Combined Natural State and History Matching Using the Adjoint or Direct Sensitivity Method

Elvar K. Bjarkason[1], John P. O'Sullivan[1], Angus Yeh[1] and Michael J. O'Sullivan[1]

[1] Department of Engineering Science, University of Auckland, Auckland, New Zealand

ebja558@aucklanduni.ac.nz

Keywords: *inversion, history matching, adjoint method, sensitivities, PEST, natural state, TOUGH2.*

ABSTRACT

Gradient based inversion of high-enthalpy geothermal reservoir models is a demanding task even for just a handful of model parameters. Moreover, as more parameters are included the inverse process becomes slower and parameter uniqueness is a concern. These issues can, nevertheless, be lessened by using appropriate mathematical tools to obtain geologically sound and well calibrated models within a reasonable amount of time.

For highly parameterized models, parameter reasonableness can be imposed by choosing an appropriate regularization scheme for the inversion. Furthermore, the derivatives of model outputs needed for gradient based inversion can be found accurately and at a low computational cost using either the adjoint or direct sensitivity methods. Using a synthetic vertical slice model, we demonstrate these methods for the inverse problem of simultaneously matching downhole natural state temperatures along with history matching of production pressure and enthalpy data. The adjoint and direct methods were found to be reliable and computationally faster than the finite differencing approaches predominantly used in geothermal modelling.

1. INTRODUCTION

For the purpose of automatically adjusting model parameters to match observations, the performance of gradient based inverse methods count on the integrity of sensitivities, which are derivatives of model outputs with respect to adjustable model parameters, to guide the way. In the application of current inverse tools such as PEST (Doherty, 2016) or iTOUGH (Finsterle, 2007) to geothermal reservoir modelling, model adjustment decisions are based on sensitivities evaluated using finite difference approximation.

Because of the inevitable differencing errors it is difficult with the finite differencing approach to produce very accurate sensitivity values. Another drawback of the finite difference approach is that it scales badly as the number of model parameters increases. One update of a vector $\boldsymbol{m}$ of N_m adjustable model parameters using either PEST or iTOUGH currently requires at the very least N_m+1 computationally expensive nonlinear reservoir simulations.

The adjoint (Li *et al.*, 2003) and direct (Anterion *et al.*, 1989) methods are computationally more economical alternatives for evaluating model sensitivities. Both methods have been developed and found favour in subsurface modelling communities outside geothermal modelling. The adjoint method has especially found traction in hydrology (Medina & Carrera, 2003) and petroleum reservoir modelling (Li *et al.*, 2003; Oliver & Chen, 2011; Oliver *et al.*, 2008), because it scales with the number of model outputs. The direct method, however, scales with the number of parameters.

Bjarkason *et al.* (2016) demonstrated the benefits of using the adjoint or direct approaches to speed up inversions of natural state geothermal reservoir models. Notably, when dealing with natural state data, Bjarkason *et al.* (2016) show that the computational burden of evaluating model sensitivities during an inversion can be reduced by a factor equalling the number of natural state simulation time-steps. Their sample problems showed results for temperature observations and adjustable model permeabilities, though the methods are not limited to these types of observation or parameters.

The present work develops the adjoint and direct methods further for geothermal reservoir simulations using the TOUGH2 simulator (Pruess *et al.*, 1999) by considering the joint inversion of both natural state and production data. Included are results for sample synthetic inverse problems, which include a variety of common model parameters and observations.

2. RESERVOIR SIMULATIONS

The nonlinear mass and energy transport equations solved at the k^{th} simulation time-step by TOUGH2 can be expressed by the following vector equation:

$$\boldsymbol{f}^k(\boldsymbol{u}^k, \boldsymbol{u}^{k-1}, \boldsymbol{m}, \Delta t^k) = \boldsymbol{0} \tag{1}$$

The superscript k indicates values at the k^{th} time-step Δt^k, $\boldsymbol{m}$ is the vector of adjustable model parameters and $\boldsymbol{u}$ is a vector of the simulation primary variables, such as pressure and temperature or vapour saturation, which the simulator needs to update between time-steps.

As outlined by Bjarkason *et al.* (2015), the forward residual can be written as

$$\boldsymbol{f}^k = \frac{1}{\Delta t^k}[\boldsymbol{M}(\boldsymbol{u}^k, \boldsymbol{m}) - \boldsymbol{M}(\boldsymbol{u}^{k-1}, \boldsymbol{m})] + \boldsymbol{FS}(\boldsymbol{u}^k, \boldsymbol{m}) \tag{2}$$

The forward residual in (2) consists of an accumulation part denoted by the two $\boldsymbol{M}$'s as well as a flux-source term $\boldsymbol{FS}$.

TOUGH2 solves (1) by Newton iteration. Using p to denote the Newton iteration index the primary variables are updated according to

$$\boldsymbol{A}^{k,p-1}\,(\boldsymbol{u}^{k,p} - \boldsymbol{u}^{k,p-1}) = -\boldsymbol{f}^k(\boldsymbol{u}^{k,p-1}, \boldsymbol{u}^{k-1}, \boldsymbol{m}) \tag{3}$$

where

$$\boldsymbol{A}^{k,p-1} = \left[\frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{u}^k}\right]^{p-1} \tag{4}$$

The Newton process is initialized at every time-step with $\boldsymbol{u}^{k,0}=\boldsymbol{u}^{k-1}$ and terminated when the convergence tolerances are met, as described in Pruess *et al*. (1999). However, if this is not achieved within a maximum number of iterations, then the time-step is decreased and solving (1) is retried. The simulation time-step is increased when a solution is found within a set number of iterations.

2.1 Natural State

The pre-exploitation natural state of a high-enthalpy geothermal reservoir is normally simulated by a transient approach to steady state (O'Sullivan *et al*., 2001). A natural state simulation involves solving a sequence of transient equations (1) while gradually increasing the time-step size. For a natural state simulation the aim is to solve the steady state version of (1):

$$\boldsymbol{f}_{\mathrm{st}} = \boldsymbol{f}_{\mathrm{st}}(\boldsymbol{u}_{\mathrm{st}}, \boldsymbol{m}) = \boldsymbol{FS}(\boldsymbol{u}_{\mathrm{st}}, \boldsymbol{m}) = \boldsymbol{0} \qquad (5)$$

to an acceptable level of accuracy. The natural state solution $\boldsymbol{u}_{\mathrm{st}}$ is the vector of primary variables at the final simulation time of the transient approach to steady state. The final natural state gives initial conditions for the production simulation that may follow.

2.2 Production Phase

The simulation of a production history proceeds by solving the transient problem (1) for a sequence of time-steps until the desired final time is reached. For the solution of the first time-step the initial condition $\boldsymbol{u}^0 = \boldsymbol{u}_{\mathrm{st}}$ is applied.

