IMPROVED SENSITIVITY CALCULATIONS

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

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

ebja558@aucklanduni.ac.nz

Keywords: Inversion, history matching, model sensitivities, adjoint method, direct simulation, TOUGH2, iTOUGH2, PEST.

ABSTRACT

Model sensitivities are a measure of how specific model results vary with selected model parameters (such as permeability and porosity). They are important for parameterizing numerical models, automatic history matching and uncertainty quantification of model predictions. However, the current way of evaluating model sensitivities using finite differencing does not scale well for large and highly nonlinear geothermal reservoir simulations.

Alternatively model sensitivities may be generated by solving related linear problems forward or backward in time. The method of choice depends on the dimensionality of the parameter and observation spaces. The proposed linear methods were tested on one-dimensional transient two-phase flow problems, solved using the TOUGH2 simulator. Results of the forward and backward propagating methods were compared with those found by using simple, but computationally expensive, finite differencing.

1. INTRODUCTION

Automatic history matching of a geothermal reservoir model is commonly done through solving a least squares minimisation problem. The aim is to calibrate a numerical model of a geothermal system using $\boldsymbol{m}$ parameters as model inputs. The model response $\boldsymbol{d(m)}$ should match the observed field measurements $\boldsymbol{d}^*$ as closely as possible. Defining the residual by:

$$\boldsymbol{r}(\boldsymbol{m}) = \boldsymbol{d}(\boldsymbol{m}) - \boldsymbol{d}^* \tag{1}$$

then the least squares functional can be written as

$$\begin{aligned} F(\boldsymbol{m}) &= \frac{1}{2}\|\boldsymbol{d}(\boldsymbol{m}) - \boldsymbol{d}^*\|^2 = \frac{1}{2}[\boldsymbol{r}(\boldsymbol{m})]^T\boldsymbol{r}(\boldsymbol{m}) \\ &= \frac{1}{2}\sum_i [r_i(\boldsymbol{m})]^2 \end{aligned} \tag{2}$$

The history matching task is to find the model parameters $\boldsymbol{m}$ that minimize F, i.e.:

$$\min_{\boldsymbol{m}} F(\boldsymbol{m}) \tag{3}$$

In practice it is usually of benefit to weight observations and regularize the above inversion problem. Weighting can be based on some prior knowledge or ideas about measurement error. Inclusion of a regularization term may help to find a solution which honours the geology of the system. However, for the following discussion we do not include weighting and regularizing, but the method introduced can easily be extended to cover the more general formulation.

Assuming continuous model parameters $\boldsymbol{m}$ and smooth enough model outputs, iTOUGH2 (Finsterle, 2007) and PEST (Doherty, 2013) offer robust implementations of the iterative Levenberg-Marquardt approach for finding an approximate solution to (3). During each iteration of the Levenberg-Marquardt method, the Jacobian matrix $\boldsymbol{S}$ is evaluated in terms of the updated model parameters $\boldsymbol{m}$:

$$\boldsymbol{S} = \frac{d[\boldsymbol{d}(\boldsymbol{m})]}{d\boldsymbol{m}} = \frac{d\boldsymbol{r}}{d\boldsymbol{m}} = \begin{bmatrix} \frac{dr_1}{dm_1} & \cdots & \frac{dr_1}{dm_n} \\ \vdots & \ddots & \vdots \\ \frac{dr_N}{dm_1} & \cdots & \frac{dr_N}{dm_n} \end{bmatrix} \tag{4}$$

The matrix contains the sensitivities, i.e. the derivatives of N model generated observations with respect to the n adjustable model parameters. For field-scale history matching, calculation of model sensitivities is one of the most computer intensive tasks.

Previously sensitivities for models implemented in TOUGH2 (Pruess *et al*., 1999) have been evaluated using finite differencing, which requires at least $n+1$ time-consuming, nonlinear forward simulations (for n model parameters). As discussed previously (Bjarkason *et al*., 2014), shorter computational times may be achieved by the use of analytic methods, including the chain rule, for calculating $\boldsymbol{S}$. This allows the sensitivity matrix $\boldsymbol{S}$ to be found by solving a sequence of linear problems either forward or backward in time.

A goal of this ongoing study is to test the applicability of the linear propagation methods to multiphase, multicomponent geothermal reservoir simulations. The intent is then to implement these methods to ease model development and uncertainty quantification. This paper discusses preliminary tests on three basic horizontal, one-dimensional transient TOUGH2 simulations. All the tests were conducted using the pure water module EOS1.

The modelled problems are all 1D and include a production injection doublet but were of increasing complexity in terms of the fluid state: a non-isothermal pure liquid flow problem, an initially liquid state flashing to two-phase, and an initially two-phase problem with cool liquid breakthrough.

2. TOUGH2 SIMULATIONS

TOUGH2 (Pruess *et al*., 1999) solves discretized heat and mass balance equations. The equations simply describe how the change in mass or energy in an element is equal to the net mass or energy that flowed in or out of the volume. Let the superscript k denote values at the end of the k^{th} time-step of duration Δt^k, $F^k_{\kappa ij}$ be the average flux of component κ over the interface area a_{ij} connecting blocks i and j (positive for flow from block i to block j), and $Q^k_{\kappa i}$ be the total source/sink flow rate into block i (positive for injection, negative for production). The amount of component κ per

unit volume $M_{\kappa i}$ in element i of volume V_i is found according to (Pruess *et al*., 1999; O'Sullivan *et al*., 2013)

$$\frac{V_i}{\Delta t^k}\left(M_{\kappa i}^k - M_{\kappa i}^{k-1}\right) = -\sum_j a_{ij} F_{\kappa ij}^k + Q_{\kappa i}^k \qquad (5)$$

Rearranging, the implicit equations solved by TOUGH2 for all volumes i and components κ are:

$$f_{\kappa i}^k = M_{\kappa i}^k - M_{\kappa i}^{k-1} + \frac{\Delta t^k}{V_i}\left[\sum_j a_{ij} F_{\kappa ij}^k - Q_{\kappa i}^k\right] = 0 \qquad (6)$$

