ACCELERATING THE SOLUTION OF GEOTHERMAL INVERSE PROBLEMS USING ADJOINT METHODS IN WAIWERA: A PROGRESS REPORT

Benjamin Gonzalez-Gutierrez[1], Scott Sung[1], Ruanui Nicholson[1], John P. O'Sullivan[1], Michael J. O'Sullivan[1] and Oliver J. Maclaren[1]

[1]Department of Engineering Science, The University of Auckland, Auckland 1142, New Zealand

oliver.maclaren@auckland.ac.nz

Keywords: *Inverse problems, Waiwera, geothermal modelling, reservoir simulation, adjoint methods.*

ABSTRACT

Mathematical and computational modelling of geothermal fields plays a key role in effective management of geothermal reservoirs. Central to the management of such reservoirs is the ability to predict future behaviour.

In the case of geothermal reservoir modelling, available data consists of noisy temperature, pressure and enthalpy measurements from pre-existing wells. In order to predict future scenarios, however, the real parameters of interest are the underlying permeability structure and the location and/or magnitude of the deep upflow of the reservoir. These parameters are what control the observed and predicted temperature and pressure data.

Mathematically, inferring the permeability and/or boundary conditions given temperature and/or pressure measurements is an ill-posed inverse problem. Typically, solving the inverse problem amounts to determining the minimiser of a suitably constructed cost function.

Given sufficient computational resources, there are well-defined computational approaches to solving inverse problems; however, these methods can require a large number of simulator runs, and large geothermal simulation models are typically computationally expensive. Adjoint methods provide a method of speeding up the solution of inverse problems, provided model Jacobian information (in particular) is available. Adjoint methods can be adapted to take into account both first derivative and second derivative information of the cost function. When using first derivative information (e.g. steepest descent), the cost to determine each search direction is one forward solve and one additional linear adjoint solve.

Here we give an overview of preliminary work on implementing and using the adjoint method to solve inverse problems for geothermal models constructed in the new simulator Waiwera. Waiwera is based on the same robust algorithm as (AU)TOUGH2, with the added benefits of a fully parallel code, additional solvers and, most importantly, easy access to model Jacobians.

1. INTRODUCTION

Geothermal modelling plays an important role in modern geothermal engineering and resource modelling. There is typically a trade-off, however, between constructing and simulating models that provide results that are more realistic and obtaining these results in a sufficiently timely manner to assist decision-making. These conflicting requirements of accuracy and speed continue to drive new research in geothermal modelling; this is especially true of so-called 'inverse' modelling, which is usually even more computationally demanding than the usual 'forward' simulation problems. This is due to the need to carry out a number of forward model simulations to solve just a single inverse problem.

This paper describes our progress so far in implementing the adjoint state method for models constructed in the newly developed Waiwera geothermal simulator, a study aimed at reducing the computational cost of solving geothermal inverse problems.

1.1 Geothermal modelling and inverse problems

Geothermal modelling involves a series of steps that typically result in a computational model that requires calibration to data (O'Sullivan *et al.*, 2001). Calibration consists of using measured data to determine parameter sets that are compatible with the given data; these parameters are either directly of interest, or of interest to enable good future predictions to be made. Another name for this process is 'inverse modelling', or solving an 'inverse problem'. A mathematical formulation of inverse problems is as constrained/penalised/regularised optimisation problems, for example:

$$\begin{aligned} &\min \quad f(\boldsymbol{u};\boldsymbol{p}) \\ &\text{such that} \quad \boldsymbol{g}(\boldsymbol{u};\boldsymbol{p}) = 0. \end{aligned} \tag{1}$$

Here f is the objective function we wish to minimise, $\boldsymbol{u}$ are the measurable quantities, $\boldsymbol{p}$ are the parameters we wish to infer, while the constraint $\boldsymbol{g}(\boldsymbol{u};\boldsymbol{p}) = \boldsymbol{0}$ represents the governing equations, i.e., nonlinear mass and energy transport equations and Darcy's law that must be satisfied.