3. INVERSION

Inverse problems are generally ill-posed and therefore require regularization (Engl *et al*., 1996). For finding reasonable model parameters we choose to minimize a regularized least-squares objective function Φ. The objective function contains a measurement misfit Φ_d and model misfit Φ_m as follows

$$\Phi(\boldsymbol{m}) = \Phi_d(\boldsymbol{m}) + \beta \cdot \Phi_m(\boldsymbol{m}) \qquad (6)$$

where

$$\begin{aligned}\Phi_d(\boldsymbol{m}) &= [\boldsymbol{d}(\boldsymbol{m}) - \boldsymbol{d}^*]^T \boldsymbol{C}_D^{-1} [\boldsymbol{d}(\boldsymbol{m}) - \boldsymbol{d}^*] \\ &= \boldsymbol{r}(\boldsymbol{m})^T \boldsymbol{C}_D^{-1} \boldsymbol{r}(\boldsymbol{m})\end{aligned} \qquad (7)$$

The last term in (6) is a regularization term, which includes a positive regularization weight β. The observation mismatch given in (7) depends on the difference between the vector of N_d observed values $\boldsymbol{d}^*$ and their corresponding simulated values $\boldsymbol{d}(\boldsymbol{m})$, or the observation residual vector $\boldsymbol{r}(\boldsymbol{m}) = \boldsymbol{d}(\boldsymbol{m}) - \boldsymbol{d}^*$. $\boldsymbol{d}(\boldsymbol{m})$ are model outputs that need to be matched to the data. For the natural state problem the data are usually downhole temperatures and for history matching they are usually downhole pressures and production enthalpies. The matrix $\boldsymbol{C}_D$ is the observation covariance matrix, which we assume is diagonal and its diagonal terms are the observation error variances.

3.1 PEST

For the inversions we use a serial version of PEST (Doherty, 2016). PEST uses a variant of the Levenberg-Marquardt scheme (Levenberg, 1944; Marquardt, 1963) to minimize (6). The main reasons for choosing PEST are: (i) the ease of including model regularization and (ii) that PEST includes the option of supplying externally evaluated model derivatives.

For PEST the regularization term can be presented similarly to the observation mismatch as a weighted dot product of regularization residuals $\boldsymbol{v}(\boldsymbol{m})$ of length N_r:

$$\Phi_m(\boldsymbol{m}) = \boldsymbol{v}(\boldsymbol{m})^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{v}(\boldsymbol{m}) \qquad (8)$$

$\boldsymbol{W}$ is a weighting matrix. The regularization residual encapsulates preferred model parameter conditions, which should only be deviated from if, for a good model, the measurement misfit is incompatible with the preferred parameter conditions.

The i^{th} element of the vector v_i can for instance be chosen as the difference between a model parameter m_j and a reference value, for example, a modeller's best guess for m_j. Another option is to choose v_i as the difference between two parameter values m_j and m_l, to achieve some smoothing of the parameter in space.

For minimizing (6), PEST iteratively updates the model parameters by solving a Levenberg-Marquardt update equation given by

$$\begin{aligned}&[\boldsymbol{S}^T \boldsymbol{C}_D^{-1} \boldsymbol{S} + \beta \boldsymbol{V}^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{V} + \lambda \boldsymbol{I}] \delta \boldsymbol{m} \\ &\quad = -\boldsymbol{S}^T \boldsymbol{C}_D^{-1} \boldsymbol{r} - \beta \boldsymbol{V}^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{v}\end{aligned} \qquad (9)$$

The λ multiplying the identity matrix is the adjustable Levenberg-Marquardt damping factor. The matrix $\boldsymbol{V}$ is the regularization Jacobian defined as

$$\boldsymbol{V} = \frac{d\boldsymbol{v}(\boldsymbol{m})}{d\boldsymbol{m}} = \begin{bmatrix} \frac{dv_1}{dm_1} & \cdots & \frac{dv_1}{dm_{N_m}} \\ \vdots & \ddots & \vdots \\ \frac{dv_{N_r}}{dm_1} & \cdots & \frac{dv_{N_r}}{dm_{N_m}} \end{bmatrix} \qquad (10)$$

The matrix $\boldsymbol{S}$ is the sensitivity matrix:

$$\boldsymbol{S} = \frac{d[\boldsymbol{d}(\boldsymbol{m})]}{d\boldsymbol{m}} = \frac{d\boldsymbol{r}(\boldsymbol{m})}{d\boldsymbol{m}} = \begin{bmatrix} \frac{dr_1}{dm_1} & \cdots & \frac{dr_1}{dm_{N_m}} \\ \vdots & \ddots & \vdots \\ \frac{dr_{N_d}}{dm_1} & \cdots & \frac{dr_{N_d}}{dm_{N_m}} \end{bmatrix} \qquad (11)$$

For full details on using PEST the reader is referred to PEST's user manual (Doherty, 2016). The main steps PEST takes during the natural state and history matching are:

(i) Begin by running the natural state and production simulations to find $\Phi(\boldsymbol{m})$ for the initial guess $\boldsymbol{m}$.
(ii) Evaluate the Jacobian matrices $\boldsymbol{S}$ and $\boldsymbol{V}$.
(iii) Adjust the regularization weight β. This step is described by Doherty (2003).
(iv) Solve (9) to find an update $\Delta\boldsymbol{m} = \alpha\delta\boldsymbol{m}$, where $\alpha = -\frac{\boldsymbol{r}^T \boldsymbol{C}_D^{-1} \boldsymbol{S} \delta \boldsymbol{m} + \beta \boldsymbol{v}^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{V} \delta \boldsymbol{m}}{\delta \boldsymbol{m}^T \boldsymbol{S}^T \boldsymbol{C}_D^{-1} \boldsymbol{S} \delta \boldsymbol{m} + \beta \delta \boldsymbol{m}^T \boldsymbol{V}^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{V} \delta \boldsymbol{m}}$ (Doherty, 2015).
(v) Run natural state and production simulations using $\boldsymbol{m} + \Delta\boldsymbol{m}$ to find $\Phi(\boldsymbol{m} + \Delta\boldsymbol{m})$.
(vi) Update λ.
(vii) PEST may repeat steps (iv) and (v) using a new λ if $\Phi(\boldsymbol{m} + \Delta\boldsymbol{m}) / \Phi(\boldsymbol{m})$ > PHIRATSUF, where PHIRATSUF is a PEST input parameter. Otherwise PEST finds the best $\Delta\boldsymbol{m}$ from the current

iteration, sets $\boldsymbol{m} = \boldsymbol{m} + \Delta\boldsymbol{m}$ as the most up to date parameters and a new iteration is entered starting with step (ii).

We chose to halt PEST inversions when the observation mismatch Φ_d fell below a target value Φ_t, though other stopping criteria may be chosen as well. PHIRATSUF was set to 0.7 for our runs using PEST. The elements of the matrix $\boldsymbol{V}$ were evaluated by either analytical means or finite differences. In either case the cost of forming $\boldsymbol{V}$ was negligible since it did not require any nonlinear simulations.

Within the scope of this paper, the computationally most expensive tasks during an inversion are running the nonlinear forward simulations and evaluating the sensitivity matrix $\boldsymbol{S}$. The following subsections discuss methods for finding $\boldsymbol{S}$.

