eighth Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, California, February 11-13, 2013.

O'Sullivan, J.P., Croucher, A.E., O'Sullivan, M.J. Stevens, L. & Esberto, M. (2011), "Modelling the evolution of a mine pit in a geothermal field at Lihir Island, Papua, New Guinea," Proceedings (electronic) 33rd New Zealand Geothermal Workshop, University of Auckland, Auckland, New Zealand, 21-23 November, 2011.

O'Sullivan, M. J., Pruess, K., & Lippmann, M. J. (2001), "State of the art of geothermal reservoir simulation," Geothermics, 30(4), 395-429.

O'Sullivan, M. J. (2012). "Geothermal reservoir modelling theory," Lecture delivered at University of Auckland, Auckland, New Zealand, unpublished.

Pruess, K., Oldenburg, K. & Moridis, G. (1999), "TOUGH2 user's guide, version 2.0," Lawrence Berkeley National Laboratory, University of California, Berkeley.

Yeh, A., Croucher, A. & O'Sullivan, M.J. (2012), "Recent developments in the AUTOUGH2 simulator", Proceedings TOUGH Symposium 2012, Berkeley, California, September 17-19, 2012.