GEOTHERMAL MODEL CALIBRATION USING A GLOBAL MINIMIZATION ALGORITHM BASED ON FINDING SADDLE POINTS AND MINIMA OF THE OBJECTIVE FUNCTION

Manuel Plasencia[a], Andreas Pedersen[a], Andri Arnaldsson[b], Jean-Claude Berthet[b] and Hannes Jónsson[a,c]

[a]Science Institute of the University of Iceland, VR-III, 107 Reykjavík , Iceland

[b]Vatnaskil Consulting Engineers, Sudurlandsbraut 50, 108 Reykjavík , Iceland

[c]Faculty of Physical Sciences, University of Iceland VR-III, 107 Reykjavík , Iceland

e-mail: mpg2@hi.is

DOI: 10.1016/j.cageo.2013.09.007

Abstract

The objective function used when determining parameters in models for multiphase flow in porous media can have multiple local minima. The challenge is then to find the global minimum and also to determine the uniqueness of the optimized parameter values. A method for mapping out local minima to search for the global minimum by traversing regions of first order saddle points on the objective function surface is presented. This approach has been implemented with the iTOUGH2 software for estimation of models parameters. The methods applicability is illustrated here with two examples: a test problem mimicking a steady-state Darcy experiment and a simplified model of the Laugarnes geothermal area in Reykjavík, Iceland. A brief comparison with other global optimization techniques, in particular simulated annealing, differential evolution and harmony search algorithms is presented.

Keywords: global optimization, inverse modeling, reservoir modeling.

1. Introduction

The development of reservoir models often involves inverse modeling, i.e. an estimation of model parameters by fitting calculated values of the response of the system to measurements at discrete points in space and time. The difference between the model calculation and the measured data at the calibration points can be represented by an objective function of the model parameters. The task of estimating the best set of model parameters is thereby formulated as an optimization problem where the goal is to determine the parameter values that minimize the objective function. Even for models with only a few parameters, the resulting objective function can have more than one minimum. This is illustrated in Fig. 1, which shows a one-dimensional cut of an objective function for a geothermal reservoir model described below. Within the parameter interval shown, three local minima are present. The occurrence of multiple local minima is more likely in models with a larger number of parameters. Hence, the task becomes to find the global minimum of the objective function among several local minima. This is a challenging problem. Furthermore, it is important to know whether additional local minima, with only insignificantly higher objective function values are present since they could, for practical purposes, represent nearly as good parameter values as the global minimum.

Numerical algorithms for optimization can be broadly categorized into local optimization methods and global optimization methods. Local optimization algorithms involve an iterative process where starting from some initial guess, new parameter values are found so as to lower the value of the objective function. Such algorithms only find local minima, typically the local minimum nearest to the initial guess. Typically, local optimization methods rely on the evaluation of the gradient of the objective function. Some examples are steepest descent, conjugate gradient, Quasi-Newton and Levenberg-Marquardt methods. By carrying out multiple minimizations starting from different initial guesses, such methods can be used to find the global minimum but this becomes an inefficient procedure when many parameters are varied.

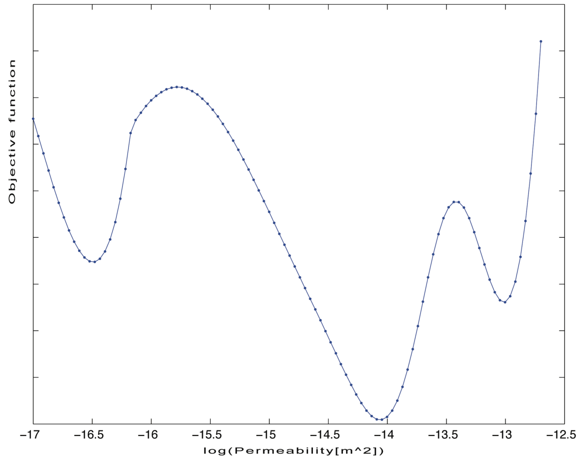

Figure 1. *A one-dimensional cut of the objective function for a geothermal reservoir model of the Laugarnes area described in section 4. The logarithm of the permeability is varied. In addition to a global minimum (near -14.1), two local minima are present (near -16.5 and -13.0).*