3.2 Finite Differencing

When PEST calculates the sensitivity matrix, model simulations are run using perturbed parameter values and derivatives of model outputs with respect to parameters are found by finite differencing.

The least computationally expensive of the finite difference methods is forward finite differencing, whereby the elements of $\boldsymbol{S}$ are approximated by:

$$S_{ij} = \frac{dr_i}{dm_j} \approx \frac{r_i\left(\boldsymbol{m} + \Delta m_j \boldsymbol{e}_j\right) - r_i(\boldsymbol{m})}{\Delta m_j} \qquad (12)$$

Here Δm_j is the value the j^{th} parameter is perturbed by and $\boldsymbol{e}_j$, denotes the j^{th} unit vector.

Finding the elements of $\boldsymbol{S}$ by equation (12) requires N_m+1 time-consuming nonlinear forwards simulations, which leads to long inversion times. Furthermore, how a good choice for Δm_j should be made is unclear (Bjarkason *et al.*, 2015). Trial and error may be needed to find perturbation values that work well.

More accurate estimates of model sensitivities can be found using higher order finite differencing schemes. For instance central differencing gives

$$S_{ij} = \frac{dr_i}{dm_j} \approx \frac{r_i\left(\boldsymbol{m} + \Delta m_j \boldsymbol{e}_j\right) - r_i\left(\boldsymbol{m} - \Delta m_j \boldsymbol{e}_j\right)}{2\Delta m_j} \qquad (13)$$

The truncation error of (13) is lower than the approximation in (12). However, the greater accuracy offered by higher-order schemes comes with the extra expense of having to run further simulations. In the case of central differencing the cost is $2N_m$ nonlinear simulations for the full sensitivity matrix.

3.3 Direct Method

Compared with the finite differencing approach the direct method (Anterion *et al.*, 1989) can provide a more efficient and reliable way to find $\boldsymbol{S}$, though it still scales with the number of adjustable model parameters. As demonstrated by Bjarkason *et al.* (2016), the direct method can be used to evaluate the natural state primary variable sensitivities $\partial\boldsymbol{u}_{\text{st}}/\partial\boldsymbol{m}$ by solving

$$\boldsymbol{A}_{\text{st}} \left[\frac{\partial \boldsymbol{u}_{\text{st}}}{\partial \boldsymbol{m}}\right] = -\boldsymbol{G}_{\text{st}} \qquad (14)$$

where

$$\boldsymbol{A}_{\text{st}} = \frac{\partial \boldsymbol{f}_{\text{st}}}{\partial \boldsymbol{u}_{\text{st}}} \;\; ; \;\; \boldsymbol{G}_{\text{st}} = \frac{\partial \boldsymbol{f}_{\text{st}}}{\partial \boldsymbol{m}} \qquad (15)$$

Equation (14) is a linear matrix equation with N_m right-hand sides. The matrix $\boldsymbol{A}_{\text{st}}$ is readily evaluated from the calculation of the forward Jacobian (4) at the end of the final natural state time-step (Bjarkason *et al.*, 2016). The elements of the matrix $\boldsymbol{G}_{\text{st}}$ can be found by analytical differentiation for the most common parameters, such as permeabilities, porosities and boundary sources.

Primary variable sensitivities can also be evaluated during the transient production simulation based on the direct method presented in Bjarkason *et al.* (2015). The transient direct approach finds the primary variable sensitivities by tracking forward in time, starting with the first production time-step. However, the simulations considered by Bjarkason *et al.* (2015) had fixed initial conditions independent of $\boldsymbol{m}$. Since the initial state for the production run is determined by a natural state, which can vary with $\boldsymbol{m}$, we need to account for that when solving for the first time-step. Using the direct method we solve for the first production time-step (k = 1):

$$\boldsymbol{A}^1 \left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^1 = -\boldsymbol{G}^1 - \boldsymbol{B}^1 \left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^0 \qquad (16)$$

where

$$\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^0 = \left[\frac{\partial \boldsymbol{u}_{\text{st}}}{\partial \boldsymbol{m}}\right] \;\; ; \;\; \boldsymbol{u}^0 = \boldsymbol{u}_{\text{st}} \qquad (17)$$

For the time-steps that follow (k > 1), the primary variable sensitivities $[\partial\boldsymbol{u}/\partial\boldsymbol{m}]^k$ are found by solving

$$\boldsymbol{A}^k \left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^k = -\boldsymbol{G}^k - \boldsymbol{B}^k \left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^{k-1} \qquad (18)$$

Note that

$$\boldsymbol{A}^k = \frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{u}^k} \;\; ; \;\; \boldsymbol{B}^k = \frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{u}^{k-1}} \;\; ; \;\; \boldsymbol{G}^k = \frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{m}} \qquad (19)$$

The three matrices above can be extracted from the forward simulation code without too much extra effort (Bjarkason *et al.*, 2015). The size of the individual linear matrix problems in equations (16) and (18) is the same as the one in (14). The cost of solving (16) or (18) is therefore the same as solving (14). More details on the construction of the matrices $\boldsymbol{G}^k$ and $\boldsymbol{G}_{\text{st}}$ are given in Appendix A.

With equations (14), (16) and (18) for all k > 1 solved, the sensitivity matrix can be found according to

$$\boldsymbol{S} = \frac{d\boldsymbol{r}}{d\boldsymbol{m}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}} + \boldsymbol{C}_{\text{st}} \frac{\partial \boldsymbol{u}_{\text{st}}}{\partial \boldsymbol{m}} + \sum_{k=1}^{N_t^{\text{pr}}} \boldsymbol{C}^k \left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^k \qquad (20)$$

where

$$\boldsymbol{C}_{\text{st}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}_{\text{st}}} \;\; ; \;\; \boldsymbol{C}^k = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}^k} \qquad (21)$$

Here N_t^{pr} denotes the number of production simulation time-steps. Evaluation of $\boldsymbol{C}^k$ and $\boldsymbol{C}_{\text{st}}$ is discussed in Appendix B. The term $\partial\boldsymbol{r}/\partial\boldsymbol{m}$ is only nonzero when there is a simulated observation which depends explicitly on a model parameter.

We expect the direct method to outperform the finite differencing methodology. This can be understood by considering the number of linear solves required by the two differing approaches, as the linear solves are the most time-consuming part for either approach.

Let N_t^{st} be the average number of time-steps needed to reach steady state and N_{NR}^{st} be the average number of Newton iterations during a natural state time-step. Forward finite differencing requires solving around $N_{NR}^{\mathrm{st}} N_t^{\mathrm{st}}$ linear problems for each of the required N_m+1 simulations. Similarly, each production simulation requires solving about $N_{NR}^{\mathrm{pr}} N_t^{\mathrm{pr}}$, where N_{NR}^{pr} is the average number of Newton iterations during a production time-step. The expected cost of using finite differencing to form $\boldsymbol{S}$ is therefore $(N_m + 1)\left(N_{NR}^{\mathrm{st}} N_t^{\mathrm{st}} + N_{NR}^{\mathrm{pr}} N_t^{\mathrm{pr}}\right)$ linear solves. Note that during a PEST iteration the simulation using the most up to date parameters $\boldsymbol{m}$ has already been performed prior to evaluating $\boldsymbol{S}$, so the added cost of finding $\boldsymbol{S}$ during an inversion is slightly less or $N_m\left(N_{NR}^{\mathrm{st}} N_t^{\mathrm{st}} + N_{NR}^{\mathrm{pr}} N_t^{\mathrm{pr}}\right)$ linear solves.

