Proceedings World Geothermal Congress 2010
Bali, Indonesia, 25-29 April 2010

Characterization of Fractures in Geothermal Reservoirs

Egill Juliusson and Roland N. Horne

Stanford University, Department of Energy Resources Engineering, 367 Panama Street, Stanford, CA 94305-2220, USA

egillj@stanford.edu

Keywords: Fracture, fracture network, deconvolution, well-to-well interaction, reservoir simulation.

ABSTRACT

It is widely agreed that the flow through geothermal reservoirs is highly fracture-dominated. This has led our research down a path of developing better methods for understanding the fracture networks in the reservoir.

In a broad sense, we would like to address questions regarding the extent to which fractures in the subsurface can be characterized. What are some of the more important properties of a fracture network? How much can actually be said about a fracture network given various types of field data? Can we generate realistic numerical simulations of the process? What types of data are most relevant to defining the various characteristics of a fracture network?

In an attempt to answer these questions we constructed stochastic fracture network models. Then, simulations with a hydrothermal reservoir simulator (TOUGH2) were used to generate various types of synthetic data.

Parts of these data, tracer returns in particular, were further processed to characterize the response in production to tracer injection. The response was characterized by a finite impulse response (kernel) found using a deconvolution technique based on both convex and direct search optimization methods.

So far we have shown that a general purpose reservoir simulator can be configured to simulate heat, mass and solute flow through a discrete fracture network. Moreover we have successfully extracted well-to-well transfer functions (kernels) that describe tracer transport through the fracture network quite accurately, given steady state flow conditions.

1. INTRODUCTION

Some of the key questions that a geothermal reservoir engineer has to address have to do with the predicted size of the resource, and variations in productivity of wells. Further plans for development are based upon this information, e.g. the estimated size of the power plant to be built, the project lifetime and the number of wells and make-up wells required.

The relationship between the wells and the resource is hard to determine because geothermal systems are generally highly heterogeneous with fracture-dominated flow. Tracer tests are commonly performed to gain an understanding of the well-to-well interaction, and various estimates can be made from the return curves. For example, the total reservoir volume and the efficiency of heat recovery can be estimated, which is very important for reducing uncertainty in volumetric Monte Carlo models. These are commonly used e.g. by the USGS for the National Geothermal Resource Estimate (Williams et al., 2008), and in the early stages of geothermal project development. At later stages, tracer return curves can be used, e.g. to predict the expected decline in production temperature with time, and the allowable increase in energy production from underutilized reservoirs, as shown by Axelsson et al. (2001).

Lovekin and Horne (1989) illustrated the usefulness of well-to-well interaction data for optimizing reinjection scheduling. The essence of the approach was to minimize the field-wide risk of thermal breakthrough. This required an estimate of a connectivity parameter quantifying the risk of breakthrough between each injector-producer pair in the field. Multiple parameters were suggested for quantifying the connectivity, many of which would be obtained from tracer tests, e.g. initial and peak return time (t_i and t_p), peak return concentration (C_p) and cumulative tracer return (f), see Figure 1. Given the connectivity parameter, the scheduling problem could be set up as a constrained quadratic program which was solved to find the optimal injection and production rates.

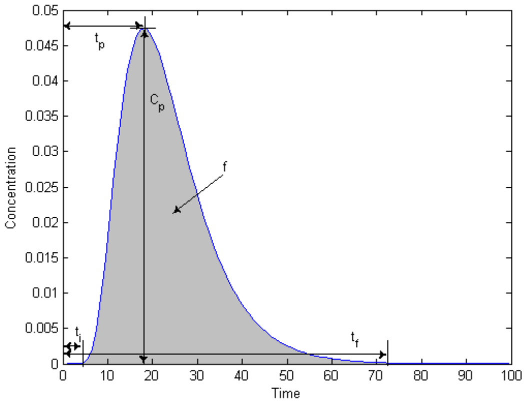

Figure 1: A tracer return curve and some relevant measures.

A drawback to the approach of Lovekin and Horne was associated with the difficulty in obtaining the connectivity data. These could be obtained by performing tracer tests for each injection well. That is, however, a nontrivial task and requires either the use of different types of tracer for each injection well, or waiting a long time (years) between tests on each individual injection well. Finding a way to determine the origins of a tracer signal based solely on the transients of the input and output is one of the goals of this work, thus allowing the application of tracer tests on multiple wells simultaneously with only one type of tracer. Situations where such data might be available could arise where natural tracers (e.g. chloride produced in brine) are constantly being injected as part of the circulation process in the geothermal power cycle. Figure 2 has an example of such data, collected from the Palinpinon geothermal field in the Philippines.

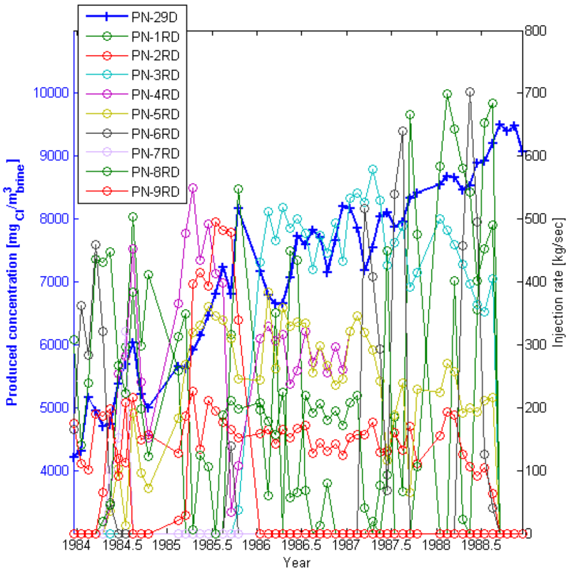

Figure 2: History of injection and produced chloride concentration in well PN-29D, in the Palinpinion field, Philippines.

Macario (1991), Sullera and Horne (2001) and Horne and Szucs (2007) worked on revealing relationships between injection and production in the Palinpinon data set, with moderate success. High well-to-well correlation in the Palinpinon data may have been especially hard to find, since the flow conditions in the reservoir were constantly being changed, thereby altering the well-to-well flow units. Another consideration that was largely disregarded by these previous studies is that there might be a time shift between a signal generated at the injector and the corresponding response in the producer. The signal will also encounter some degree of dispersion.