These nonlinear equations, solved at the k^{th} time-step of a TOUGH2 simulation, can be represented using vector notation. For every time-step, TOUGH2 finds updated primary variables $\boldsymbol{u}^k$ (such as pressures and temperatures for a single-phase or pressure and saturation for two-phase conditions) for all volumes by solving

$$\boldsymbol{f}^k(\boldsymbol{u}^k, \boldsymbol{u}^{k-1}, \boldsymbol{m}) = \boldsymbol{0} \qquad (7)$$

Each element of the vector $\boldsymbol{f}^k$ represent a residual $f_{\kappa i}^k$, see equation (6). Note that the forward equations (7) for the k^{th} time-step are a function of the new primary variables $\boldsymbol{u}^k$, primary variables for the previous time-step $\boldsymbol{u}^{k-1}$ and the model parameters $\boldsymbol{m}$. Further, the forward residuals in (7) can be written as

$$\boldsymbol{f}^k = \boldsymbol{M}(\boldsymbol{u}^k, \boldsymbol{m}) - \boldsymbol{M}(\boldsymbol{u}^{k-1}, \boldsymbol{m}) + \boldsymbol{FS}(\boldsymbol{u}^k, \boldsymbol{m}) \qquad (8)$$

Here the $\boldsymbol{M}$'s are column vectors of accumulated mass and energy, and $\boldsymbol{FS}$ represents the flux and source terms.

For every time-step the nonlinear equations (7) are solved using Newton-Raphson iterations. For the p^{th} Newton iteration the primary variables are updated according to

$$\boldsymbol{A}^{k,p-1}\left(\boldsymbol{u}^{k,p} - \boldsymbol{u}^{k,p-1}\right) = -\boldsymbol{f}^k(\boldsymbol{u}^{k,p-1}, \boldsymbol{u}^{k-1}, \boldsymbol{m}) \qquad (9)$$

where

$$\boldsymbol{A}^{k,p-1} = \left[\frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{u}^k}\right]^{p-1} \qquad (10)$$

In TOUGH2 the initial guess for every new time-step is $\boldsymbol{u}^{k,0}=\boldsymbol{u}^{k-1}$. The updating procedure is halted when all of the forward residuals meet one of the following convergence criteria (Pruess *et al*., 1999):

$$\left|\frac{f_{\kappa i}^{k,p}}{M_{\kappa i}^{k,p}}\right| \le \varepsilon_1 \qquad (11)$$

or

$$\left|f_{\kappa i}^{k,p}\right| \le \varepsilon_1 \varepsilon_2 \qquad (12)$$

The tolerances ε_1 and ε_2 have default values of 10^{-5} and 1, respectively. With (11) or (12) realized for all components of each numerical block, then $\boldsymbol{u}^k=\boldsymbol{u}^{k,p}$ and the simulation moves on to the next time-step.

3. LINEAR PROPAGATION

For the linearized approaches to calculating model sensitivities the important matrices are:

$$\boldsymbol{A}^k = \frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{u}^k} \;;\; \boldsymbol{B}^k = \frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{u}^{k-1}}$$
$$\boldsymbol{G}^k = \frac{\partial \boldsymbol{f}^k}{\partial \boldsymbol{m}} \;;\; \boldsymbol{C}^k = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}^k} \qquad (13)$$

3.1 Direct Simulations

Using the forward propagation or direct simulation method (Anterion *et al*., 1989; Bjarkason *et al*., 2014) we track forward in time solving (7) and

$$\boldsymbol{A}^k\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^k = -\boldsymbol{G}^k - \boldsymbol{B}^k\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^{k-1} \qquad (14)$$

for every time-step. Equation (14) is a linear matrix equation with n (the number of adjustable model parameters) right-hand sides. By solving (14) for $\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^k$ we obtain the model sensitivities of every primary variable at that time. Note that if the initial conditions are constant, then

$$\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^0 = \boldsymbol{0} \qquad (15)$$

However, this may not be the case for a production scenario for which the initial state is determined through a natural state simulation.

The sensitivity matrix is subsequently found according to

$$\boldsymbol{S} = \frac{d\boldsymbol{r}}{d\boldsymbol{m}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}} + \sum_k \boldsymbol{C}^k\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{m}}\right]^k \qquad (16)$$

The linear propagation method outlined here is unlike simple finite differencing, which require at least $n+1$, time-consuming, nonlinear forward simulations. Instead the forward propagation method solves only one nonlinear and n linear problems. Replacing n nonlinear tasks with corresponding linear ones should result in reduced computational effort.

3.2 Adjoint Simulations

Another method for finding $\boldsymbol{S}$ is to propagate backward in time (Bjarkason *et al*., 2014; Li *et al*., 2003). The backtracking approach involves solving linear matrix problems, where the coefficient matrix is the transpose or adjoint of the one found in (14). For this approach we first solve all the nonlinear forward equations (7) up to and including the last observational time-step Δt^k, where $k=N_t$. Subsequently we solve a related adjoint problem for the Lagrange multipliers $\boldsymbol{\Psi}^k$. The backtracking method starts by solving

$$[\boldsymbol{A}^{N_t}]^T \boldsymbol{\Psi}^{N_t} = -[\boldsymbol{C}^{N_t}]^T \qquad (17)$$

and going back in time solves

$$[\boldsymbol{A}^k]^T \boldsymbol{\Psi}^k = -[\boldsymbol{C}^k]^T - [\boldsymbol{B}^{k+1}]^T \boldsymbol{\Psi}^{k+1} \qquad (18)$$

for all previous times. The number of right-hand sides involved in equations (17) and (18) are equal to the number of individual observations N. Note, however, that in practice the number of right-hand sides is at most N, because a column of $\boldsymbol{\Psi}^k$ only becomes nonzero when the corresponding columns of $\boldsymbol{C}^k$ or $\boldsymbol{\Psi}^{k+1}$ are nonzero.