The objective function, f, is often, but not always, in the form of a regularised sum of squares, i.e.

$$f(\boldsymbol{u};\boldsymbol{p}) = \|\boldsymbol{W}_1(\boldsymbol{B}(\boldsymbol{u}) - \boldsymbol{d}_{obs})\|^2 + \alpha\|\boldsymbol{W}_2(\boldsymbol{p} - \boldsymbol{p}_*)\|^2, \tag{2}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are suitably chosen weighting matrices (or possibly a derivative operator in the case of $\boldsymbol{W}_2$), $\boldsymbol{B}(\cdot)$ is the observation operator, $\boldsymbol{d}_{obs}$ is the observed data, α is the regularisation parameter, and $\boldsymbol{p}_*$ is a reference value for the parameters (Aster et al., 2005).

In general, regularisation is required as the inverse problem is *ill posed* – there are typically large numbers of solutions with as good, or nearly as good, a fit to the observed data and this can introduce numerical instabilities; regularisation introduces a preference for simple, stable and/or smooth solutions. Choosing the regularisation parameter α amounts to making a choice between the relative importance of fitting the given data and having a simple, stable or smooth model; this is somewhat of an art, but a number of methods exist including both *a priori* methods and data-based methods (Aster et al., 2005; Kaipio and Somersalo, 2006).

The trade-off between fitting the data and prior preferences for parameter values has an interpretation in terms of Bayesian statistics (see, for example, Kaipio and Somersalo, 2006; Maclaren *et al.*, 2016; Stuart, 2010; Tarantola, 2006). In this view, the solution of the optimisation problem represents the MAP – maximum a posteriori – solution. We keep the discussion here as general as possible, and will not assume an explicit form of the objective function f unless stated otherwise.

1.2 Solving the optimisation problem

Two important classes of methods used to solve the optimisation problems associated with inverse problems are derivative-based and derivative-free methods.

Although gradient-free methods, such as the Nelder-Mead simplex method (Wright and Nocedal, 1999), are applicable in a range of scenarios, such methods require a large number of forward simulations. Due to the nonlinearity of the forward problem in geothermal simulations (and the need for time-marching in natural state scenarios), however, even a single forward simulation can be (extremely) time consuming. For this reason, gradient-based optimisation methods, such as gradient/steepest descent and/or higher order derivative methods, such as Gauss-Newton, Levenberg-Marquardt, or other quasi-Newton algorithms are desirable (Wright and Nocedal, 1999; Boyd and Vandenberghe, 2004).

The standard approach to computing gradients and higher order derivatives such as Jacobian matrices is to use finite differencing (Wright and Nocedal, 1999); unfortunately, finite differencing requires as many forward solves at each step of the optimisation as there are parameters. Thus for cases involving a large number of parameters, e.g. a large number of rock types, the computational time required to carry out the optimisation can be prohibitive. For example, computing the (total) derivative of the objective function with respect to the parameters using forward differencing requires solving

$$\frac{df(\boldsymbol{u};\boldsymbol{p})}{d\boldsymbol{p_i}} \approx \frac{f(\boldsymbol{u}(\boldsymbol{p}+h\boldsymbol{e_i});\boldsymbol{p}+h\boldsymbol{e_i}) - f(\boldsymbol{u}(\boldsymbol{p});\boldsymbol{p})}{h} \qquad (3)$$

for each i, where h is the step size, $\boldsymbol{e_i}$ is the unit vector in the i^{th} direction and $\boldsymbol{u}(\boldsymbol{p}+h\boldsymbol{e_i})$ is the solution to the governing equations under a parameter perturbation in the i^{th} component, i.e. $\boldsymbol{u}$ satisfying $\boldsymbol{g}(\boldsymbol{u};\boldsymbol{p}+h\boldsymbol{e_i})=0$. One solve of the unperturbed equation $g(\boldsymbol{u};\boldsymbol{p})=0$ is also required. Thus, this method requires N_p+1 forward simulations to compute, where N_p is the number of parameters, which can quickly become too expensive for problems involving large numbers of parameters.