This paper discusses a deconvolution approach which takes the some of these effects into account (i.e. time lag and dispersion), without assuming much else about the outcome. Similar approaches, using pressure data, have been discussed by Levitan (2007) and Lee et al. (2008, 2009). We do not have an actual data set with sufficient temporal resolution to validate the method. Therefore, synthetic data were generated using the TOUGH2 reservoir simulator. In these simulations, fractures were modeled as discrete blocks, which allowed us to capture some of the characteristics of tracer returns often seen in the field.

Our approach still has some limitations, e.g. predictions with the deconvolution approach depend on the flow field being close to steady state. Moreover, the matrix to fracture heat transfer and diffusion has not been well captured in our flow simulations. However, with some moderate improvements, this work does seem applicable to many cases, in particular Enhanced Geothermal Systems (EGS) and other liquid dominated systems.

2. RESERVOIR SIMULATIONS WITH DISCRETE FRACTURE NETWORKS

This chapter discusses the process of setting up flow simulations through discrete fracture networks (DFNs). It provides insight into flow through fractured reservoirs and helps understand the limitations and capabilities of commonly available software.

2.1 Setting up a Discrete Fracture Network Reservoir Model for TOUGH2

This section describes reservoir simulations that were set up to simulate flow through fractured medium. The fractures are simulated as thin rectangular blocks with high permeability and porosity. The porous ("non-fracture") medium is broken into triangular matrix blocks which conform to the previously generated fracture network. Some numerical manipulation was required to generate a list of reasonable transmissibilities between elements, and the TOUGH2 simulator could be slightly "tricked" to get the desired result, as shown by McClure (2009). The simulation involves a calculation of all relevant thermodynamic properties and the mixing of water and tracer (i.e. "two-waters"). The theoretical grounds for the DFN approach taken here were introduced by Karimi-Fard et al. (2003).

The first step in setting up a discrete fracture network simulation was to generate a discrete fracture network. A number of software packages are available for creating these in 3D, e.g. FRACMAN from Golder & Associates and FRACA from Beicip-Franlab. In order to keep the exercise simple a two-dimensional network was created in MATLAB with a code loosely based on the geomechanical process of fracture formation. Further discussion on the code can be found in Juliusson and Horne (2009a).

After the fracture network had been generated, a suitable computational grid was formed. This was accomplished using the open source mesh generator Triangle, written by Shewchuk (1996). The program and all associated commands and files are well described on the Triangle website, http://www.cs.cmu.edu/~quake/triangle.html.

Figure 3 shows a stochastic fracture network with a conforming triangular grid. The fractures were given a porosity value of 0.9 and randomly assigned a width, *w*, of 1.0, 0.8 or 0.6 mm; the corresponding permeability was determined by

$$k = \frac{w^2}{24} \tag{1}$$

The matrix blocks were set to porosity of 0.12 and permeability 1.0 md (10^{-15} m^2).

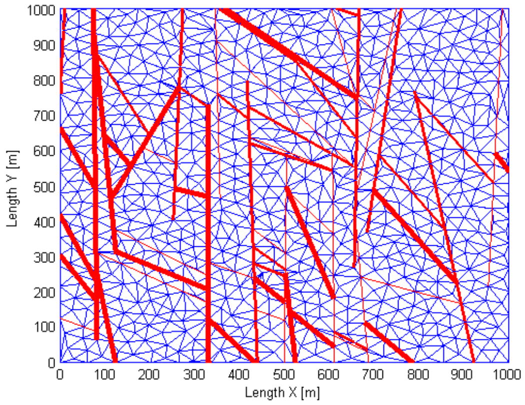

Figure 3: An example of a synthetic fracture network (red) and a mesh (blue) that conforms to the fractures. The fracture width is representative of the permeability assigned.

Each of the elements created by the mesh (i.e. the triangles and fracture segments) was assigned a transmissibility to enable the use of the mesh with a general reservoir simulator. This was accomplished using a specialized computer code developed and described by Karimi-Fard et al. (2003). The transmissibility values obtained are related to the flow between two adjoining elements, *i* and *j*, as

$$Q_{ij} = T_{ij}(p_j - p_i) \tag{2}$$

where *Q* is the flow rate, *T* is the transmissibility, and *p* is the pressure in the gridblock.

The TOUGH2 simulator is not equipped to handle transmissibility terms as input data. To account for that, the permeability of the gridblocks was set to 1 m^2 in the ROCKS section. Then, in the CONNE section listing the properties of element connections, the distances D1 and D2 were each set to 0.5 and the AREAX values were set equal the previously calculated transmissibility values, T_{ij}. This way we were able reconfigure TOUGH2 to perform two-dimensional horizontal simulations. To be able to take molecular diffusion effects into account, which are based on the spatial gradient, the correct distances (D1 and D2) could have been supplied. In this case the transmissibility values should also have been modified by multiplication with $D_{ij}=D_i+D_j$.

2.2 Case Study: Injection of Variable Amounts of Tracer

Following is a description of a simulation case that was created to illustrate the performance of this set up. The simulation was carried out on a two-dimensional horizontal grid with dimensions 1000x1000x200 m^3. The boundaries were modeled as closed (no-flow). Three injectors were configured to inject water each at 10 kg/sec with enthalpy 500 kJ/kg. Two production wells were modeled to deliver against a bottomhole pressure of 30 bar with productivity index of 4x10^{-12} m^3 (as specified for TOUGH2). The well configuration is shown in Figure 4.

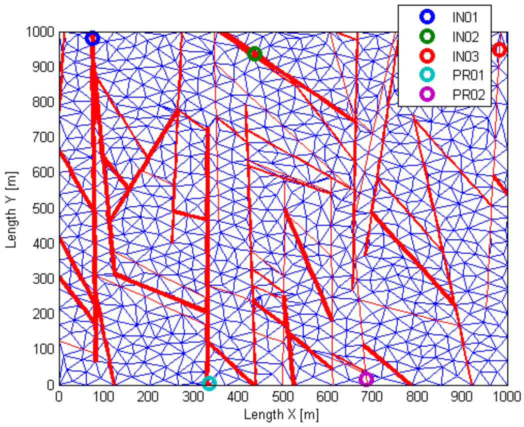

Figure 4: A two dimensional DFN simulation scenario with three injectors (IN01-03) and two producers (PR01 and PR02).