The sensitivities are now found by

$$\boldsymbol{S} = \frac{d\boldsymbol{r}}{d\boldsymbol{m}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}} + \sum_k [\boldsymbol{\Psi}^k]^T \boldsymbol{G}^k \tag{19}$$

Although these propagation methods have found favour in other scientific disciplines (Medina & Carrera, 2003; Oliver *et al*., 2008; Reuther *et al*., 1996), it is not at all trivial to see whether they will easily work for highly nonlinear geothermal simulations. Geothermal reservoir simulations involve notoriously ill-posed or badly conditioned forward matrices $\boldsymbol{A}^k$ and the forward problem itself may not able to be solved when modelling a system under great stresses.

3.3 Implementation

Our tests used the TOUGH2 forward code available along with the iTOUGH2 parameter estimation suite (Finsterle, 2007). The preliminary tests were aimed at checking the suitability of the linear methods. For ease of implementation we calculated the sensitivity matrices after completion of the forward simulations using Python. The TOUGH2 forward code was edited in order to write out and store the necessary information for the backward and forward propagation methods.

The TOUGH2 code already calculates the converged $\boldsymbol{A}^k$ and $\boldsymbol{B}^k$ matrices that are needed. They are evaluated, as part of a standard forward run, along with the forward residuals (8) before convergence and conclusion of a time-step. These matrices are calculated within TOUGH2 using finite differentiation, though this could be done using automatic differentiation in the future (Kim & Finsterle, 2003; Wong *et al*., 2015).

For the accumulation matrices $\boldsymbol{B}^k$ we utilized the TOUGH2 algorithm which calculates the forward Jacobian $\boldsymbol{A}^k$ in the following sequence:

$$\bar{\boldsymbol{A}}^k = \frac{\partial \boldsymbol{M}(\boldsymbol{u}^k, \boldsymbol{m})}{\partial \boldsymbol{u}^k} = -\frac{\partial \boldsymbol{f}^{k+1}}{\partial \boldsymbol{u}^k} = -\boldsymbol{B}^{k+1} \tag{20}$$

$$\boldsymbol{A}^k = \bar{\boldsymbol{A}}^k + \frac{\partial \boldsymbol{FS}(\boldsymbol{u}^k, \boldsymbol{m})}{\partial \boldsymbol{u}^k} \tag{21}$$

During our model runs we evaluated the matrices $-\boldsymbol{B}^{k+1}$ and $\boldsymbol{A}^k$ using equations (20) and (21), respectively, and stored them for later use. However, in the final implementation primary variables could instead be stored and the required matrices calculated only when necessary. Observe that we could also have used the accumulation matrices calculated at the first Newton iteration at each time-step:

$$-\boldsymbol{B}^{k+1} = \frac{\partial \boldsymbol{M}(\boldsymbol{u}^{k+1,0}, \boldsymbol{m})}{\partial \boldsymbol{u}^{k+1,0}} \tag{22}$$

since $\boldsymbol{u}^{k,0}=\boldsymbol{u}^{k-1}$. This interchangeability holds even for varying time-steps because the time component is only included in the flux and source terms (see (6)).

The matrices $\boldsymbol{C}^k$ and $\boldsymbol{G}^k$ depend on the types of simulated observations and model parameters, respectively. The matrices $\boldsymbol{C}^k$ are calculated for each simulation. Evaluation of $\boldsymbol{G}^k$ is addressed as follows for the specified model experiments.

For the tests conducted the only adjustable model parameters were logarithms of formation permeabilities. The permeabilities only appear in the forward residuals through the advective fluxes of each phase β described using Darcy's law (Pruess *et al*., 1999):

$$F^k_{\beta ij} = -k_{ij}\left(\frac{k_{r\beta}}{\nu_\beta}\right)^k_{ij}\left[\frac{P^k_j - P^k_i}{D_{ij}} - \rho^k_{\beta ij} g_{ij}\right] \tag{23}$$

Here the subscripts ij designate averaged values over the interface between blocks i and j, and as before the superscript k denotes values at the k^{th} time-step. $F^k_{\beta ij}$ is the flux of phase β, k_{ij} is the absolute permeability and P^k_i is the pressure of block i. ρ^k_β, $k_{r\beta}$ and ν_β are the density, relative permeability and kinematic viscosity, respectively, belonging to phase β. g_{ij} is the component of gravity acting through the interface and $D_{ij}=D^i_{ij}+D^j_{ij}$, where D^i_{ij} is the distance between the block centre i and the interface ij (O'Sullivan *et al*., 2013).

In our simulations, we used harmonic weighting of the absolute permeabilities according to (O'Sullivan *et al*., 2013)

$$\frac{D_{ij}}{k_{ij}} = \frac{D^i_{ij}}{k_i} + \frac{D^j_{ij}}{k_j} \tag{24}$$

where k_i is the permeability of block i. All simulations used zero capillary pressures and linear relative permeability curves with residual liquid saturations of 0.5 and the relative permeabilities summing to 1.

The flux of water (κ=w) is found by summing over the liquid (l) and gas (g) phases:

$$F^k_{wij} = \sum_\beta F^k_{\beta ij} = F^k_{lij} + F^k_{gij} \tag{25}$$

The advective energy (κ=e) flux is found similarly according to

$$F^k_{eij} = \sum_\beta h_\beta F^k_{\beta ij} = h_l F^k_{lij} + h_g F^k_{gij} \tag{26}$$

Here h_β stands for the specific enthalpy of phase β.