In addition to computational costs, finite differencing can give inaccurate approximations to derivatives (Wright and Nocedal. 1999); attempting to improve accuracy by using higher-order differencing increases computational costs further. Nevertheless, finite difference methods are useful for checking the implementation of alternative methods, and are straightforward to implement.

An alternative to the use of finite differencing to calculate gradients and Jacobians is the use of the so-called adjoint state method, or simply the adjoint method (Bjarkason *et al.*, 2014, 2015, 2016a, 2016b; Cao *et al.*, 2003; Granzow, 2014; Gunzburger, 2003). The adjoint method has been successfully applied in various subsurface modelling communities, but is less common in geothermal applications. Recently, however, Bjarkason *et al.* (2014, 2015, 2016a, 2016b) successfully employed the adjoint method in geothermal modelling using (AU)TOUGH2. We discuss this approach in detail in Section 2 below. Given information typically available from a single run of the forward solver, along with additional analytical derivatives, the adjoint method allows one to compute the derivative of the objective function with respect to the parameters without additional forward model simulations, and at a cost independent of the number of parameters (Bradley, 2013).

1.3 Waiwera

Our current research aims at bringing together the work of Bjarkason *et al.* (2014, 2015, 2016a, 2016b) on adjoint methods in a geothermal context with the newly developed geothermal simulator Waiwera. A key aim is to obtain a stable and re-usable implementation of the adjoint method for other researchers to use. Waiwera was developed, and continues to be developed, at the University of Auckland with the hope that it will supersede (AU)TOUGH2 and its role in geothermal modelling. Here (AU)TOUGH2 is used to represent either TOUGH2 (Pruess *et al.*, 1999) or the University of Auckland's customised version AUTOUGH2 (Yeh *et al.*, 2012). The main computational benefits of using Waiwera for forward simulations are the parallel code and access to additional solvers.

Waiwera has been evaluated against various benchmark test cases, based on the Geothermal Model Intercomparison Study (Croucher *et al.*, 2016, 2018). These include a 3-D reservoir flow simulation and full-scale simulations of a model of the Ngawha geothermal system. In a majority of benchmark tests, Waiwera performed equally or better than the industry-standard (AU)TOUGH2, in terms of scalability to large problems and rapid solution convergence. For example, some models which required days of simulation time in (AU)TOUGH2 could be solved in hours using Waiwera (Croucher *et al.*, 2018).

In the context of automated model calibration, perhaps the biggest benefit to using Waiwera is the ease of access to model Jacobian matrices, which are required in the adjoint method (see Section 2 below). This reduces the need for manual intervention, which is generally required to carry out the same calculations in (AU)TOUGH2. Ultimately, we aim to incorporate the ability to carry adjoint calculations as automatically as possible, either directly into Waiwera or into supporting software packages.

2. THE ADJOINT METHOD

The adjoint method is an analytic alternative to finite differencing for calculating derivatives. There are several ways of deriving the adjoint method (see for example Granzow, 2014). Here we consider the method of Lagrange multipliers.

2.1 Lagrange multiplier method

The first step in the method of Lagrange multipliers is to introduce a *Lagrangian function*, or simply a *Lagrangian*, which incorporates the constraints into a modified objective function to give an unconstrained optimisation problem (Boyd and Vandenberghe, 2004). Here the Lagrangian is given by

$$\mathcal{L}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{\lambda}) \equiv f(\boldsymbol{u};\boldsymbol{p}) + \boldsymbol{\lambda}^T\boldsymbol{g}(\boldsymbol{u};\boldsymbol{p}), \qquad (4)$$