The initial conditions were set to 40 bar, 230 °C, and tracer mass fraction 10^{-10} kg$_{tracer}$/kg$_{total}$ (the tracer mass was not zero because the simulator had problems with that initial condition). Then the production was allowed to continue for about 250 days, or until the production pressure and temperature in the wells had stabilized. Figures 5 and 6 illustrate this steady state condition for pressure and temperature, respectively. The steady state temperature found in the production wells was overpredicted because the discretization was not fine enough to adaquately capture the cooling of the fracture walls.

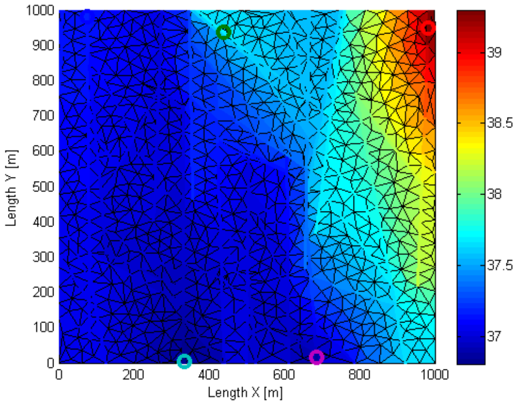

Figure 5: Pressure distribution (steady state) for the DFN simulation scenario.

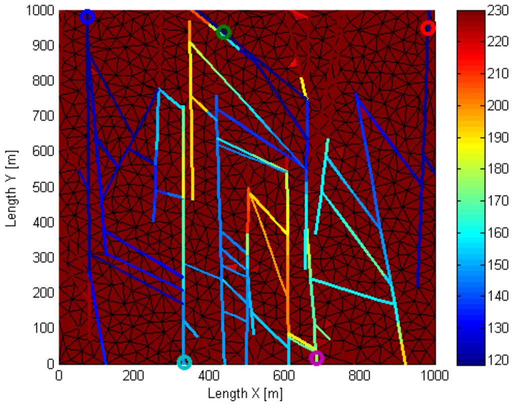

Figure 6: Temperature distribution (steady state) for the DFN simulation scenario.

A second simulation was run, starting with the steady state conditions described above and the same injection and production rates. In addition to that, a small, random fraction of tracer was added to the injection stream. The modeled tracer injection and production are shown in Figure 7.

The path taken by the tracer could be viewed by feeding the simulator with a slug input into only one of the wells at a time. Further inspection of this process revealed that IN01 only feeds into PR01 and IN03 only feeds into PR02. However, IN02 feeds into both PR01 and PR02, and it is also clear that it feeds PR02 through at least two different flow paths. The return curves were also lacking the rarefaction commonly observed in tracer return curves, the reason being that the simulator did not include molecular diffusion effects. An illustration of the tracer mass fraction in the reservoir at around 90 days is shown in Figure 8.

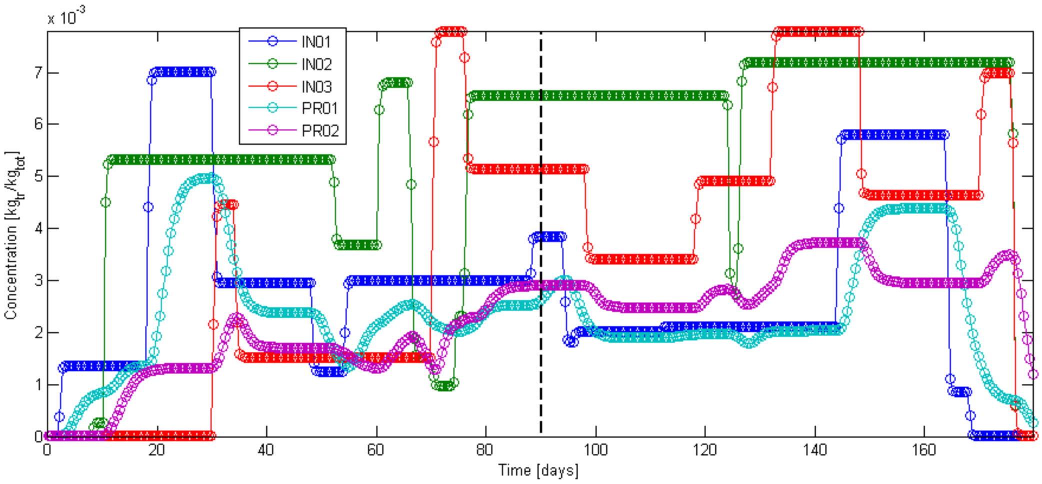

Figure 7: Tracer injection and production history for the DFN simulation scenario. Tracer distribution at day 90 (black dashed line) is shown in Figure 8.

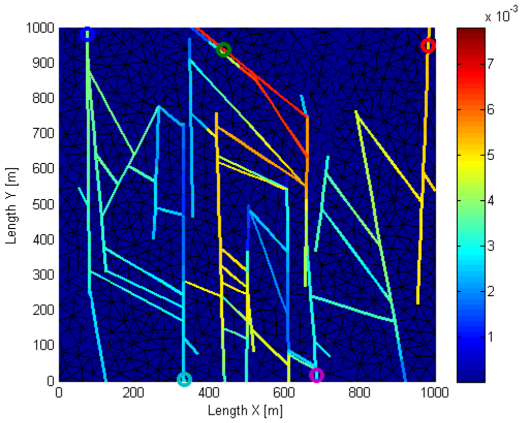

Figure 8: Distribution of tracer concentration at approximately 90 days (corresponds to black dashed line in Figure 7).

3. CHARACTERIZING WELL-TO-WELL CONNECTIONS

As discussed in the introduction, one of the goals of our study was to find a relationship between the input to an injection well and the corresponding response in a production well. While minimal assumptions should be made about the outcome, the deconvolution approach assumes that the relationship is causal (time delay), linear and time invariant (flow is steady state). Moreover, we used an inversion technique that constrains the estimated response to have some degree of smoothness and be nonnegative. The following sections describe the mathematical formulation of the problem and two example applications.