The direct method, on the other hand, allows us to find the sensitivity matrix by solving one nonlinear simulation for parameters $\boldsymbol{m}$ at a cost of $N_{NR}^{\mathrm{st}} N_t^{\mathrm{st}} + N_{NR}^{\mathrm{pr}} N_t^{\mathrm{pr}}$ linear solves followed by $N_m\left(1 + N_t^{\mathrm{pr}}\right)$ linear solves. Again, during an inversion, the nonlinear simulation comes at no extra cost for the direct method. The added cost of the direct method is therefore only $N_m\left(1 + N_t^{\mathrm{pr}}\right)$ linear solves.

From the above we expect the direct method to be very efficient compared to finite differencing when the natural state simulation is long compared to the production history simulation. If the number of production time-steps is far more than the number of natural state time-steps, the savings that can be made will be somewhat lessened and depend mostly on the average number of Newton iterations.

3.4 Adjoint Method

When using the adjoint method (Bjarkason *et al*., 2015; Li *et al*., 2003) to find the sensitivity matrix $\boldsymbol{S}$ we track backward in time finding matrices of Lagrange multipliers $\boldsymbol{\Psi}^k$ during each production time-step and end by finding the adjoint variables $\boldsymbol{\Psi}_{\mathrm{st}}$ for the natural state. We begin by solving for the last production time-step a linear problem with N_d right-hand sides:

$$\left[\boldsymbol{A}^{N_t^{\mathrm{pr}}}\right]^T \boldsymbol{\Psi}^{N_t^{\mathrm{pr}}} = -\left[\boldsymbol{C}^{N_t^{\mathrm{pr}}}\right]^T \tag{22}$$

This is followed by solving for each preceding production simulation time:

$$[\boldsymbol{A}^k]^T \boldsymbol{\Psi}^k = -[\boldsymbol{C}^k]^T - [\boldsymbol{B}^{k+1}]^T \boldsymbol{\Psi}^{k+1} \tag{23}$$

Note that for the adjoint method the matrices of the linear equations are the transpose or adjoint of the ones used in the direct scheme. Finally, we solve for the natural state:

$$[\boldsymbol{A}_{st}]^T \boldsymbol{\Psi}_{st} = -[\boldsymbol{C}_{st}]^T - [\boldsymbol{B}^1]^T \boldsymbol{\Psi}^1 \tag{24}$$

The sensitivity matrix is then found according to

$$\boldsymbol{S} = \frac{d\boldsymbol{r}}{d\boldsymbol{m}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}} + [\boldsymbol{\Psi}_{st}]^T \boldsymbol{G}_{st} + \sum_{k=1}^{N_t^{\mathrm{pr}}} [\boldsymbol{\Psi}^k]^T \boldsymbol{G}^k \tag{25}$$

As presented the adjoint method requires the solution of $1 + N_t^{\mathrm{pr}}$ linear matrix equations with N_d right-hand sides. In practice, however, the effective number of right-hand sides in equations (22) and (23) is at most N_d. Essentially the number of right-hand sides in the equations is equal to the number of simulated observations after time-step $k - 1$. The adjoint method should therefore be preferred over the direct one when $N_m \geq N_d$. In some instances the adjoint method may even be faster when there are fewer parameters than observations, especially if most of the observations are from the natural state.

3.5 Running PEST with External Derivatives

As in Bjarkason *et al*. (2016) we used the University of Auckland's version of TOUGH2, called AUTOUGH2 (Yeh *et al*., 2012), as the reservoir simulator. A new AUTOUGH2 executable was created which outputs in HDF5 binary format (The HDF5 Group, 2016) the additional information needed to execute the direct and adjoint sensitivity methods. This new executable was used when running PEST with the sensitivity matrix evaluated externally by the direct or adjoint methods. When using other methods the executable used was one that did not have the extra adjoint/direct output.

Both the adjoint and direct methods were implemented using Python. For evaluating the sensitivity matrix by either the adjoint or direct methods, PEST ran a Python script provided to it. The script finds the most up to date HDF5 output files and forms $\boldsymbol{S}$ using the adjoint or direct method as described earlier. As in Bjarkason *et al*. (2016), we used the SuperLU direct sparse linear solver to solve equations (14, 16, 18, 22, 23, 24). By doing so we take advantage of the fact that for a given time-level the linear problems only differ by their right-hand sides. After working out the sensitivity matrix, the script outputs $\boldsymbol{S}$ for PEST to use.

4. TEST PROBLEMS

The adjoint and direct methods were tested on the 2D vertical slice model used in Bjarkason *et al*., (2016), with some modifications. First a “true” 2D model was constructed and run to generate synthetic natural state and production data. Simulations were run using AUTOUGH2 with the pure water equation of state module, EOS1. The AUTOUGH2 executables used in this study are newly compiled versions which output simulation results as HDF5 binary files, which allows for fast and accurate processing of simulation outcomes.

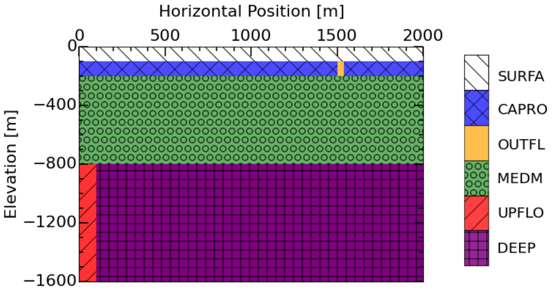

Figure 1: The true vertical slice model, showing the distribution of the six rock-types.

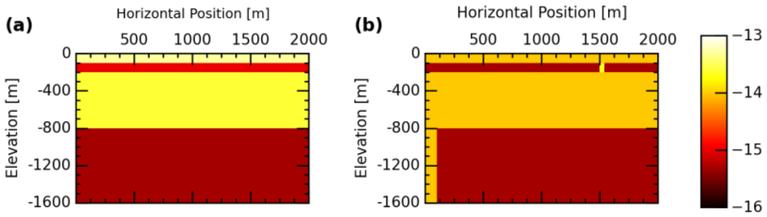

Figure 2: Base ten logarithms of the true formation permeabilities: (a) the horizontal permeability k_x [m^2]; (b) the vertical permeability k_z [m^2].

To showcase the applicability of the methods for high enthalpy systems, the energy flowing into the true system was chosen to be higher than in Bjarkason *et al*. (2016). The enthalpy was high enough so that a two-phase zone is present in the natural state. Moreover, the inversion tests included a wider variety of parameters and observation types than the previous study by Bjarkason *et al*. (2016).