where $\boldsymbol{\lambda}$ are the so-called Lagrange multipliers or *adjoint variables*. Although $\boldsymbol{g}$, $\boldsymbol{u}$, $\boldsymbol{p}$ and $\boldsymbol{\lambda}$ are all vectors, to keep notation tidy from now on we will simply denote them by g, u, p and λ, respectively. Similarly, we will drop all other bolding of terms in the following. The method of Lagrange multipliers states that at a constrained minimum for f we must have each of the derivatives, $\mathcal{L}_\lambda$, $\mathcal{L}_u$, and $\mathcal{L}_p$ equal to zero. That is, we require

$$\begin{aligned} \frac{d\mathcal{L}}{d\lambda} &= g^T = 0 \\ \frac{d\mathcal{L}}{du} &= \lambda^T \frac{\partial g}{\partial u} + \frac{\partial f}{\partial u} = 0 \\ \frac{d\mathcal{L}}{dp} &= \lambda^T \frac{\partial g}{\partial p} + \frac{\partial f}{\partial p} = 0, \end{aligned} \tag{5}$$

where we have used the convention that derivatives of scalar-valued functions with respect to column vectors are row vectors. The first equation of the system (5) simply requires the forward problem be satisfied. The second equation is equivalent to

$$\left(\frac{\partial g}{\partial u}\right)^T \lambda = -\left(\frac{\partial f}{\partial u}\right)^T, \tag{6}$$

which is known as the adjoint equation. This equation is a linearised adjoint (or transpose) version of the forward problem, which is solved for λ so that it can be used in the final equation in the system (5). This equation can be solved efficiently in highly-parameterised problems as the dimensions of the vectors and matrices involved are independent of the number of parameters, only depending on the dimension of the state variables u, i.e. N_s.

The $\frac{\partial f}{\partial u}$ term in (6) can usually be worked out analytically; for example in the case where f is as in (2), and the observation operator $B(\cdot)$ is linear in u, i.e., $B(u) = Bu$, then

$$\frac{\partial f}{\partial u} = 2B^T W_1^T W_1 (Bu - d_{obs}). \tag{7}$$

On the other hand, Waiwera automatically provides the more complicated $\frac{\partial g}{\partial u}$ term, since this is required in the solution of the nonlinear forward problem. In principle this term is also available from (AU)TOUGH2, but it is less accessible and requires some manual intervention (as was done by Bjarkason *et al.*, 2014, 2015, 2016a, 2016b). In contrast, obtaining this term from Waiwera simply requires running it with an additional flag (see below).

The final equation of the system (5), i.e.

$$\frac{d\mathcal{L}}{dp} = \lambda^T \frac{\partial g}{\partial p} + \frac{\partial f}{\partial p} = 0, \tag{8}$$

states that the total derivative (gradient) of $\mathcal{L}$ with respect to parameters is equal to zero at an optimal solution. When λ is determined as in (6) above, and the model constraint is enforced as in the first term in (5), it can be shown that the total derivative of $\mathcal{L}$ above is the same as the total derivative (gradient) of f along directions satisfying the model constraint, i.e. that

$$d_p f = \frac{df(u(p); p)}{dp} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial p} + \frac{\partial f}{\partial p} = \lambda^T \frac{\partial g}{\partial p} + \frac{\partial f}{\partial p} \tag{9}$$

for $u(p)$ satisfying $g(u, p) = 0$ and λ satisfying (6). Hence $\mathrm{d}_p f$ is also equal to zero at an optimal solution, as expected.

The main output of the adjoint method is the (negative of the) above term, $-\mathrm{d}_p f$, and possibly higher-order derivative terms, for use in descent-based optimisation algorithms (see below). For simplicity we will focus here on methods which only require $-\mathrm{d}_p f$, but the adjoint method extends naturally to higher derivatives.