3.1 Formulation of a multiwell deconvolution problem

The method applied here is based on the assumption that well-to-well connectivity can be characterized by the convolution Equation (3). This implies that the production, $c_p(t)$, at one well can be described as a linear function (weighted sum) of previous injection, $c_r(t)$, into another well. The weights, $\kappa(t)$, depend on the time lag between injection and production and form a curve referred to as the kernel. This curve is analogous to a tracer return curve from a slug injection tracer test (within a multiplicative constant, depending on the mass injected).

$$c_p(t) = \int_0^t c_r(t-\tau)\kappa(\tau)d\tau \tag{3}$$

Equation (3) can be set forward in discrete numerical form as:

$$\vec{c}_p = H\vec{\kappa} \tag{4}$$

where H is an n by m matrix, n is the number of conditioning data points and m is the number of discretization points for $\vec{\kappa}$. The kernel, $\vec{\kappa}$, is the unknown to be estimated.

The elements of the H matrix representing the injection will have the formulation described by Equation (5), assuming a Riemann numerical integration scheme with discretization points $\tau_j = \Delta\tau/2 + (j-1)\Delta\tau$, and $j \in \{1,..., m\}$.

$$H_{ij} = \begin{cases} c_r(t_i - \tau_j)\Delta\tau \quad , & t_i > \tau_j - \dfrac{\Delta\tau}{2} \\ 0 \quad , & t_i <= \tau_j - \dfrac{\Delta\tau}{2} \end{cases} \tag{5}$$

The convolution equation can be generalized to handle the case where there are multiple (N_r) injectors, i.e. where the response in the producer is described by

$$c_p(t) = \sum_{k=1}^{N_r} \int_0^t c_{rk}(t-\tau)\kappa_k(\tau)d\tau \tag{6}$$

In this case the discrete form becomes

$$\vec{c}_p = \sum_{k=1}^{N_r} H_k\vec{\kappa}_k = H\vec{\kappa} \tag{7}$$

where now

$$H = \left[H_1 \ H_2 \ldots H_{N_r}\right] \tag{8}$$

and

$$\vec{\kappa} = \left[\vec{\kappa}_1 \ \vec{\kappa}_2 \ldots \vec{\kappa}_{N_r}\right]^T \tag{9}$$

Various solution methods have been proposed for solving the convolution equations (deconvolution). We have experimented with a number of those, in particular a method introduced by Kitanidis (2009) based on Bayesian statistics. Our results for the Bayesian approach are discussed in Juliusson and Horne (2009b). The method introduced here is more in line with the work of Levitan (2007).

The deconvolution problem can be particularly challenging because the *H* matrix has columns that are shifted versions of the previous columns. This generally means that *H* will be singular or very close to singular so unfiltered solution approaches are highly susceptible to noise in the injection and/or production data. Such direct inversions often lead to highly erratic estimates of the kernel function. To counterbalance this, we introduce from physics and observations, the fact that tracer returns will have some degree of continuity (smoothness). This can be modeled by viewing the deconvolution problem as a minimization problem, where the data misfit is minimized in balance with a penalty term for the roughness in the kernel estimate. Hence, the objective function becomes

$$F(\vec{\kappa}) = \underbrace{\tfrac{1}{2}(\vec{c}_p - H\vec{\kappa})^T(\vec{c}_p - H\vec{\kappa})}_{data \quad misfit} + \underbrace{\tfrac{1}{2}\vec{\kappa}^T R\vec{\kappa}}_{roughness \quad penalty} \tag{10}$$

where *R* denotes a roughness penalty matrix, the purpose of which is to introduce some degree of smoothness into the solution, which can be obtained through several different formulations. The approach taken here is to formulate *R* so as to minimize the dth derivative of the kernel function. For example, if smoothness is to be enforced by minimizing the first derivative, the roughness penalty term can be formulated as

$$\frac{1}{2}\vec{\kappa}^T R\vec{\kappa} = \frac{\sigma}{2}\sum_{i=1}^{m-1}(\kappa_{i+1} - \kappa_i)^2 \tag{11}$$

where σ is a scaling parameter that determines the emphasis on smoothness in the optimization. The first derivative approach corresponds to seeking the shortest line through the data. Similarly one can minimize the curvature of the kernel estimate, by using the second derivative, i.e.

$$\frac{1}{2}\vec{\kappa}^T R\vec{\kappa} = \frac{\sigma}{2}\sum_{i=1}^{m-2}(\kappa_{i-1} - 2\kappa_i + \kappa_{i+1})^2 \tag{12}$$

In general this formulation is simply requiring there to be some correlation between consecutive elements of the kernel estimate, $\vec{\kappa}$. The dth derivative based roughness penalty term can be formulated as

$$\frac{1}{2}\vec{\kappa}^T R\vec{\kappa} = \frac{\sigma}{2}\sum_{i=1}^{m-d}\left(\sum_{j=0}^{d}(-1)^j\binom{d}{j}\kappa_{i+j}\right)^2 \tag{13}$$

The general formulation of *R* can be found from Equation (13) by differentiating twice with respect to κ, i.e.

$$R_{pk} = \sigma\sum_{i=1}^{m-d}\left((-1)^{k-i}\binom{d}{k-i}(-1)^{p-i}\binom{d}{p-i}\right) \tag{14}$$

where

$$\binom{d}{j} = 0, \quad if \quad j < 0 \quad or \quad j > d \ . \tag{15}$$

When working with the multiple injector case, *R* will be an $m{*}N_r$-by-$m{*}N_r$ block diagonal matrix, with each *m*-by-*m* matrix block described as in Equation (14). The scaling parameters, σ, can and should be tuned for each matrix block to obtain optimal results. In our experience it has worked well to select the scaling parameters as a multiple of the discretization interval for each kernel estimate, e.g.

$$\vec{\sigma} = s\Delta\vec{\tau} \tag{16}$$

where $\vec{\sigma}$ and $\Delta\vec{\tau}$ are vectors of length N_r. The scaling parameter, *s*, was tuned manually, but that was quite easy since only the approximate order of magnitude had to be found. A formulation of *R* corresponding to the 2nd derivative was generally found to give sufficiently smooth solutions.