A representation of the true model, which has six rock-types, is depicted in Figure 1. The model grid includes 100 constant boundary blocks along the top of the model for constant boundary conditions of 15°C and 1 atm pressure. The main body of the reservoir is made up of 8,000 equally sized 20 m × 20 m × 20 m blocks. The side boundaries of the model are closed. At the bottom boundary, fluid with an enthalpy of 1,200 kJ/kg is injected at a rate of 0.03 kg/s into each of the five left-most blocks. A constant heat flux of 80 mW/m^2 is applied over the remainder of the bottom boundary.

Figure 2 shows the base ten logarithms of the true horizontal and vertical permeabilities. For all simulations we used harmonic weighting of permeabilities at block interfaces (O'Sullivan *et al*., 2013). The true rock-types had different porosity values as shown in Figure 3. Other parameters did not vary between rock-types. All rock-types had a rock grain density of 2,500 kg/m^3, a thermal conductivity of 2.5 W/(m·K) and a rock specific heat of 1.0 kJ/(kg·K). Simulations used zero capillary pressures and linear relative permeability curves with residual liquid saturations of 0.5 and relative permeabilities summing to 1.

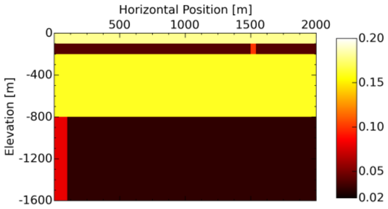

Figure 3: True formation porosity.

Natural state simulations were conducted as described in Bjarkason *et al*. (2016). The simulated production period lasts for three years. For all simulations the initial simulation time-step size was set to one day and the time-step was limited to a maximum of about nine days. Production was simulated for three years, and included three production wells and two injection wells as depicted in Figure 4. Production and injection was maintained at constant rates for each well during the three years. Numbering the production wells in Figure 4 from left to right, Producers 1 and 2 produced 0.5 kg/s each, Producer 3 gave 0.4 kg/s, and 0.56 kg/s of 167.5 kJ/kg enthalpy fluid was injected into each injection well. Every well had one feed zone which coincide with the bottom of the wells, see Figure 4.

Synthetic natural state and production data was generated using the true model. Natural state temperature observations were taken from every third model block intersected by five natural state observation wells, see Figure 4, giving 135 temperature observations in total. The production history observations included the flowing enthalpy and pressure of the three producing feed zones, measured every three months over the three year period. In total there are 36 pressure observations and 36 enthalpy observations. The study, therefore, has N_d = 207 observations in total.

For the final synthetic data used for this study we added zero mean Gaussian noise to all of the observations. The standard deviation of the noise was chosen as 0.5°C for temperature observations, 0.1 bar for pressure observations and 10 kJ/kg for observed enthalpies. These values were assumed to be known when forming $\boldsymbol{C_D}$ for the inversions that followed. The synthetic temperature, enthalpy and pressure observations are shown in Figures 5, 6 and 7, respectively.

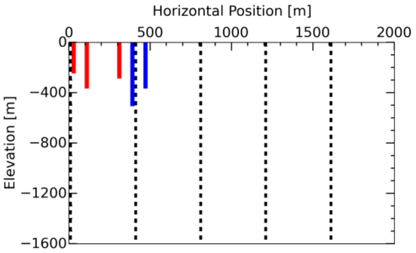

Figure 4: The 3 production wells (red), 2 injection wells (blue) and 5 natural state observation wells (black dashed lines).

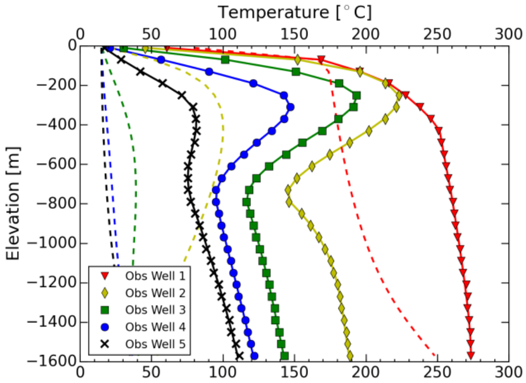

Figure 5: Observed natural state temperatures (markers), matched profiles (solid lines) and temperature profiles at commencement of the inversion (dashed lines). The wells are the natural state observation wells shown in Figure 4, numbered from left to right.

4.1 First Test Case

To compare the adjoint and direct methods against finite differencing a 21 parameter inversion problem was constructed. The task was to match the 207 synthetic observations of the true 2D model. As unknowns we chose the horizontal and vertical permeabilities of the six rock-types, as well as their porosities. This gave 12 adjustable permeability values and 6 adjustable porosities. The three remaining unknowns were the bottom boundary source enthalpy, the mass flow rate into each of the five left-most bottom boundary blocks and the heat flow rate into the other 95 bottom boundary blocks. For the inversions PEST estimated the log base 10 of each unknown parameter.

During inversion the permeabilities were allowed to take values ranging between 10^{-17} and 10^{-12} m^2. Permeabilities were initially set at 10^{-14} m^2. Porosities had initial guesses of 11% along with upper and lower limits of 2% and 20%. The source enthalpy was initialized as 1,100 kJ/kg and allowed to vary between 1,000 and 1,400 kJ/kg. The mass flow rate was allowed to vary between 0.01 and 0.05 kg/s and had an initial value of 0.02 kg/s. The heat flow into the other boundary blocks was initially chosen as 28 W, and could vary between 20 and 40 W. The initial observation mismatch had a value of 4.09×10^6. Figures 5, 6 and 7 show the temperature, enthalpy and pressure profiles for the initial set of parameters as dashed lines.

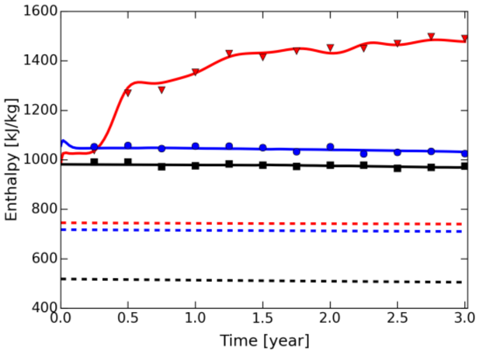

Figure 6: Production enthalpy data (markers), flowing enthalpies for initial parameter guess (dashed lines) and matched enthalpies using the direct method (solid lines). Producer 1 (▼), Producer 2 (●) and Producer 3 (■).

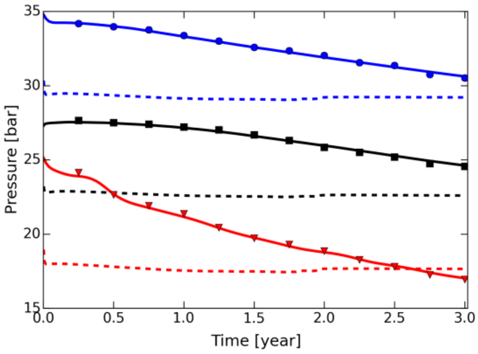

Figure 7: Production pressure data (markers), pressure transients for initial parameter guess (dashed lines) and matched pressures using the direct method (solid lines). Producer 1 (▼), Producer 2 (●) and Producer 3 (■).