The benefit of the adjoint approach to the calculation in (9), other than avoiding the use of finite differencing, is that by using the right-most expression we can avoid calculating $\frac{\partial u}{\partial p}$. This term is often (extremely) difficult and/or expensive and has dimensions $N_s \times N_p$, i.e. number of state variables times number of parameters.

Instead, and assuming that we first solve the adjoint problem (6) for λ, all that remains in order to be able to directly compute $\mathrm{d}_p f$ in (9) are the partial derivatives $\frac{\partial g}{\partial p}$ and $\frac{\partial f}{\partial p}$. These are available analytically, though the $\frac{\partial f}{\partial p}$ term is more straightforward; assuming the objective given in (2), this is simply

$$\frac{\partial f}{\partial p} = 2\alpha W_2^T W_2 (p - p_*). \tag{10}$$

The partial derivative of the model function with respect to parameters is more involved, but in principle straightforward. Bjarkason *et al.*, (2014, 2015, 2016b) outline these terms in detail in the context of (AU)TOUGH2; Waiwera solves the same governing equations and so the terms given there are the same as used here (but see the Discussion section). A future possibility is to also access these terms directly from Waiwera. This would make the approach even more accessible to the general user, as well as reducing the possibility of user error.

2.2 Descent algorithms: steepest descent and BFGS

The basic element of all descent algorithms for minimising a general function $f(x)$ is a method of choosing a sequence of $x_1, x_2, \ldots$ values for which $f(x_{k+1}) < f(x_k)$ for each x_k unless x_k is optimal (Boyd and Vandenberghe, 2004). The method of choosing the next member of the sequence amounts to choosing a *descent* or *search* direction.

Assuming we use the Euclidean norm, the steepest descent optimisation algorithm simply updates the parameter p using the search direction equal to the negative gradient $-\mathrm{d}_p f$, that is the direction in which f decreases the fastest (Boyd and Vandenberghe, 2004). This is also known as gradient descent. Alternative algorithms use different descent directions; these can be based on different norms and/or use (exact or approximate) higher derivative information.

In this paper we employ the the Broyden, Fletcher, Goldfarb and Shanno (BFGS) quasi-Newton algorithm (Wright and Nocedal. 1999). The BFGS method does not explicitly require second order derivative information, nor does it require the Jacobian matrix, $\frac{\partial u}{\partial p}$, which is required for the Gauss-Newton and Levenberg-Marquardt methods. Instead, the BFGS method achieves super-linear convergence by making successively improved approximations to the Hessian matrix at each iteration using only first derivative information.

The general routine for applying the BFGS algorithm using the adjoint method is as follows:

1. Initialise parameter value, p.
2. Solve forward problem for u.
3. Solve adjoint equation for λ.
4. Calculate the steepest descent direction $d_p f$.
5. Build an approximate Hessian, $\tilde{H}$, using previous BFGS updates
6. Solve $\tilde{H}s = d_p f$ for the BFGS descent direction s and update p in this direction.
7. Repeat from 2 until converged.

3. IMPLEMENTATION

3.1 Adjoint method with Waiwera

Here we give a basic overview of how to implement the adjoint method using the Waiwera simulator for calibrating a natural state geothermal reservoir model. The general routine consists of the same five key steps listed above. However, only steps 2 and 3 involve Waiwera. In terms of Waiwera, the two key steps are:

1. Run Waiwera on a given set of values for the permeabilities and deep sources. This involves modifying a .json file to update parameter values.
2. Obtain output from Waiwera. This includes the predicted measurements from a .h5 file, and the Jacobian matrix, $\frac{\partial g}{\partial u}$.

When the objective function is as in equation (2) and observation operator is linear, the difference of the predicted measurements and the measured data is used as a forcing term for the adjoint equation. The adjoint variables, λ, are then solved for, i.e.

$$\lambda = -2\left(\frac{\partial g}{\partial u}\right)^{-T} B^T W_1^T W_1 (Bu - d_{obs}), \qquad (11)$$

although this is carried out using efficient linear solvers rather than direct matrix inversion. It is worth noting that the Jacobian matrix, $\frac{\partial g}{\partial u}$, is initially returned from Waiwera as a binary file which must be converted to an array. Furthermore, due to the parallel processing in Waiwera, special care is required when ordering the array.