The optimization problem specified in Equation (10) was solved in MATLAB using the interior-point algorithm implemented in the function `fmincon`, which is available in the Optimization Tool Box. The solution was constrained to be nonnegative and the initial and final elements of each kernel estimate were set equal to zero. The solution time was reduced significantly by providing the gradient

$$G(\vec{\kappa}) = -H^T(\vec{c}_p - H\vec{\kappa}) + R\vec{\kappa} \tag{17}$$

and the Hessian

$$E(\vec{\kappa}) = H^T H + R \tag{18}$$

for the problem.

Despite all of the constraints, the solution method would often fail when tested with kernel estimates that were discretized over a time scale that spanned a time greater than needed to cover the nonzero part of the kernel. Therefore, additional effort was put into estimating the final nonzero response time, t_f (see Figure 1), for each kernel. In other words, the appropriate timescale for each estimate had to be found.

It turned out that the appropriate timescale could be found by solving a second optimization problem. This time it involved finding the vector $\vec{t}_f$ which denotes the final time for the discretization of each kernel. The function being minimized is still the one described by Equation (10), but we are searching for the discretization interval for $\vec{\kappa}$, that gives the smallest possible *F*. A contour plot of *F* as a function of t_{f1} and t_{f2} is shown in Figure 9. The plot was created for a two injector case, with a random injection pattern, and using the synthetic kernels shown in Figure 10 to create the corresponding production data.

The shape of the contour plot (Figure 9) can be understood in the sense that *F* gets large if t_f is too small, since the kernel will not span a timescale large enough to explain the data misfit. On the other hand, if t_f is too large, there will be fewer discretization points to characterize the actual kernel, which leads to a poorer data fit. It could also be argued that the roughness penalty term will generally increase (and thereby *F*) since variations in the estimate where the kernel should be zero must be attributable to noise. Figure 9 also clearly illustrates that the objective function, in this context, is nonconvex. Therefore, direct search methods were needed to solve this second optimization problem.

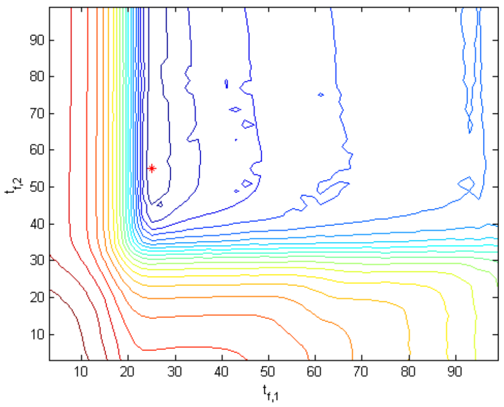

Figure 9: A contour plot of how the log of the objective function varies depending on the assumed end time (t_f) for each kernel estimate. The minimum is shown as a red star. This example was generated using the kernels shown in Figure 10.

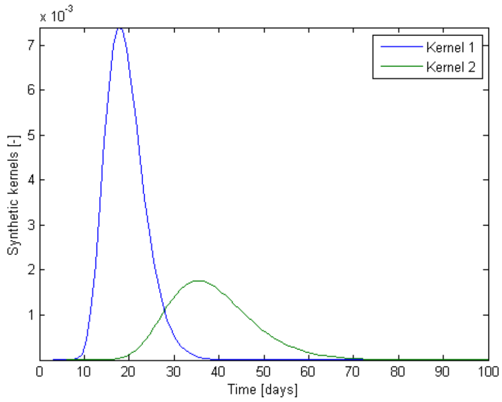

Figure 10: Synthetic kernels used to illustrate the dependence of the objective function (Figure 10) on the final time used in the inversion for each kernel.

A direct search strategy that we found quite successful in determining the timescale was to first apply a Genetic Algorithm to find the approximate location of the minimum, and then follow up with a Pattern Search algorithm. Both of these searches were implemented with functions from of the Optimization Tool Box in MATLAB.

3.2 Deconvolution using DFN Simulation Data

The first application example that will be discussed is the deconvolution of the synthetic data generated from the DFN simulation discussed in Section 2.2. Some Gaussian noise was added to the first 90 days of the data and then that part was used to estimate the kernels.

The kernel estimates are shown in Figures 11 and 12. As can be seen, injectors IN01 and IN02 have a well-characterized connection to producer PR01, while IN03 contributes very little to the signal in PR01. Note that the timescale estimate for IN03 is quite arbitrary since the kernel values are all close to zero and therefore have little effect on the objective function. Similar observations can be made for the kernels relating to PR02, where IN03 is well connected and IN01 has very limited connection. A range of other conclusions can be drawn from the shape of the return curves, about the fraction of tracer produced in each well, the travel time, level of dispersivity in the reservoir etc.

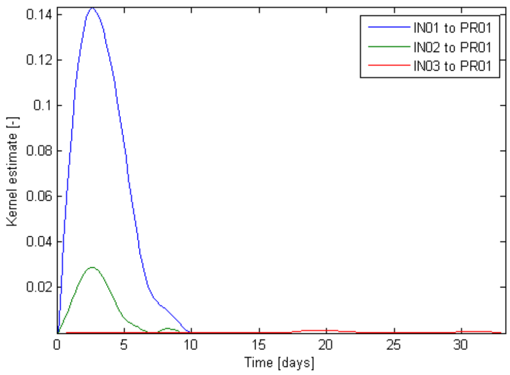

Figure 11: Kernel estimates for connections to producer PR01.

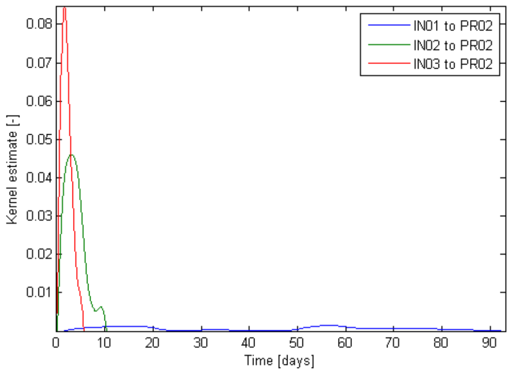

Figure 12: Kernel estimates for connections to producer PR02. Note that the time scale is different from that in Figure 11.

The kernel estimates were used to predict the future response in the production wells, using Equation (4). The resulting predictions were quite accurate, as illustrated in Figure 13. This serves as a type of validation test for our results.