Global optimization algorithms, on the other hand, attempt to find the global minimum by also allowing the increases of the objective function during the iterative procedure. Some examples are, simulated annealing using Markov chain Monte Carlo methods and evolutionary algorithms such as differential evolutionary, harmony search, and particle swarm optimization. These methods do not make use of the gradient of the objective function and tend to converge more slowly to minima of the objective function, but have the advantage over local optimization methods that they can identify the global minimum. Three of these algorithms will be briefly described here, the simulated annealing, differential evolution and harmony search algorithms. These are implemented in the iTOUGH2 software, and will be compared with the global optimization method proposed here.

Simulated Annealing (Kirkpatrick et al., 1983) is an iterative procedure where an initial guess of the parameter values is iteratively updated with random increments and a selection criterion until a termination condition is reached. There, the objective function is taken to represent an 'energy' of the system, and a fictitious temperature is introduced. The temperature is introduced to control the probability of accepting increases in the objective function as an intermediate step to ultimately reach lower function values. A central issue in simulated annealing calculations is the ‘time scale’ of the cooling of the system from high temperature to zero temperature. The slower the cooling rate, the more likely the global minimum is found, but the computational effort becomes larger. It has been shown that in the impossible limit of infinitely long simulations with infinitesimal cooling rate, the method is guaranteed to give the global minimum (Haario and Saksman, 1991; Tsallis and Stariolo, 1996). For a given amount of computational effort, an implementation that can simulate a longer time interval is, more likely to reach the global minimum.

The Differential Evolution algorithm (Storn and Price 1996) uses a randomly generated initial population, preferably covering the entire parameter space, which is then modified by differential mutation and crossover along with a selection criterion to find the minimum of the objective function. It has emerged as one of the simplest and most efficient techniques for solving global optimization problems. The method has been applied to diverse domains of science and engineering, such as mechanical engineering (Joshi and Sanderson 1999), chemical engineering (Wang and Jang 2000), machine intelligence, and pattern recognition (Das et al. 2008). Some weaknesses of the method have been identified (Lampinen and Zelinka 2000). Furthermore, the performance of the method deteriorates as the number of parameters increases (Ali et al. 2012). Several suggestions for improving its performance have been proposed (Ali and Pant 2011).

Harmony search (Geem et al., 2001) is also a population-based optimization algorithm using a stochastic random search (Lee and Geem, 2004). It has been applied to a wide variety of optimization problems (Geem et al., 2002, 2005; Kang and Geem, 2004; Kim et al., 2001; Lee and Geem, 2004). However, problems with the method, such as the need for parameter tuning, have been a topic of much research over the last 10 years where improvements have been proposed (Fourie at al., 2013).

The global optimization method presented here can be considered as a descendant of a method for long time scale simulations of atomic scale models of solids, known as adaptive kinetic Monte Carlo (AKMC) (Henkelman and Jónsson, 2001). The AKMC method has been successfully applied to atomic scale problems in solid-state physics and chemistry, see for example: (Henkelman and Jónsson, 2003; Karssemeijer et al., 2012; Pedersen et al., 2009a, 2009b; Pedersen and Jónsson 2010). There, the time evolution is described by visiting local minima on the energy surface and identifying transitions by searching for first order saddle points on the objective function surface (Henkelman and Jónsson, 1999). Here, we modify the AKMC method to adapt it better to global optimization (Pedersen et al., 2012). The method, which we will refer to as global optimization using saddle traversals (GOUST) is described in detail below. It has been implemented in the EON software (Pedersen and Jónsson 2010), which makes it possible to carry out the calculations using distributed and cloud computing

2. GOUST method

The GOUST method relies on a fast way to identify first order saddle points on the objective function surface. We therefore first describe briefly the tool used for this purpose. A more detailed description is given in Henkelman and Jónsson (1999).