After solving for the adjoint variables, the steepest descent direction is calculated. This can be used in a standard optimisation algorithm, either hand-coded or from a library. Here we use the BFGS implementation from the scipy.optimize library in SciPy (Jones *et al.*, 2001), a Python library of tools for computational science and engineering; given the derivative $-\mathrm{d}_p f$, this updates the parameters using the BFGS algorithm as (briefly) described above.

3.2 Test problem

A medium-sized 2D model is being used as a case study for this project, and is shown in Figure 1. The model is single-phase, steady-state, has 2 deep sources, 340 blocks and 6 different rock types. This model is based on that used by Bjarkason *et al.* (2016b) and Maclaren *et al.* (2016).

This model contains 14 parameters to be calibrated: 2 permeabilities for each of the 6 rock types, as well as the mass flow rates of the 2 deep sources. At this stage of the project, however, we have only carried out inversion for the rates of the two deep sources: $Q_{m,1}$ and $Q_{m,2}$ (i.e. we have assumed known rock permeabilities for the time being). We have implemented a preliminary version of the inversion for permeabilities, but this is not yet ready for proper testing. In particular, the analytical derivatives required to invert for these terms have been more difficult to verify.

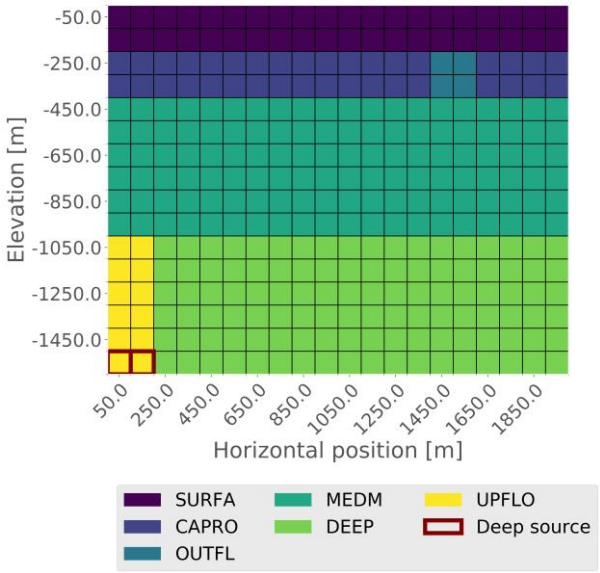

Figure 1: Medium-sized model used for testing.

Artificial observed data, d_{obs}, was generated by running Waiwera using a set of 'true' parameters, p_{true} (see Table 1). We added Gaussian noise to the temperature observations with a standard deviation of 0.5°C. This provides synthetic data to be used to recover estimates of the underlying parameters.

Given the above synthetic data, the inversion problem (1) was solved with an unregularised least-squares data objective function f (which was sufficient for the small number of parameters considered). A randomised initial parameter guess, p_0, was generated by taking p_{true} and multiplying it by random numbers ranging from 0.8 to 1.2 (see Table 1). This generates initial guesses that are quite close to the truth; this was sufficient for our present purposes, i.e. for an initial verification of the algorithms.

Table 1: True value and initial guess for parameters in inversion.

Parameter	$Q_{m,1}$	$Q_{m,2}$
True value	0.0750	0.0750
Initial guess	0.0708	0.0779

4. PRELIMINARY RESULTS

In this experiment, we compared the use of a derivative-based optimisation method, the BFGS algorithm, using the adjoint method, against a derivative-free optimisation method, the Nelder-Mead method. The aim was to investigate whether the adjoint method gives reasonable results, and to measure the efficiency gain for an inverse problem with a small number of parameters.