To regularize the inverse problem we chose $\boldsymbol{v}(\boldsymbol{m})$ in equation (8) as $\boldsymbol{v}(\boldsymbol{m}) = \boldsymbol{m} - \boldsymbol{m}_0$, where $\boldsymbol{m}_0$ is the vector of the initial parameter guess. The aim of this regularization choice was to force the underdetermined parameters to be close to their initial guesses. The matrix $\boldsymbol{W}$ in (8) was chosen to be diagonal with elements W_{ii}=1/σ_i, where σ_i is a quarter of the difference between the upper and lower bounds of the relevant adjustable parameter m_i. In this way σ_i reflects the expected parameter variability or standard deviation of m_i.

Tables 1 and 2 compare inversion results for the 21 parameter problem when using PEST with one of the four sensitivity variants discussed in this paper: forward finite differencing (FWD), central finite differencing (CNTRL), the direct method and the adjoint method. All inversions were halted when the measurement misfit Φ_d fell below the number of data N_d, resulting in matches which are within the measurement noise level.

Table 1: Inversion results for the 21 parameter problem using PEST.

Derivative Method	Iter-ations	$\Phi(\boldsymbol{m})$	Φ_d	Φ_m	Time [s]
FWD	19	755.4	206.7	6.48	27,467
CNTRL	25	638.2	206.4	6.70	73,097
Direct	19	653.7	203.1	6.63	4,907
Adjoint	23	770.6	207.0	6.47	6,290

When evaluating model sensitivities by finite differencing, we chose to perturb permeabilities and porosities by 1% of their respective values. This is the default choice when using iTOUGH (Finsterle, 2007) and similar values are often used by TOUGH2 practitioners. However, from preliminary tests we found using 1% parameter perturbations to be ineffective and unreliable for the three boundary parameters. For a trial inversion test, use of 1% perturbations with the FWD method required about three times as many PEST iterations as using the direct or adjoint methods. We found that perturbing boundary parameters by smaller values gave better results and sensitivities more consistent with the direct and adjoint methods. Consequently, we chose to perturb the boundary parameters by 0.01% when using finite differencing for the final test results presented here. The uncertainty of how much to perturb parameters by is a known drawback of using finite differencing for derivative evaluations (Bjarkason *et al.*, 2015).

The results in Table 1 show that the direct and adjoint methods outperformed finite differencing, in terms of computational time. Inverting with sensitivities calculated by finite differencing took about eight hours or more. The adjoint and direct methods, on the other hand required less than a couple of hours. The extra computational expense of using the more reliable central differencing scheme over forward differencing was not worth it for this problem. All four methods required a similar number of PEST iterations.

We performed a limited number of additional tests whereby the PEST settings were varied for this 21 parameter problem. The number of inversion iterations required by each method relative to the other methods appeared arbitrary. No conclusions could, therefore, be made on the effectiveness of the adjoint or direct method for reducing the number of inversion iterations. A larger sample size is needed to draw any conclusions on which method can be considered the most reliable.

Table 2 compares the computational cost of the four sensitivity methods under consideration. The cost for one PEST iteration includes the average number of TOUGH2 simulations and the average number of simulation time-steps. The average natural state simulation required about 30 time-steps to reach the final target natural state simulation time of 10^{16} s. Production history simulations rarely deviated from using 123 time-steps.

The average total number of linear solves shown in Table 2 include the number of linear solves as well as the number of linear solves required for the adjoint or direct methods, when applicable.

Assuming that computational time is principally due to the number of linear solves it should be expected that the direct method will only be 2.5 times faster than using forward differencing. However, the speedup is more than twice that, since the average cost of solving for each right-hand side in equations (14), (16) and (18) is lowered by using the SuperLU solver. The cost of using the adjoint method is similarly less than what we would expect based on the number of linear solves. Nevertheless, the adjoint method is somewhat slower than the direct method every iteration, which is expected with the number observations far exceeding the number of adjustable parameters.

Table 2: Average computational cost of a PEST iteration for the 21 parameter inversions.

Derivative Method	FWD	CNTRL	Direct	Adjoint
Nonlinear Simulations	22.79	44.44	1.74	1.74
Simulation Time-steps	3,573	6,970	273	271
Newton Iterations	8,162	16,443	704	678
Linear Solves	8,162	16,443	3,292	5,783
Time [s]	1,446	2,924	258	273

4.2 Second Test Case

The previous inversion test case assumed ideal knowledge of the reservoir structure and parameterization. This will of course not be the case in a real case study. Instead the reservoir structure will largely be unknown apart from what can be inferred from surface geology and wellbore data.

Highly parameterized and regularized inversion offers a way of trying to deal with the lack of certainty with regards to model parameterization and the structure of a reservoir. As long as there are not too many observations, the adjoint method can be used to evaluate model sensitivities to guide the inverse process, when there are more parameters than observations.

We now assume that we do not know the extent of the six rock formations and divide the model up into 320 adjustable rock-types, excluding the atmospheric boundary blocks. Every adjustable rock-type covers a square in the vertical plane and contains 25 blocks. Each rock-type has unknown horizontal and vertical permeabilities, and an unknown porosity. This gives a total of 960 adjustable rock-type parameters. We can then use PEST to estimate these 960 parameters as well as the three bottom boundary parameters from the previous test case. Otherwise the model is the same as the true one.

To account for the large number of unknowns we included a regularization term which tends to smooth out spatial discontinuities for a certain type of parameter that varies in space, such as porosity. For this inverse problem, with 963 parameters, the regularization functional was

$$\Phi_m(\boldsymbol{m}) = \Phi_m^1 + \Phi_m^2 + \Phi_m^3 \tag{26}$$

where

$$\Phi_m^1 = \sum_{ij} \left\{ \begin{array}{c} (0.8)^2\left[\log\left(k_x^i\right) - \log\left(k_x^j\right)\right]^2 + \\ + (0.8)^2\left[\log\left(k_z^i\right) - \log\left(k_z^j\right)\right]^2 + \\ + 4^2\left[\log\left(\phi^i\ \right) - \log\left(\phi^j\ \right)\right]^2 \end{array} \right\} \tag{27}$$

$$\Phi_m^2 = \sum_i (0.8)^2\left[\log\left(k_x^i\right) - \log\left(k_z^i\right)\right]^2 \tag{28}$$

$$\begin{aligned} \Phi_m^3 &= (5.7)^2[\log(q) - \log(q_0)]^2 + \\ &+ (27.4)^2[\log(h) - \log(h_0)]^2 + \\ &+ (13.3)^2[\log(F) - \log(F_0)]^2 \end{aligned} \tag{29}$$

The summation Φ_m^1 is over neighbouring pairs of rock-types i and j. k^i_x, k^i_z and ϕ^i are the horizontal permeability, vertical permeability and porosity of rock-type i, respectively. This regularization term should make the inverse problem better posed but still allow spatial heterogeneity to arise where needed. The second term Φ_m^2 in (26) is included since we expect similar values for the horizontal and vertical permeabilities. Finally, Φ_m^3 includes the same type of regularization terms as were used in the first test case for the boundary parameters, where x_0 denotes the initial guess of parameter x. q is the enthalpy source rate, h is the source enthalpy and F is the bottom boundary flux parameter.