2.1. Minimum mode following

Let the number of variables of the objective function (parameters in the model to be fitted) be denoted by N. The objective function can be denoted as

$$f : \mathbb{R}^N \rightarrow \mathbb{R} \qquad (1)$$

this defines a surface in N-dimensional space. The function is assumed to be differentiable. The extremal points where the gradient vanishes, $\nabla f = 0$, and the function value is low are of particular interest as these are local minima and low lying saddle points. To distinguish between these two kinds of extrema, the matrix of second order derivatives (the Hessian matrix, Eq. (2)) can be used. The Hessian has only positive eigenvalues at a local minimum, whereas one of the eigenvalues is negative at a first order saddle point (SPs).

$$H_{ij} = \frac{\delta^2 f}{\delta x_i \delta x_j} \qquad (2)$$

To locate SPs, it is assumed that the gradient ∇f of the objective function can be evaluated readily (recent developments in automatic differentiation see (Gregory et al., 1997)) could prove valuable in this context), but second derivatives are not needed. The method used to find SPs involves a minimization using a transformed gradient where the component along the minimum mode of the Hessian has been reversed

$$\nabla f^{eff} = \nabla f - 2(\nabla f \cdot \hat{v}_\lambda)\hat{v}_\lambda \qquad (3)$$

here, $\hat{v}_\lambda$ is a normalized eigenvector corresponding to the minimum eigenvalue, λ, of the Hessian. This projection (Eq. (3)) locally transforms the gradient in the vicinity of a SP to a gradient characteristic of the vicinity of a minimum. A number of local minimization methods can then be used to converge on SPs when $\nabla f^{eff}$is used as input, for example the conjugate gradient method. This will be referred to as the minimum mode following (MMF) method. It is important to note that only the minimum mode of the Hessian matrix is required here. The minimum mode vector can be estimated efficiently using either the dimer method (Henkelman and Jónsson, 1999) or the Lanczos method (Lanczos, 1950) where only first order derivatives are required. The full Hessian does not need to be computed.

Once a SP search has exited the region around a local minimum where all eigenvalues are positive, a steepest descent search path is stable and deterministic. That is, a MMF search starting from a point outside the positive region will converge on a particular SP.

2.2. The algorithm

The GOUST algorithm is based on the principle of finding new local minima on the objective function surface by climbing up from known minima so as to identify SPs, and then sliding down from there using a local minimization algorithm. For a given local minimum, several SP searches are carried out starting from slightly perturbed values of the variables. For each of the perturbed parameter values, the MMF method is used to climb up the surface and converge onto a SP. After a SP has been located, the adjacent local minima are found. This is done by displacing the system along the minimum mode in both directions from the SP, followed by a local minimization to slide down to the minima. The method is presented in greater detail below.

GOUST algorithm:

1. From the parameter values, x_0, provided by the user as an initial starting point, a local minimization is carried out to

bring the system to a local minimum. A list of minima is initialized by storing this local minimum, x_{m0}, and the corresponding value of the objective function, $f(x_{m0})$.

2. n new parameter values, $x^r_1, x^r_2, \ldots x^r_n$, are generated as starting points for saddle point searches by applying small random displacements, sampled from a Gaussian distribution, to the parameter values at the minimum. The displacements are small and all parameter sets $x^r_1, x^r_2, \ldots x^r_n$ are in the vicinity of the minimum. (The superscript "r" stands for random).
3. From each parameter set, a saddle point search using the MMF method is conducted.
4. Step 2 and step 3 are repeated until a predetermined number of unsuccessful attempts to locate new saddle points has been exceeded.
5. This minimum is marked as visited.
6. All saddle points found are stored in the list of saddle points, $x_{sp1}, x_{sp2}, \ldots x_{spn}$ and $f(x_{sp1}), f(x_{sp2}), \ldots f(x_{spn})$. Each saddle point is listed only once.
7. The saddle point (which has not been sampled) with the lowest value of the objective function is selected, and a displacement along the minimum mode vector made in both directions from the saddle point followed by local minimization to determine the two adjacent minima, x_{m1} and x_{m2}.
8. The minima, x_{m1}, x_{m2} and $f(x_{m1}), f(x_{m2})$, are stored in the list of minima (each minimum only listed once).
9. This saddle point is marked as sampled.
10. The lowest minimum in the list of minima that has not been visited is selected and steps 2 through 9 repeated until (a) all saddle points have been sampled, (b) all minima in the list have been visited, and (c) any other stopping criterion is fulfilled such as CPU time, maximum number of iterations, etc.