Each adjustable model parameter m_q considered here is the logarithm of a permeability $k_{RTq,AXq}$, where RTq is the rock type associated with parameter m_q and AXq is the principal axis which the permeability belongs to. Assuming a regular grid with block connections along the principal axes, $\boldsymbol{G}$ can be calculated using:

$$\begin{aligned} G^k_{\kappa iq} &= \frac{\partial f^k_{\kappa i}}{\partial m_q} = \frac{\partial f^k_{\kappa i}}{\partial\left[\log_{10}\left(k_{RTq,AXq}\right)\right]} \\ &= \frac{\Delta t^k}{V_i}\ln(10)\sum_j a_{ij}\frac{\partial F^k_{\kappa ij}}{\partial\left[\log_{10}\left(k_{RTq,AXq}\right)\right]} \\ &= \frac{\Delta t^k}{V_i}\ln(10)\sum_j a_{ij}F^k_{\kappa ij}\,\delta_{AXq,AXij} \\ &\cdot\left[\frac{D^i_{ij}}{D_{ij}}\frac{k_{ij}}{k_i}\delta_{RTq,RTi} + \frac{D^j_{ij}}{D_{ij}}\frac{k_{ij}}{k_j}\delta_{RTq,RTj}\right] \end{aligned} \tag{27}$$

Here δ_{nm} is the Kronecker delta function, $AXij$ is the principal axis along the connection between i and j, and RTi is the rock type of volume i. The first delta in equation (27) is to check whether the flux is along the same direction as

the adjustable permeability in question. The terms given in the square bracket result from differentiating the harmonically weighted permeabilities at the interface *ij*. The first term in the bracket is included if block *i* was assigned the parameter m_q and the second term is included if this parameter was assigned to the neighbouring block *j*.

4. SIMULATION EXPERIMENTS

The following three subsections detail some of the main results of our preliminary study of using linear propagation methods to evaluate model sensitivities. The model setup used for all three simulation variants is depicted in Figure 1.

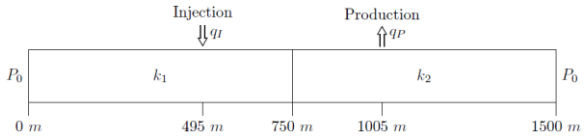

Figure 1: The one-dimensional, horizontal model setup, showing boundaries of constant pressure P_0=30 bar. There are two formations with permeabilities k_1 and k_2.

The model domain is one-dimensional with a length of 1,500 m. The standard horizontal model for all simulations has 52 elements. The boundary conditions at 0 m and 1,500 m were modelled as constant pressure boundary conditions using large volume blocks. For all simulations, the initial pressure over the entire domain was P_0=30 bar. Aside from the constant pressure blocks, the elements are cubes of volume 27,000 m^3 with nodal points at the geometric centres.

There are two formations or rock types, as shown in Figure 1, having different permeabilities and each zone spans 750 m. During the simulations, water is injected at a constant injection rate q_I=q and constant enthalpy h_I into the first formation. The temperature of the injected water corresponds to 20°C and the injection point is at about 500 m from the left boundary in Figure 1. Roughly 500 m away from the right boundary, fluid is produced at a constant production rate q_P=-q. The injection formation has a permeability k_1 and the production formation has a permeability k_2. Hereafter we refer to these two formations as the injection and production formations, respectively. The simulation parameters used for all model runs are listed in Table 1.

For our simulations we used the MOMOP option available when using iTOUGH2 (Finsterle, 2014). We set MOP2(1)=2 to ensure that primary variables were updated at every time-step. Moreover, the relative convergence tolerance ε_1 was set to 10^{-8}, see equations (11). The LUBAND direct solver method (Pruess *et al*., 1999) was selected for working out the linear Newton-Raphson updates (9).

For all simulations we ran TOUGH2 for 10 constant time-steps and took the pressure at the injection block P_I and the pressure at the production block P_P at the final time as observations. The pressure observations were measured in pascals (Pa). The adjustable model parameters were taken to be the base ten logarithms of the two formation permeabilities. This results in the model Jacobian or sensitivity matrix as

Table 1: Simulation parameters for the two formations shown in Figure 1 and parameters common to both.

	Parameter	Value
Injection formation	Permeability, k_1	1.5×10^{-14} m^2
	Injection rate, q_I	0.3 kg/s
	Injection enthalpy, h_I	83.9 kJ/kg
Production formation	Permeability, k_2	2.5×10^{-14} m^2
	Production rate, q_P	-0.3 kg/s
Common parameters	Rock grain density, ρ_R	2500 kg/m^3
	Conductivity, K	2.5 W/(m·K)
	Specific heat, C_R	1.0 kJ/(kg·K)
	Pore compressibility, c_R	1.0×10^{-9} Pa^{-1}
	Porosity, ϕ	0.1

$$\boldsymbol{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial P_I^{10}}{\partial[\log_{10}(k_1)]} & \dfrac{\partial P_I^{10}}{\partial[\log_{10}(k_2)]} \\ \dfrac{\partial P_P^{10}}{\partial[\log_{10}(k_1)]} & \dfrac{\partial P_P^{10}}{\partial[\log_{10}(k_2)]} \end{bmatrix} \quad (28)$$

The forward and backward propagation methods were used to calculate the matrix $\boldsymbol{S}$. To check the accuracy of our implementations, we compared the evaluated sensitivities with those found using iTOUGH2's forward and central finite difference methods (Finsterle, 2007).

Note, since the observations are themselves primary variables of the injection and production elements, the matrices $\boldsymbol{C}^k$ are sparse with two entries which are 1 for the final time-step, and

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}} = \boldsymbol{0} \quad (29)$$

The partial derivatives in (29) are zero because the observation pressures do not depend explicitly on the model parameters. The pressures only depend implicitly on the permeabilties through (7). Therefore, we only required the summation terms in (16) or (19) to find the sensitivities.

4.1 Non-Isothermal Liquid Only

The first test case was a non-isothermal problem, where the water was exclusively in the liquid phase over the entire simulation. At the outset all elements had a temperature of 200°C.

The linear propagation methods were tested for various time-step sizes. Figure 2 illustrates the normalized differences between the elements of the Jacobian matrix (28) found using the two linear propagation methods. The results show that the two methods agree closely. Due to this good agreement only the forward propagation method was used to compare with the finite difference approaches.