The initial guesses for the parameters were the same as in the previous example. We used PEST with adjoint evaluated sensitivities to estimate the 963 parameters given the synthetic data. The aim was to achieve an observation mismatch of N_d = 207, however the inversion stalled after 43 iterations having found an observation mismatch of 209.3, but this is close enough to the target value for practical purposes.

The inversion took 13,554 s to run and 97 nonlinear forward TOUGH2 simulations were carried out. The average execution time for a PEST iteration was therefore only about 315 seconds, which is not much more than the time needed for the 21 parameter problem.

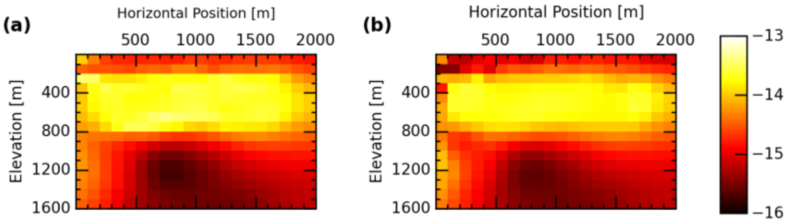

Figure 8: Base ten logarithms of the estimated formation permeabilities: (a) the horizontal permeability k_x [m^2]; (b) the vertical permeability k_z [m^2].

Employing a finite differencing approach to evaluate sensitivities is not very practical for such a large problem. Using finite differencing to evaluate the model sensitivities for this inverse problem would be likely to require nearly a day for each iteration using serial PEST. Though, the execution time for finite differencing can be reduced by employing parallel versions of PEST, we would need about 200 processors or more to achieve execution times similar to those achieved by using the adjoint method.

Using PEST, the bottom boundary heat flux was estimated to be 80.2 mW/m^2 which is very close to the true value of 80 mW/m^2. The estimated source enthalpy was only 2 kJ/kg lower than the true value. The mass source rate had an estimated value of 0.027 kg/s, which is also not far off the true value of 0.03 kg/s.

The inverted permeability values are shown in Figure 8. The estimated horizontal and vertical permeabilities show all the same large-scale features as the true system, see Figure 2. However, the data was unable to inform PEST of the smaller scale features such as the discontinuity between the top two rock formations.

Figure 9 shows the estimated porosity values. The inversion does not appear to have revealed any structural information regarding porosity variability in the reservoir. There is only slight variability in the estimated porosity values around the production area.

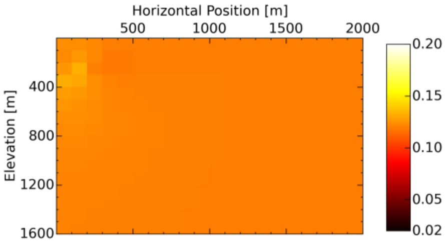

Figure 9: Estimated formation porosity.

5. CONCLUSIONS

The adjoint and direct methods appear to be viable options for reducing computational effort in inverse modelling of geothermal reservoirs. Both methods were found to work reliably for high enthalpy, two-phase natural state and production history simulations.

As the sample problems show, the use of the direct and adjoint methods can offer considerable reduction in execution times for model inversions using gradient based methods. The computational savings are mostly due to a reduction in the number of nonlinear forward reservoir simulations performed for each iteration of parameter updating compared with the traditionally used finite differencing methods.

ACKNOWLEDGEMENTS

The authors appreciate the contribution of the NZ Ministry of Business, Innovation and Employment for funding this work through the grant: C05X1306, “Geothermal Supermodels”. This work was also kindly supported by Landsbankinn and an AUEA Braithwaite-Thompson Graduate Research Award.

REFERENCES

Anterion, F., Eymard, R., & Karcher, B.: Use of parameter gradients for reservoir history matching. *Paper SPE 18433*, presented at the SPE Symposium on Reservoir Simulation, Houston, Texas. (1989).

Bjarkason, E.K., O’Sullivan, M.J., & O’Sullivan, J.: Improved sensitivity calculations. *Proc., 37th New Zealand Geothermal Workshop.* Taupo, New Zealand. (2015).

Bjarkason, E.K., O’Sullivan, M.J., O’Sullivan, J.P. & Yeh, A.: Accelerating Calibration of Natural State Geothermal Models. *Proc., 41st Workshop on Geothermal Reservoir Engineering.* Stanford University, Stanford, California. (2016).

Doherty, J.: Ground water model calibration using pilot points and regularization. *Ground Water*, *41* (2), pp. 170-177. (2003).

Doherty, J.: *Calibration and Uncertainty Analysis for Complex Environmental Models.* Watermark Numerical Computing, Brisbane, Australia. (2015).

Doherty, J.: *PEST: Model-Independent Parameter Estimation. User Manual PART I: PEST, SENSAN and Global Optimisers*, 6th edition. Watermark Numerical Computing, Brisbane, Australia. (2016).

Engl, H.W., Hanke, M. and Neubauer, A.: Regularization of inverse problems. *Mathematics and Its Applications*, vol. 375. Kluwer Academic Publishers, Dordrecht. (1996).

Finsterle, S.: *iTOUGH2 User's Guide.* Report LBNL-40040, Lawrence Berkeley National Laboratory, Berkeley, California. (2007).

Levenberg, K.: A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, *2*, pp. 164-168. (1944).

Li, R., Reynolds, A.C., & Oliver, D.S.: History matching of three-phase flow production data. *SPE Journal*, *8* (4), pp. 328-340. (2003).

Marquardt, D.W.: An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society for*

Industrial and Applied Mathematics, 11, pp. 431-441. (1963).

Medina, A., & Carrera, J.: Geostatistical inversion of coupled problems: Dealing with computational burden and different types of data. *Journal of Hydrology, 281* (4), pp. 251-264. (2003).

Oliver, D.S., & Chen, Y.: Recent progress on reservoir history matching: A review. *Computational Geosciences, 15* (1), pp. 185-221. (2011).

Oliver, D.S., Reynolds, A.C., & Liu, N.: *Inverse Theory for Petroleum Reservoir Characterization and History Matching*, 1st edition. Cambridge University Press, Cambridge. (2008).

O'Sullivan, J., Croucher, A., Yeh, A., & O'Sullivan, M.: Improved convergence for air-water and CO_2-water TOUGH2 simulations. *Proc., 35th New Zealand Geothermal Workshop*. Rotorua, New Zealand. (2013).