3.3 Deconvolution using Data from an Analytical Equation

As a second example, synthetic production data were generated for a case with five injectors and one producer. The data were generated using an analytical solution of the one dimensional advection-dispersion equation, specifically the finite impulse response on an infinite domain, i.e.

$$c_p(t) = \frac{\bar{f}M}{\sqrt{4\pi Dt}} \exp\left[-\frac{(x-ut)^2}{4Dt}\right] \qquad (19)$$

where c_p is the produced concentration [kg/m], $\bar{f}$ is the fraction of tracer retrieved in the producer, M is the injected mass [kg], D is the dispersivity coefficient [m^2/s], x is the flow distance between the wells [m], and u is the mean flow velocity [m/s].

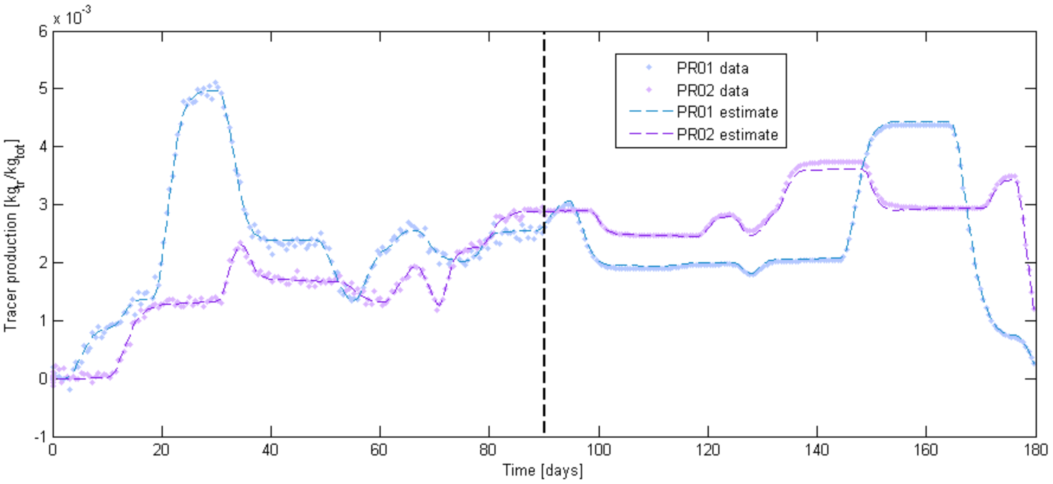

Figure 13: The data for the first 90 days was used to estimate the kernels and the response for the next 90 days was quite successfully predicted, as shown here.

The relationship between mass and concentration can be written as

$$M = c_r(t)udt \tag{20}$$

where c_r is the concentration of injected fluid and dt is the duration of the injection pulse. This can be substituted into (**Error! Reference source not found.**) to obtain any given response by convolution

$$c_p(t) = \int_0^t \frac{\bar{f}c_r(t-\tau)u}{\sqrt{4\pi D\tau}} \exp\left[-\frac{(x-u\tau)^2}{4D\tau}\right] d\tau \tag{21}$$

So in this case the kernel has the formulation

$$\kappa(t) = \frac{\bar{f}u}{\sqrt{4\pi Dt}} \exp\left[-\frac{(x-ut)^2}{4Dt}\right] \tag{22}$$

Note that the kernel depends only on the characteristics of the well-to-well connection, and not the mass or concentration of tracer injected. The kernel is however dependent on the amount of injected fluid, through the average flow velocity, u. This means that meaningful kernel estimates require that the fluid injection and production rates stay constant (or close to constant), but the amount of injected tracer can and should vary.

Taking this analytical approach has several advantages for validating our method, for example we can (easily) generate a large variety of kernels, and the true shape of the kernels will be known a priori.

The deconvolution method was subject to a slightly harder test this time as the number of injectors was increased to five. Each of the five injectors had an arbitrary connection to the producer as illustrated in Figure 14.

Figure 15 shows the kernels associated with each injector-producer connection. Importantly, these illustrate a fair amount of variety in dispersivity, retrieval fraction, flow velocity and distance.

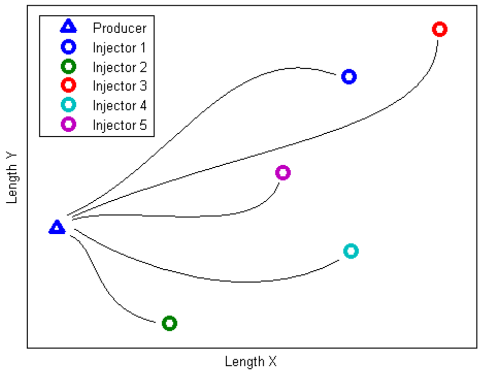

Figure 14: Configuration with one producer and five injectors. The connecting path to the producer is unknown.

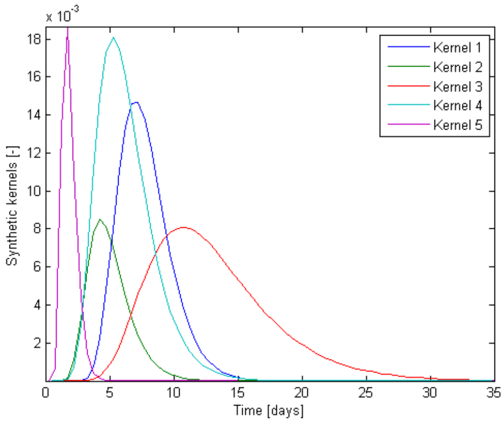

Figure 15: Five synthetic kernels used in this example

Each kernel in our estimate was discretized into 50 elements. This means that the total number of parameters being estimated was 250. The production data generated in this case contained 400 measurements. Therefore, it was very important to have as little redundancy as possible in the convolution equations, i.e. the output signal had to contain as much information as possible about the response of the producer to changes at each injector. To accomplish this, an approach, similar to that presented by Lee et al.

(2008, 2009), was taken, which involved creating a series of Haar wavelet based injection signals. This way a wide combination of injection signals was tested and all injectors received the same total amount of tracer, but in varying concentration over varying time spans. Figure 16 illustrates the tracer injection pattern used, and the corresponding output signal.