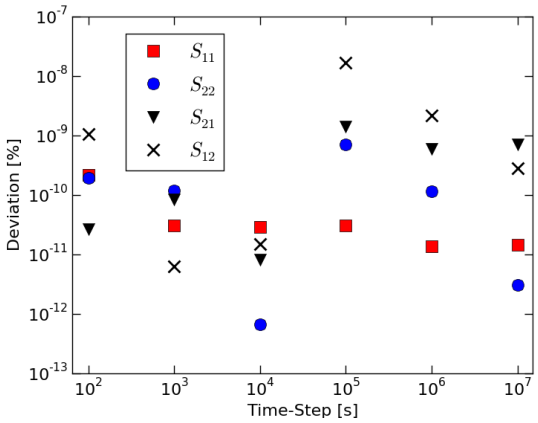

Figure 2: Plot of the normalized difference between the model sensitivities found using the forward and backward propagation methods as a function of time for the liquid only problem.

Figure 3 shows for a time step of 10^7 s (almost four months) the difference between the sensitivity matrices found by the linear forward propagation method and the ones evaluated using the two finite difference approaches. Comparisons were not made using longer time-steps ($\Delta t^k \geq 10^8$ s), since the simulator would not converge within the maximum default number of Newton iterations without changing the time-step size.

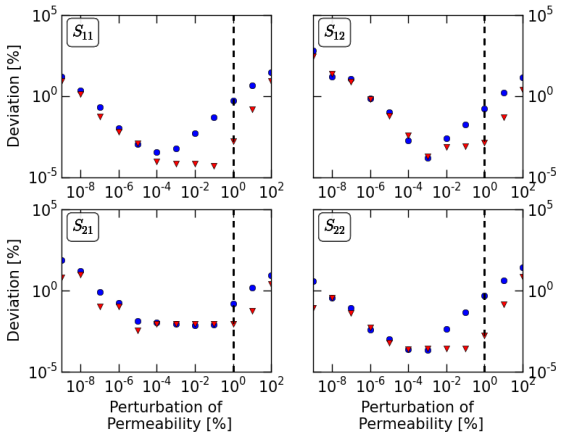

Figure 3: Absolute normalized differences between elements of the Jacobian matrix *S* using the linear forward propagation approach and finite differencing for various model parameter perturbations. The blue circles (•) denote deviations obtained when using iTOUGH2's forward finite differencing and the red triangles (▾) central finite differencing. All terms were normalized using the corresponding value of the forward propagation method. The results are for a liquid only simulation and a constant time-step $\Delta t^k = 10^7$ s. The black dashed line indicates the default perturbation setting in iTOUGH2 of 1%.

The results in Figure 3 show that the forward propagation method is in good agreement with the finite differencing approaches. Supposing that the linear propagation method is close to the analytical solution, then for relatively large perturbations the central difference method should generally be expected to give smaller deviations than forward differences. That pattern can be seen in Figure 3 and was found to be the case for most of our tests. This is due to the smaller truncation error of the central scheme, but for inversions it comes at the cost of having to run twice as many simulations. As expected, the comparison worsens for smaller perturbations due to finite precision round-off error. The typical results as given in Figure 3 indicate that the linear propagation methods give accurate model derivatives. However, these tests cannot quantify how close the results are to the true theoretical sensitivities.

Figure 4 shows results, for a much smaller time-step, where the above pattern is not observed for all sensitivities.

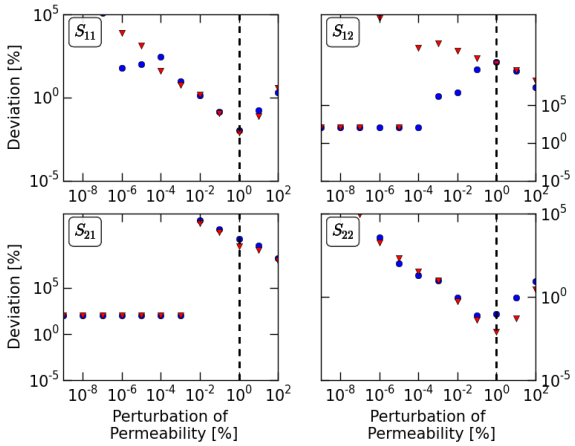

Figure 4: The same comparison as the one shown in Figure 3, but using a time-step $\Delta t^k = 100$ s.

The results show a relatively large discrepancy between the calculated values for the elements S_{12} and S_{21}. Nonetheless, the elements S_{11} and S_{22} compare favourably. The main reason is that the sensitivities S_{12} and S_{21} are so small that it makes them difficult to evaluate. This is because perturbing a permeability induces relatively large pressure changes in nearby elements, but has little to no effect far away. From Figure 4 it is hard to say which method is correct, only that the forward propagation method and the finite difference approach produce different results.

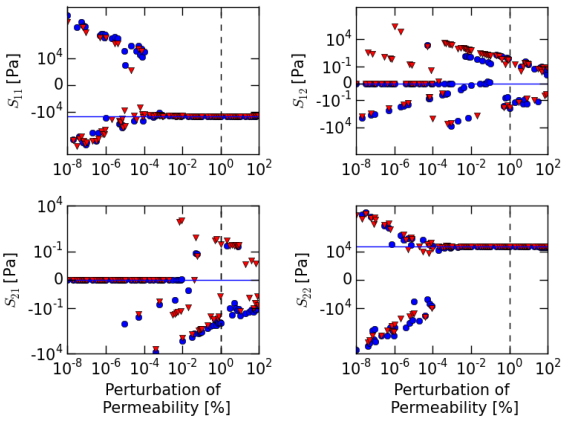

Figure 5: Model sensitivities for liquid only flow and a time-step $\Delta t^k = 100$ s. The blue circles (•)and red triangles (▾) denote values obtained with iTOUGH2's forward and central finite differencing methods, respectively. The blue lines indicate the sensitivities found using either of the linear propagation methods.