Finally, the sorted list of minima is reported as output, the lowest one being the best estimate of the global minimum and the other low lying minima giving an indication of how unique the optimal values of the parameters are. The algorithm, thereby, generates a map of the relevant minima. One advantage of the algorithm can be that only relatively low values of the objective function need to be evaluated and the regions with high values, which can be less well defined, are avoided. An example of this is discussed below.

The GOUST algorithm can be applied in many different contexts. Here, we report our work on interfacing the algorithm with iTOUGH2 for inverse modeling. The goal is to demonstrate its applicability to parameterization of geothermal reservoir models. The algorithm is illustrated below in the context of two application problems.

3. Model 1: steady-state Darcy experiment

We illustrate the occurrence of multiple minima and the performance of the optimization algorithm proposed here with a test problem presented in the iTOUGH2 documentation (Finsterle, 2007). Water is injected at constant pressure into a one-dimensional, horizontal column filled with uniform, partially saturated sand. This setup is similar to the steady-state Darcy experiment (Finsterle, 2007). However, there is a certain amount of free gas initially present in the column. Information about the transient behavior of pressure and flow rate is used to determine two-phase flow parameters. The model parameters to be optimized are the permeability of the sand and the initial gas saturation.

3.1. Objective function

The objective function (see Eq. 4) is defined as the squared deviation between measured and calculated pressure and flow rate at two selected points within the column.

$$S(p_1,p_2)=\sum_{i=1}^{60}\frac{\left[P_i^*-P_i(p_1,p_2)\right]^2}{\sigma_{P_i^*}^2}+\sum_{i=1}^{60}\frac{\left[Q_i^*-Q_i(p_1,p_2)\right]^2}{\sigma_{Q_i^*}^2} \qquad (4)$$

here, p_1 and p_2 are the two variable parameters: logarithm of permeability and initial gas saturation. P and Q stands for pressure and flow rate respectively, the superscript '*' denotes measured data and σ^2 is the variance.

Fig. 2 shows the shape of the objective function as the two parameters are varied. This was obtained by mapping out the function on a two-dimensional regular grid using the grid search option implemented in iTOUGH2. It is clear that the objective function has several minima. The global minimum occurs for log (permeability) of -11.7 and initial gas saturation of 0.30. One local minimum occurs in the vicinity of (-13.1, 0.32), and another minimum is located beyond the boundary of the parameter space, in the vicinity of (-11.6, 0). Even though these local minima are shallow compared to the global minimum and one of them is located outside the predefined parameter space, they still signify a problem in that local minimization algorithms can converge onto these values, as will be shown in the next section. Models involving more parameters are likely to have more local minima, several of which can correspond to reasonable parameter values. We use simple two-parameter examples here to illustrate the problem, but the goal is to be able to deal with multiple local minima in realistic problems involving many parameters.

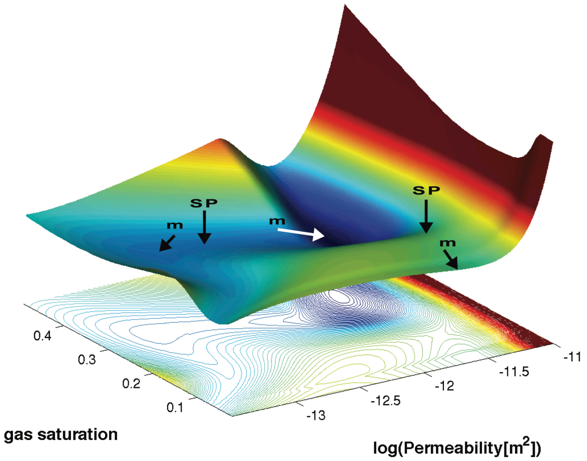

Figure 2. *Objective function surface for the steady-state Darcy experiment test problem showing the effect of varying two parameters: logarithm of permeability [log (k)] and initial gas saturation. The arrows point to the position of minima (m) and saddle points (SP). In addition to the global minimum (white arrow), one shallow local minimum is within the allowed parameter range and one is just outside that area.*

For each local minimum, there exists a range of parameter values such that a steepest descent minimization converges onto this local minimum. The parameter space can be divided up into such basins of attraction corresponding to each of the local minima. The boundaries of the basins of attraction correspond to ridges on the objective function surface. Minima along these ridges correspond to SPs on the objective function surface.