The main difficulty when comparing methods is that they have different stopping criteria, and thus it is hard to set an equivalent stopping condition for both. Nonetheless, it is possible to manually tune the tolerances for each algorithm so that they reach a similar accuracy. Table 2 shows the results we obtained.

Table 2: Results from model inversion using two different methods.

Method	Parameters found	Function evaluations	Time taken (in s.)
BFGS with Adjoint	0.07500162, 0.07499781	36	345.13
Nelder-Mead	0.0750000, 0.0750000	109	1014.99

These results suggest two key findings. Firstly, the methods agree well with each other and with the true values of the parameters. This provides some initial validation of the functionality of the adjoint method implemented. Secondly, the adjoint method completes the inversion in a significantly shorter amount of time than the derivative-free method: almost three times as fast. Thus, although this is only a very simple test case, we do observe the expected efficiency gains from using the adjoint method. These gains are expected to be much higher when performing inversions involving a larger number of parameters.

In addition to the above experiments, we have successfully verified the derivative calculations (of the objective function with respect to parameters) using the adjoint method against finite differencing. We have not yet, however, compared the computational speed-up of the adjoint method compared to gradient methods based on finite differencing.

5. DISCUSSION AND LIMITATIONS

5.1 Permeability inversion

We have successfully implemented and verified derivatives with respect to the deep sources, both with finite differencing and on a simple test problem. The permeability terms have proven more difficult, however. Since we are unsure of the correctness of these terms, we have not presented any permeability results here.

There are multiple possible sources of the inaccuracy in the permeability calculations; we suspect that there may be an issue with the derivation and/or implementation of the analytical model derivatives $\frac{\partial g}{\partial p}$. At the moment these terms are based on those for (AU)TOUGH2, since the same governing equations are solved, but implementation differences may lead to some discrepancies. Since we are solving the steady-state problem we are also neglecting transient terms, but these may not be exactly zero at the end of model simulations and hence we may need to account for these explicitly.

We are currently addressing these issues with the aid of the Waiwera team to ensure accuracy of the governing equation calculations and investigating other possible error sources such as unit precision and consistency.

5.2 Geometry, dimensions and model information

We are currently only running a simple, square two-dimensional test case. In principle, the code developed is directly applicable to three-dimensional problems, but we have not yet tested this.

An additional limitation is that we have used a simple uniform mesh block; while this is not a limitation in terms of Waiwera simulations, block neighbour structure is required for calculating the analytical derivatives $\frac{\partial g}{\partial p}$. Complex mesh designs require access to this structure or, alternatively, we could attempt to return the extra model derivative $\frac{\partial g}{\partial p}$ directly from Waiwera.

In principle, we can extract much more information directly from Waiwera than we currently are, including the flux values and block connection information. As mentioned, this may eliminate some of the potential issues with calculation errors and inaccuracy that we are currently facing, especially with respect to the $\frac{\partial g}{\partial p}$ term. Again, we are currently discussing these issues with the Waiwera team and we hope this information will be available in the future.

5.4 Transient state implementation

We are currently only implementing steady-state inversions. In principle, the same adjoint method ideas are applicable to transient problems, but this would require additional modifications to our existing code.

6. CONCLUSIONS

The overall goal of this project is to accelerate the solution of geothermal inverse problems using a reproducible and easy-to-use implementation of the adjoint method for models constructed in Waiwera. We have presented preliminary progress towards these goals in the present report. While our results are preliminary, and restricted to simple test cases, we have carried out successful inversions for deep source terms and verified potential speed-up from using the adjoint method in terms of computational time required relative to derivative-free methods. Key next steps include: verifying the permeability derivatives (if possible by obtaining the additional term $\boldsymbol{\frac{\partial g}{\partial p}}$ directly from Waiwera), carrying out successful permeability inversions, comparing the speed-up to derivative-based optimisation using finite differencing, and implementing the approach for higher-dimensional problems and/or more complex geometries. If these steps are successful then we would anticipate that efficient derivative-based inversion methods using Waiwera would be much more widely accessible to the geothermal community.