Greater insight can be gained by looking at Figure 5, which shows the variability of sensitivities evaluated using finite differentiation. Figure 5 shows that even the larger sensitivities S_{11} and S_{22} suffer from machine rounding errors when evaluated using either finite difference method along with very small parameter perturbations. The presence of rounding errors is evident in the sign change of sensitivities as perturbations are varied. However, for larger perturbations the finite difference methods give values which agree with the forward and backward propagation approaches.

In contrast it is clear from Figure 5 that the finite difference methods did not settle on an estimate of S_{12} or S_{21}, even for large parameter perturbations. For very small perturbations the values of both sensitivities were usually found to be zero using finite differences, but that was not always the case as Figure 5 shows. Due to rounding errors the finite difference methods can clearly not determine the sign of either S_{12} or S_{21}. For the larger perturbations these sensitivities hover about ±0.1 Pa. Looking at the finite difference results, setting both sensitivities to zero would likely be the best compromise.

By comparison the linear propagation methods appeared to give fairly reasonable values for the model sensitivities. The linear forward and backward propagation methods gave the following sensitivity matrix for Δt^k=100 s:

$$\boldsymbol{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} -3.3 \times 10^{4} & -6.4 \times 10^{-9} \\ 2.0 \times 10^{-8} & 3.3 \times 10^{4} \end{bmatrix} \text{ Pa}$$

Yet iTOUGH2's central difference method gave, using the default 1% parameter perturbation:

$$\boldsymbol{S} = \begin{bmatrix} -3.3 \times 10^{4} & -2.3 \\ 0.64 & 3.3 \times 10^{4} \end{bmatrix} \text{ Pa}$$

The evaluated S_{12} and S_{21} given in the two matrices above differ by many orders of magnitude. In our opinion the linear propagation methods gave much more reliable model sensitivities than the simpler finite difference approaches.

These results emphasize that there is no all-purpose perturbation which can be used for black-box finite differentiation. This usually results in using trial and error to find a suitable perturbation, though the iTOUGH2 default value may usually be suitable for large sensitivities. By contrast, the proposed linear propagation methods do not rely on having to pick a rather speculative parameter perturbation, which could impact the accuracy of the sensitivities. Inaccurately evaluated sensitivities can undermine the efficiency of model inversions using iTOUGH2 or PEST. The linear propagation approaches have the potential to mitigate inaccuracies in sensitivity evaluations. This aspect alone could speed up history matching. There is even greater room for improvement for steady state simulations, which commonly suffer from convergence issues. We are currently working on a steady state implementations of the linear propagation methods.

4.2 Non-Isothermal Flashing

The second simulation experiment begins with a system in a pure liquid state at 230°C. As fluid is extracted from the production zone, the water flashes to two-phase around the production well. Figure 6 shows the approximate onset of two-phase conditions at the production block.

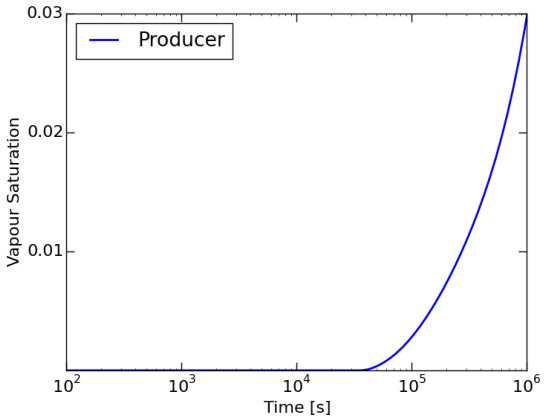

Figure 6: Saturation at the production well, for the flashing problem. The system became two-phase after a few hours of extraction.

The purpose of this simulation experiment and the one discussed in the next subsection was to test the effects of phase transitioning on the linear propagation methods. This makes the methods more complicated because during a phase transition some of the primary variables switch from temperature to saturation. For this case a constant simulation time-step of 10^6 s was selected. Figure 7 shows the saturation profile across the simulation domain for three selected time-steps, including the final time.

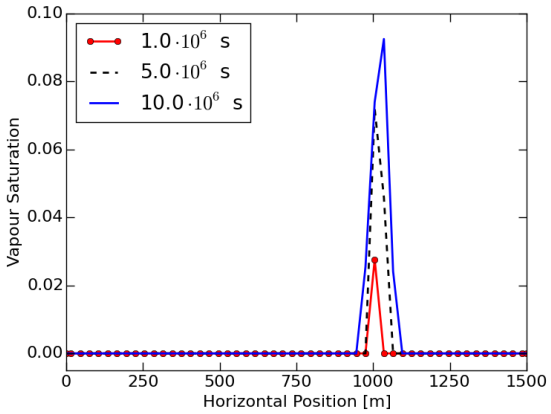

Figure 7: Vapour saturation over the simulation domain for the flashing flow problem. Four blocks changed to two-phase conditions by end of the simulation.

Like the previously discussed flow problem Figure 8 shows that the two linear propagation methods give near identical results for this case. Figures 8 and 9 demonstrate that the reliability of the backward and forward propagation methods were unaffected by the phase transition.

4.3 Two-Phase Initially

The third and final simulation experiment was a problem with two-phase initial conditions. Initially the vapour saturation was 10% over the whole domain. As the cool injection fluid gets added to the system a liquid zone developed as illustrated in Figures 10 and 11.

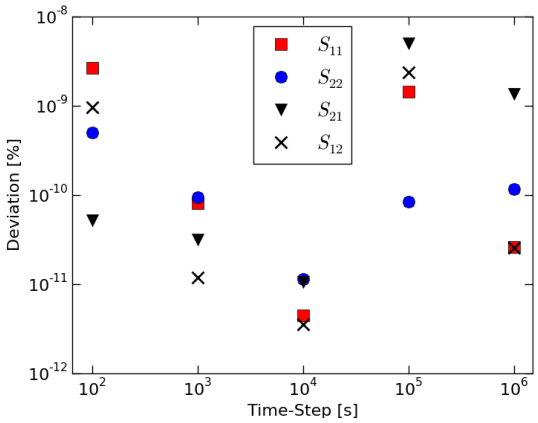

Figure 8: Normalized differences between sensitivities found using the forward and backward propagation methods for the flashing problem. Only time-steps $\Delta t^k \geq 10^5$ s produced two-phase conditions.