3.2. Finding the nearest local minimum

The Levenberg-Marquardt minimization algorithm is found to perform well for most iTOUGH2 applications (Finsterle, 2007) and can be made to converge efficiently by selecting appropriate values for the Levenberg parameter and Marquardt parameter. However, as for most other local minimization algorithms, it only strives to converge to the minimum closest to the initial guess of the model parameters. This is illustrated in Fig. 3 showing minimization paths obtained using the iTOUGH2 software and the Levenberg-Marquardt minimization algorithm starting from two different initial guesses for the parameter values. Minimization paths obtained using the Gauss-Newton minimization algorithm and the conjugate gradient method from the same initial guesses are also shown for comparison.

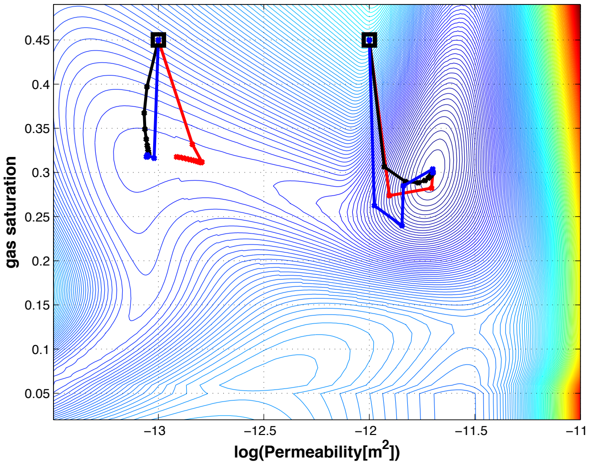

Figure 3.: *Minimization paths using Levenberg-Marquardt algorithm (black lines), Gauss-Newton algorithm (red lines) and conjugate gradient algorithm (blue lines) for the steady-state Darcy experiment test problem. Calculations are started from different initial guesses (squares) for the two parameters: log (Permeability) and initial gas saturation. This illustrates the possibility that a minimization using these methods can lead to convergence to a local minimum with a substantially higher value of the objective function than the global minimum.*

The above example illustrates the need for exploring the objective function surface beyond local minima nearest to the initial guess. While it is easy to set up enough minimization calculations to cover a fine grid of possible initial parameter values when the number of parameters is small, this will quickly become unmanageable as the number of parameters increases.

3.3. Comparison of various optimization methods

In this section, four of the global optimization algorithms mentioned above are illustrated by application to the two-parameter Darcy experiment test problem using the iTOUGH2 software. Since this application only involves two parameters, making it easier to visualize the results, it does not represent a proper benchmark problem where a performance, such as computational efficiency and robustness can be tested. However, the different approaches can be illustrated with this example. Figs. 4 - 7 show results of simulated annealing, differential evolution, harmony search and the proposed GOUST algorithms. In all cases, the calculation is started from the same initial parameter values, (-13.0, 0.30), from which a local minimization would lead to convergence to one of the local minima rather than the global minimum.

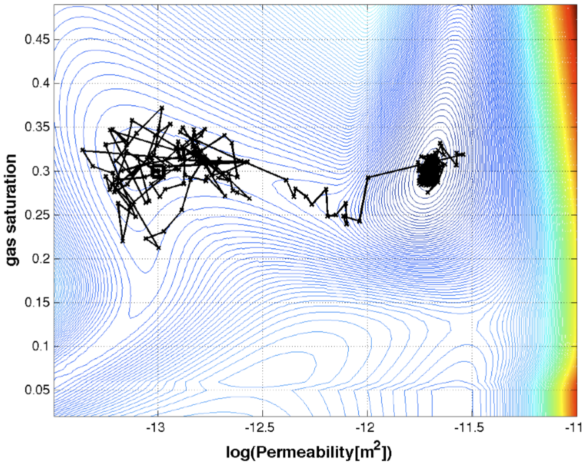

Figure 4: *A simulated annealing minimization for the steady-state Darcy experiment test problem with two parameters: log (Permeability) and initial gas saturation. The initial guess for the parameter values was (-13.0, 0.3). Each evaluation of the objective function is shown with a 'x'. The calculation does reveal the global minimum but requires a large number of objective function evaluations, a total of 3404.*