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

Pruess, K., Oldenburg, C. & Moridis, G.: *TOUGH2 User's Guide*, version 2.0. Lawrence Berkeley National Laboratory, Berkeley, California. (1999).

The HDF5 Group: *HDF5 User's Guide*. (2016).

Yeh, A., Croucher, A.E., & O'Sullivan, M.J.: Recent developments in the AUTOUGH2 simulator. *Proc., TOUGH Symposium*. Berkeley, California. (2012).

APPENDIX

A. Forming the *G* Matrices

For the sake of brevity we look at forming the elements of the ***G*** matrices for adjustable porosities, and only consider the flow of water and energy. The ***G*** matrices can be found analytically in a similar fashion for the permeabilities and the bottom boundary parameters considered in this work.

Let the n^{th} adjustable model parameter m_n be the porosity of some rock type. Note that the flux and source terms of the TOUGH2 forward equations are independent of porosity. Accordingly, the n^{th} column of the matrix $\boldsymbol{G}_{\text{st}}$ is zero, see (15). That is for all i:

$$[\boldsymbol{G}_{\text{st}}]_{i,n} = \frac{\partial}{\partial m_n}[\boldsymbol{f}_{\text{st}}]_i = 0 \tag{A1}$$

Every production simulation time-step we need to evaluate:

$$[\boldsymbol{G}^k]_{i,n} = \frac{\partial}{\partial m_n}[\boldsymbol{f}^k]_i = \frac{1}{\Delta t^k}\frac{\partial}{\partial m_n}\left(M_i^k - M_i^{k-1}\right) \tag{A2}$$

If the forward residual $[\boldsymbol{f}^{\,k}]_i$ represents the mass or water component, then the accumulation terms are given by

$$M_i^k = \phi_i(\rho_l\theta_l + \rho_v\theta_v)^k \tag{A3}$$

Here ϕ_i is the porosity of block i, ρ is density and θ is saturation. The subscripts l and v denote values for liquid water and vapour, respectively.

If, on the other hand, the forward residual $[\boldsymbol{f}^k]_i$ represents the energy component, then the accumulation terms are given by

$$M_i^k = (1-\phi_i)\rho_r c_r T_i^k + \phi_i(\rho_l\theta_l\omega_l + \rho_v\theta_v\omega_v)^k \tag{A4}$$

Here ρ_r is the rock grain density, c_r is the specific heat of the rock, ω is specific internal energy and T_i is the block temperature.

Now, for the mass equations we need to evaluate

$$\frac{\partial M_i^k}{\partial m_n} = \frac{\partial\phi_i}{\partial m_n}\frac{M_i^k}{\phi_i} \tag{A5}$$

and for the energy equations we need to find

$$\frac{\partial M_i^k}{\partial m_n} = \frac{\partial\phi_i}{\partial m_n}\frac{1}{\phi_i}(M_i - \rho_r c_r T_i)^k \tag{A6}$$

For $\partial\phi_i/\partial m_n$ we need only know what rock-type block i belongs to. If block i belongs to the same rock type as the porosity parameter m_n, then $\partial\phi_i/\partial m_n = 1$, else $\partial\phi_i/\partial m_n = 0$.

B. Forming the *C* Matrices

As in Appendix A, we restrict the discussion on the ***C*** matrices in equation (21) to the flow of water and energy. In that case every simulation block has two primary variables. For a single-phase block i the primary variables are pressure P_i and temperature T_i, but for two-phase conditions the second primary variable is vapour saturation θ_i instead of temperature.

In TOUGH2 the vector of primary variables $\boldsymbol{u}^k$ is ordered such that the element number $2i-1$ is P_i and the following element is the second primary variable of block i.

For the adjoint or direct solve of the natural state we need to find the following values of the matrix $\boldsymbol{C}_{\text{st}}$:

$$[\boldsymbol{C}_{\text{st}}]_{j,n} = \frac{\partial r_j}{\partial[\boldsymbol{u}_{\text{st}}]_n} = \frac{\partial}{\partial[\boldsymbol{u}_{\text{st}}]_n}[\boldsymbol{d}(\boldsymbol{m})]_j \tag{B1}$$

For our tests the only natural state observations were block temperatures and (B1) was only nonzero if r_j corresponded to that type of observation. Let observation $[\boldsymbol{d}(\boldsymbol{m})]_j$ be the natural state temperature in block i, then the only nonzero elements in row j of the matrix $\boldsymbol{C}_{\text{st}}$ were

$$[\boldsymbol{C}_{\text{st}}]_{j,2i-1} = \frac{\partial[T_{\text{st}}]_i}{\partial[P_{\text{st}}]_i} = \begin{cases} 0\,, & \text{if } \theta_i = 0 \text{ or } \theta_i = 1 \\ \dfrac{\partial[T_{\text{st}}]_i}{\partial[P_{\text{st}}]_i}\,, & \text{if } 0 < \theta_i < 1 \end{cases} \tag{B2}$$

and

$$[\boldsymbol{C}_{\text{st}}]_{j,2i} = \begin{cases} \dfrac{\partial[T_{\text{st}}]_i}{\partial[T_{\text{st}}]_i} = 1\,, & \text{if } \theta_i = 0 \text{ or } \theta_i = 1 \\ \dfrac{\partial[T_{\text{st}}]_i}{\partial[\theta_{\text{st}}]_i} = 0\,, & \text{if } 0 < \theta_i < 1 \end{cases} \tag{B3}$$

The derivatives $\partial T_i/\partial P_i$ were evaluated analytically. Here we have used the fact that temperature and pressure are independent variables for single-phase conditions, but for two-phase conditions temperature is only a function of pressure.

For production the matrix $\boldsymbol{C}^k$ is only nonzero when the simulation time corresponds to an observation time. If

$[\boldsymbol{d}(\boldsymbol{m})]_j$ is the pressure in block *i* at time-step *k* then the only nonzero element in row *j* of the matrix $\boldsymbol{C}^k$ is

$$[\boldsymbol{C}^k]_{j,2i-1} = \frac{\partial r_j}{\partial P_i^k} = \frac{\partial P_i^k}{\partial P_i^k} = 1 \tag{B4}$$

For a flowing enthalpy observations h_q from block *i* at time-step *k* the only nonzero elements to consider were

$$[\boldsymbol{C}^k]_{j,2i-1} = \frac{\partial r_j}{\partial P_i^k} = \frac{\partial h_q^k}{\partial P_i^k} \tag{B5}$$

and

$$[\boldsymbol{C}^k]_{j,2i} = \begin{cases} \dfrac{\partial h_q^k}{\partial T_i^k}\,, & \text{if } \theta_i^k = 0 \text{ or } \theta_i^k = 1 \\ \dfrac{\partial h_q^k}{\partial \theta_i^k}\,, & \text{if } 0 < \theta_i^k < 1 \end{cases} \tag{B6}$$

The derivatives in equations (B5) and (B6) were evaluated internally using finite differencing by the special adjoint/direct version of the AUTOUGH2 executable.