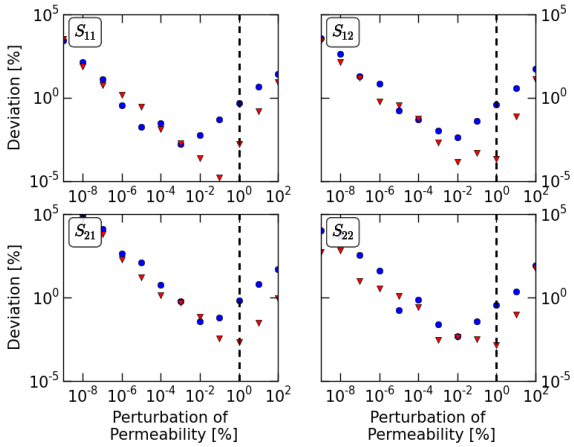

Figure 9: Same as Figure 3, but for the flashing problem and a time-step of 10^6 s.

A constant time-step of 10^6 s was chosen for this experiment. The selected time-step was long enough for phase changes to take place around the injection point, but small enough to allow for a constant time-step. Figure 11 shows some simulation results at various times.

Figure 12 compares the forward propagation method with the two finite difference methods. As shown, the forward propagation method does a good job of evaluating the sensitivity matrix. The same applies to the backtracking method. The linear tracking methods found the sensitivity matrix as

$$\boldsymbol{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} -6.3 \times 10^5 & -2.2 \times 10^1 \\ -1.1 \times 10^{-2} & -2.8 \times 10^5 \end{bmatrix} \text{ Pa}$$

Figure 12 shows that the forward propagation method and the finite differencing methods compare less favourably for element S_{21}. Still, the difference is only about 0.5% between the S_{21} found using the forward propagation method and the one found using central differencing and the iTOUGH2 default 1% perturbation. The reason for the less favourable comparison for S_{21} is due to its small absolute value of about 0.01, which is three orders of magnitude smaller than S_{12} and seven orders smaller than the other sensitivities. The finite differencing methods compare a bit better for the intermediate sensitivity S_{12}. This result is consistent with the one discussed for the liquid only experiment, that finite differencing gives less accurate results for small sensitivities.

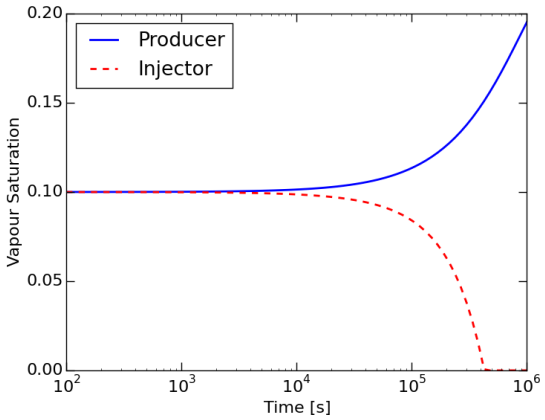

Figure 10: Saturation at the production and injection wells, for the initially two-phase problem. The injection block was single-phase after a few days.

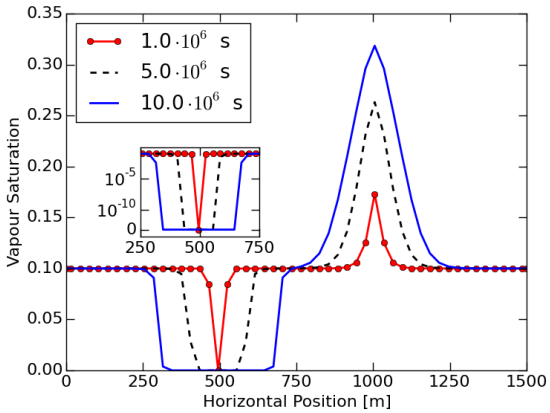

Figure 11: Vapour saturation over the simulation domain for the initially two-phase flow problem.

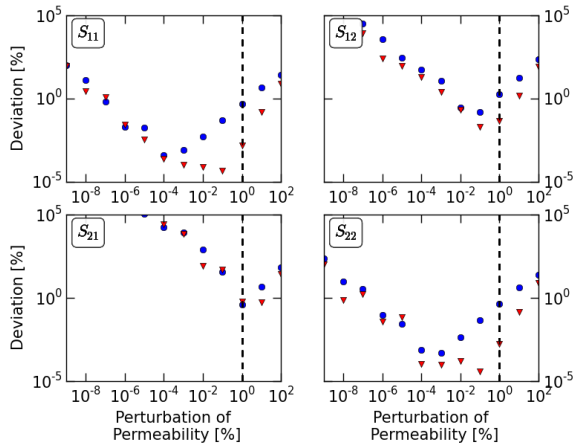

Figure 12: Same as Figure 3, but for the flow problem with two-phase initial conditions and a time-step of 10^6 s.

As before a comparison was made between the forward and backward tracking linear methods, (Figure 13). It showed that there was only one model sensitivity for which the normalized difference was greater than 1%. The absolute value of the responsible sensitivity itself was evaluated to be about 10^{-53}, by both linear propagation methods. The insignificance of the sensitivity and arithmetic error is the probable cause of the larger relative discrepancy.