ACKNOWLEDGEMENTS

The authors appreciate the contribution of the NZ Ministry of Business, Innovation and Employment for funding parts of this work through the grant: C05X1306 “Geothermal Supermodels”. The authors would also like to acknowledge a number of members of the Geothermal Institute of the University of Auckland for their help with Waiwera and adjoint methods, especially Adrian Croucher, Angus Yeh and Elvar Bjarkason.

REFERENCES

Aster, R. C., Borchers, B., & Thurber, C. H. (2011). *Parameter estimation and inverse problems* (Vol. 90). Academic Press.

Bjarkason, E. K., O'Sullivan, M. J., & O'Sullivan, J. (2014). Efficient sensitivity computations for automatic geothermal model calibration. In *Proceedings, 36th New Zealand Geothermal Workshop*. Auckland, New Zealand.

Bjarkason, E. K., O'Sullivan, M. J., & O'Sullivan, J. (2015). Improved sensitivity calculations. In *Proceedings, 37th New Zealand Geothermal Workshop.* Auckland, New Zealand.

Bjarkason, E. K., O'Sullivan, J. P., Yeh, A., & O'Sullivan, M. J. (2016a). Combined natural state and history matching using the adjoint or direct sensitivity method. In *Proceedings of the 38th New Zealand Geothermal Workshop.* Auckland, New Zealand.

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

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Bradley A. M. (2013). PDE-constrained optimization and the adjoint method, http://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf. [Tutorial notes; accessed 01-04-2018].

Croucher, A., O'Sullivan, M. J., O'Sullivan, J., Pogacnik, J., Yeh, A., Burnell, J., & Kissling, W. (2016). Geothermal Supermodels Project: an update on flow simulator development. In *Proceedings of the 38th New Zealand Geothermal Workshop.* Auckland, New Zealand.

Croucher, A., O'Sullivan, J., Yeh, A., & O'Sullivan, M. (2018). Benchmarking and experiments with Waiwera, a new geothermal simulator. In *Proceedings, 43rd Workshop on Geothermal Reservoir Engineering*. Stanford University, Stanford, California.

Granzow, G. D. (2014). A tutorial on adjoint methods and their use for data assimilation in glaciology. *Journal of Glaciology*, 60(221), 440-446.

Gunzburger, M. D. (2003). *Perspectives in flow control and optimization* (Vol. 5). SIAM.

Jones E, Oliphant E, Peterson P, et al. (2001). SciPy: Open source scientific tools for Python, http://www.scipy.org/ [Online; accessed 01-08-2018].

Kaipio, J., & Somersalo, E. (2006). *Statistical and computational inverse problems* (Vol. 160). Springer Science & Business Media.

Maclaren, O. J., Bjarkason, E., O'Sullivan, J., & O'Sullivan, M. J. (2016). Inverse modelling of geothermal reservoirs – a hierarchical Bayesian approach. In *Proceedings of the 38th New Zealand Geothermal Workshop*. Auckland, New Zealand.

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

Pruess, K., Oldenburg, C. M., & Moridis, G. J. (1999). *TOUGH2 user's guide version 2*.

Stuart, A. M. (2010). Inverse problems: a Bayesian perspective. *Acta Numerica*, 19, 451-559.

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation* (Vol. 89). SIAM.

Tröltzsch, F. (2010). *Optimal control of partial differential equations: theory, methods, and applications* (Vol. 112). American Mathematical Society.

Wright, S., & Nocedal, J. (1999). *Numerical optimization*. Springer, 35(67-68), 7.

Yeh, A., Croucher, A. E., & O'Sullivan, M. J. (2012). Recent developments in the AUTOUGH2 simulator. In *Proceedings, TOUGH Symposium*. Lawrence Berkeley National Laboratory, Berkeley.