The deconvolution method described in Section 3.1 was used to solve this five injector case. As Figure 17 shows, a solution was found that reproduced the data very well. Moreover, all of the kernels, which were known in this case, were reproduced quite accurately. This is illustrated in Figure 18. Note that the inversion worked well even though the timescale (time beyond which the kernel essentially goes to zero) was a bit off in a few cases, e.g. for kernels 3 and 5.

Admittedly, the inversion was less robust when some of the kernels were extremely dissimilar, especially for kernels that were highly dispersive or if the fraction of tracer retrieved was very small, since the contribution of these to the signal would be negligible or lost in noise. On the other hand, the high uncertainty associated with those kernel estimates could possibly be quantified and linked to low connectivity between wells.

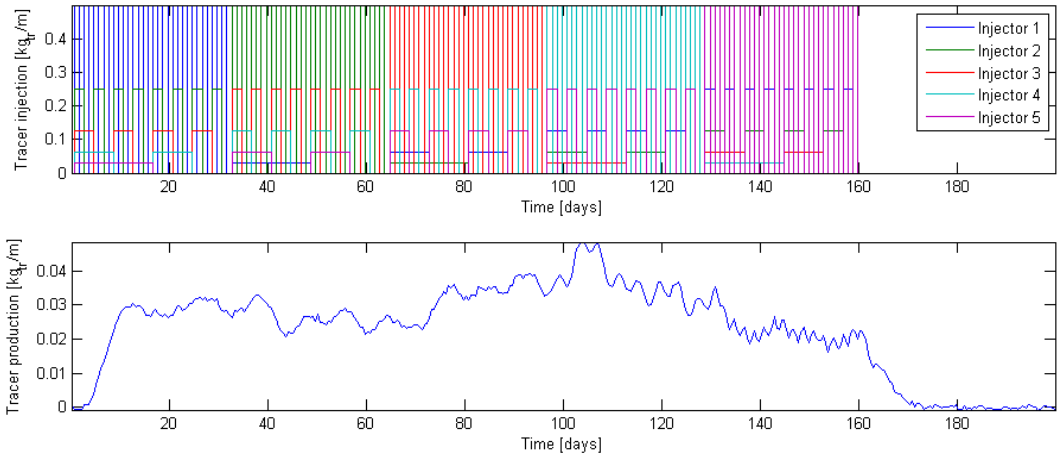

Figure 16: Controlled injection and corresponding production for a case with five injectors and one producer.

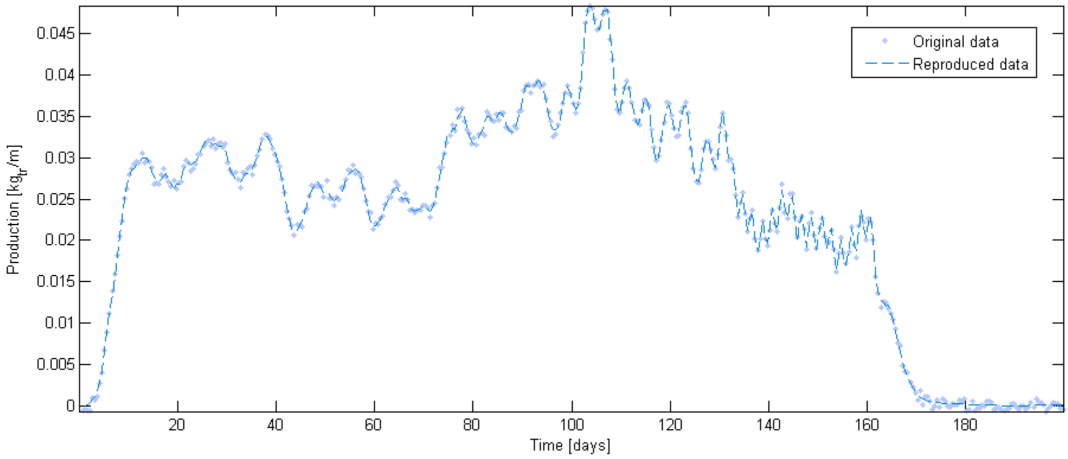

Figure 17: Data reproduction after solving the five injector case.

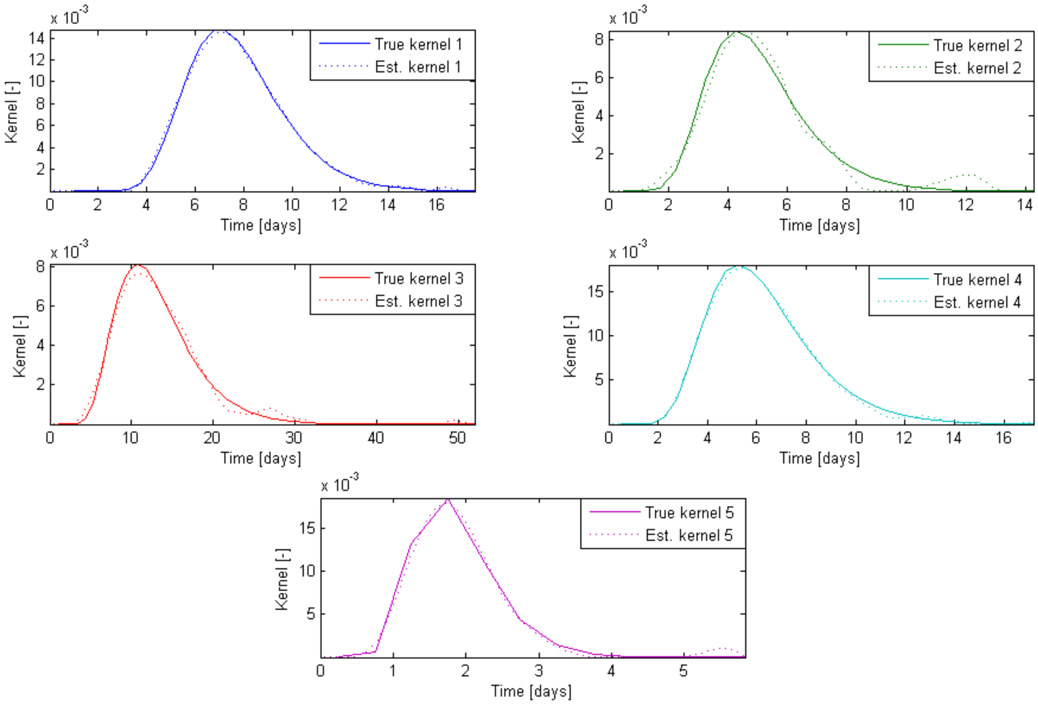

Figure 18: Estimated kernels compared to actual kernels used.