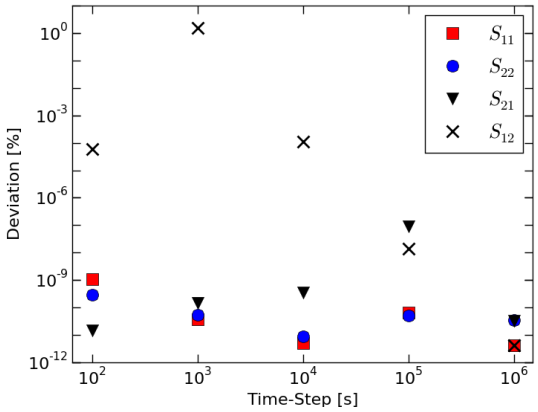

Figure 13: Normalized difference between sensitivities found using the forward and backward propagation methods for the initially two-phase problem. Only time-steps $\Delta t^k \geq 10^5$ s gave phase transitions from two-phase to single-phase.

As for the flashing test model, the phase transitions for the initially two-phase problem do not result in degradation of the linear propagation methods. The consistently good agreement between the forward and backward propagation methods (Figures 2, 8 and 13) shows that conditioning of the forward matrices $\boldsymbol{A}^k$ did not severely impact the results.

Finally Table 3 gives a rough idea of the relative computational cost of the four methods that were tested for calculating model sensitivities. Note that the number of Newton iterations is counted as ITER-1, where ITER is the nonlinear Newton iteration counter in TOUGH2. As would be expected, the number of Newton iterations increases as the simulations become more complex.

Table 3 shows the number of linear problem solves needed to calculate the model sensitivities. The number of linear solves for the forward (Dir) and backward (Adj) propagation methods was the same because the number of observations N equalled the number of model parameters n and all the observations were taken at the last simulation time.

Notice that Table 3 denotes within the parenthesis the cost of each method if they were used to generate the sensitivity matrices for history matching. Within an inversion context the effective cost for the linear propagation methods and forward finite differences is discounted by one nonlinear forward solve, which at any rate is needed to calculate the objective function (2). The accounting assumes that the linear problems (14), (17) and (18) are solved separately for each right-hand side. For these small test problems, however, the cost of the forward and backward propagation methods could be nearly independent of number of right-hand sides (N/n) by solving the linear problems using direct linear solvers.

Table 3: Number of Newton-Raphson iterations (NR Iter) and the number of linear matrix solutions needed to evaluate the sensitivity matrix using the linear propagation methods (Dir/Adj), forward finite differencing (Fwd) and central differencing (Cntrl). The results are in all cases using forward simulations with ten fixed time-steps of 10^6 s. Shown are the number of linear matrix problems solved by each method. Note that the numbers within the parenthesis would be the cost per inversion iteration.

	NR Iter	Number of Linear Solves		
		Dir/Adj	Fwd	Cntrl
	N_{IT}	$N_{IT}+N_t\times n$ ($N_t\times n$)	$N_{IT}\times[n+1]$ ($N_{IT}\times n$)	$N_{IT}\times 2n$ ($N_{IT}\times 2n$)
Liquid	22	42 (20)	66 (44)	88 (88)
Flashing	33	53 (20)	99 (66)	132 (132)
Initially Two-Phase	36	56 (20)	108 (72)	144 (144)

For these simple problems the results suggest that we can expect the forward propagation method to result in at least half the computational cost compared with using finite differences for history matching. For two-phase problems the savings can be even greater. Note however that this accounting omits any added memory cost of the linear propagation methods.

5. CONCLUSIONS AND FUTURE WORK

The numerical experiments indicated that either linear forward or backward propagation should be viable options for calculating derivatives of model outputs from geothermal reservoir simulations. Phase transitioning did not make the proposed linear propagation methods any less tractable for calculating derivatives of model outputs. The linear forward and backward propagation methods were found to be in good agreement with each other when evaluating model sensitivities for selected problems. They likewise gave results which were mostly consistent with using finite differentiation to evaluate model sensitivities. The finite difference methods were, however, found to be inconsistent and inaccurate for evaluating small sensitivities. The reliability of the finite difference methods rely heavily on the sound choice of model parameter perturbations. The forward and backward propagating methods offer in comparison a consistent way of finding model sensitivities without having to choose model parameter perturbations.

Encouraged by our preliminary results we plan to test the forward and backward propagation methods for inverting larger simulation problems, with greater variability in block volume sizes and adaptive time-stepping. The tests should include proper measures of any computational savings made by employing the proposed methods. Additionally we aim to adapt the linear propagation methods to natural state modelling.

A. NOTATION

In our mathematical notation, unless stated otherwise, bold faced lowercase letters **x** denote column vectors and bold faced uppercase letters **X** represent matrices. Other letters, whether they are upper- or lowercase, denote scalars.

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". The first author holds an AUEA Braithwaite-Thompson Graduate Research Award and a Landsbankinn scholarship.

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.: Efficient sensitivity computations for automatic geothermal model calibration. *Proceedings 36th New Zealand Geothermal Workshop*. Auckland, New Zealand. (2014).

Doherty, J.: *PEST: Model-Independent Parameter Estimation*. Watermark Numerical Computing, Brisbane. (2013).

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

Finsterle, S.: *iTOUGH2 Command Reference*. Report LBNL-40041, Lawrence Berkeley National Laboratory, Berkeley, California. (2014).

Kim, J.G., & Finsterle, S.: Application of automatic differentiation in TOUGH2. *Proceedings TOUGH Symposium 2003*. Lawrence Berkeley National Laboratory, Berkeley, California. (2003).

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

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., Reynolds, A.C., & Liu, N.: *Inverse Theory for Petroleum Reservoir Characterization and History Matching* (1st ed.). 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. *Proceedings 35th New Zealand Geothermal Workshop*. Rotorua, New Zealand. (2013).

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

Reuther J., Jameson A., Farmer J., Martinelli L., & Saunders D.: Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. Paper 96-0094, presented at the *34th AIAA Aerospace Science Meeting and Exhibit*. (1996).

Wong, Z.Y., Horne, R., & Voskov, D.: A geothermal reservoir simulator with AD-GPRS. *Proceedings World Geothermal Congress 2015*. Melbourne, Australia. (2015).