4. DISCUSSION

This section discusses some additional observations, caveats and ideas for where this research direction might lead.

First, an important observation should be pointed out regarding the limitations of tracer tests. The DFN simulation presented in Sections 2.2 and 3.2 (and this was verified by further simulation) showed that well IN01 sent tracer only to well PR01 and none to PR02. Similarly, tracer from IN03 was only produced in PR02. The kernel estimates also revealed this information. Note, however, that the *tracer returns do not say anything about which wells are not connected*. For example, it is clear from the DFN model that there is a fracture connection between IN03 and PR01, but that is not seen in the tracer returns because of the specific operating conditions in our model.

It should also be reiterated here, that the reason the deconvolution method works so well in this case is that the model conditions set up in this case involved steady state pressure and temperature. The only thing that was varied was tracer concentration, which at these conditions gives a linear response. These conditions are somewhat idealized, but conditions that are very close will surely occur, especially in EGS and other liquid dominated systems.

This deconvolution method should be extendable to other data types, e.g. pressure, temperature and enthalpy, and those will be considered in future research. The information conveyed by the kernels should also be useful for characterizing the fracture network that created them.

A real data set that would help illustrate the usefulness of this approach has not been found, and it seems unlikely that a data set with sufficient temporal resolution exists. Therefore, the take-home message is that collecting chemical samples more regularly (daily) may be worth the effort. Moreover, a well controlled injection schedule will make the method more likely to succeed.

5. CONCLUSIONS

This paper discusses motivation and methods for characterization of fractures in geothermal reservoirs. Some previous work leading to this research is briefly reviewed.

A two-dimensional reservoir model with a discrete fracture network was built and used for numerical simulations with the TOUGH2 code. A case study of tracer injection, with three injectors and two producers, was presented. An important lesson learned from the simulations was that tracer returns can give information about which wells are connected, but they can not be used to make conclusive remarks about which wells are *not* connected through a fracture network. Finally, some of the DFN simulation capabilities and limitations were discussed. For example, heat and solute transfer effects between the fracture and the matrix need to be captured more effectively.

A formulation of the deconvolution problem for tracer returns was presented. The formulation included the multiple-injector – single-producer case. The method was then used to solve two examples, based on synthetic tracer data. The results indicate that this deconvolution method can successfully reveal the underlying transfer functions (kernels) for tracer transport between wells. The kernels are analogous to tracer return curves and can therefore be used to make any predictions made from conventional tracer tests. The kernels can also be used to predict the response to future variations in the injection of solute. It is important, however, that the flow field remains at steady state (roughly) throughout the data collection and prediction period.

5. ACKNOWLEDGEMENTS

This research was conducted with financial support to the Stanford Geothermal Program from the US Department of Energy under grant DE-FG36-08GO18192, the contribution of which is gratefully acknowledged.

Motivation and support from Landsvirkjun Power is also gratefully acknowledged.

REFERENCES

Axelsson, G., Flovenz, O.G., Hauksdottir, S., Hjartarson, A. and Liu, J.: Analysis of tracer test data, and injection-induced cooling, in the Laugaland geothermal field, N-Iceland, *Geothermics*, **30**, (2001), 697-725.

Horne, R.N., and Szucs, P.: Inferring Well-to-Well Connectivity Using Nonparametric Regression on Well Histories, *Proceedings*, 32nd Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, CA (2007).

Juliusson E. and Horne, R.N.: Fracture characterization using production and injection data, DOE quarterly report (2009 January to March), Contract DE-FG36-08GO18192, Stanford Geothermal Program, Stanford University, California, (2009a), 1-17.

Juliusson E. and Horne, R.N.: Fracture characterization using production and injection data, DOE quarterly report (2009 April to June), Contract DE-FG36-08GO18192, Stanford Geothermal Program, Stanford University, California, (2009b), 1-24.

Lee, K.H., Ortega, A., Nejad, A.M., and Ershaghi, I.: A Method for Characterization of Flow Units between Injection-Production Wells Using Performance Data, SPE 114222, SPE Western Regional and Pacific Section AAPG Joint Meeting, Bakersfield, CA (2008).

Lee, K.H., Ortega, A., Nejad, A.M., Jafroodi, N., and Ershaghi, I.: A Novel Method for Mapping Fractures and High Permeability Channels in Waterfloods Using Injection and Production Rates, SPE 121353, SPE Western Regional Meeting, San Jose, CA (2009).

Levitan, M.M.: Deconvolution of Multiwell Test Data, SPE Annual Technical Conference and Exhibition, San Antonio, TX (2006).

Lovekin, J. and Horne, R.N.: Optimization of injection scheduling in geothermal fields, MS thesis, Stanford University, Stanford, CA (1989).

Karimi-Fard, M., Durlofsky, L.J. and Aziz, K.: An efficient discrete fracture model applicable for general purpose reservoir simulators, SPE 79699, SPE Reservoir Simulation Symposium, Houston, TX (2003).

Kitanidis, P.K.: Applied Stochastic Inverse Problems, unpublished textbook for class CEE362G, Stanford University, Stanford, CA (2009).

Macario, M.E.: Optimizing Reinjection Strategy in Palinpinon, Philippines Based on Chloride Data, MS thesis, Stanford University, Stanford, CA (1991).

McClure, M.: Fracture Stimulation in Enhanced Geothermal Systems, MS thesis, Stanford University, Stanford, CA (2009).

Shewchuk J.R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, *Applied Computational Geometry: Towards Geometric Engineering*, **1148**, Springer-Verlag, Berlin, (1996), 203-222.

Sullera, M.M., and Horne, R.N.: Inferring Injection Returns from Chloride Monitoring Data, *Geothermics*, **30**, (2001), 591-616.

Williams, C.F., Reed, M.J., and Mariner, R.H.: A Review of Methods Applied by the US Geological Survey in the Assessment of Identified Geothermal Resources., US Geological Survey, Menlo Park, CA (2008).