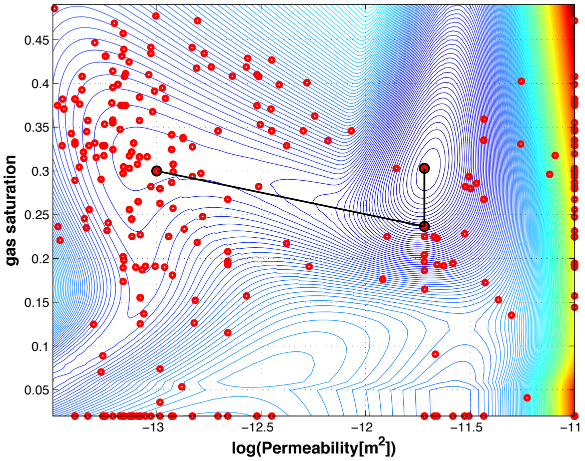

Figure 5: *A differential evolution minimization for the steady-state Darcy experiment test problem with two parameters: log (Permeability) and initial gas saturation. The initial guess for the parameter values was (-13.0, 0.3). Each evaluation of the objective function is shown with a red circle. The calculation reveals the global minimum and requires 778 objective function evaluations.*

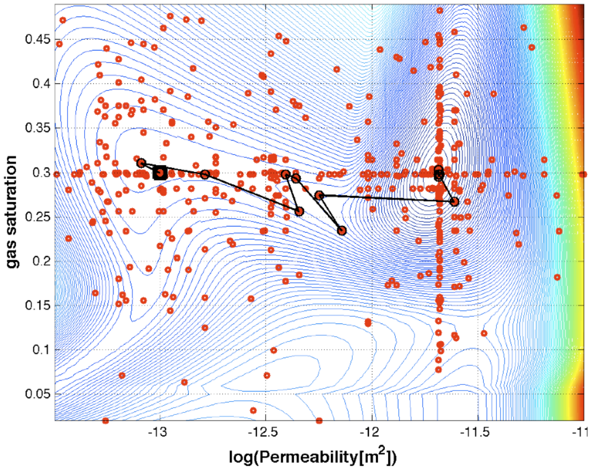

Figure 6: *A harmony search minimization for the steady-state Darcy experiment test problem with two parameters: log (Permeability) and initial gas saturation. The initial guess for the parameter values was (-13.0, 0.3). Each evaluation of the objective function is shown with a red circle. The calculation reveals the global minimum and requires 569 objective function evaluations.*

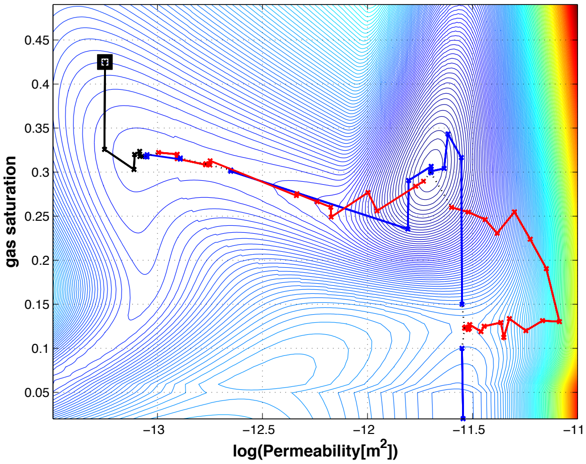

Figure 7: *A GOUST exploration of the objective function for the steady-state Darcy experiment test problem by traversing from one local minimum to another via first order saddle points. Given an initial guess (square), a minimization is carried out (solid black line) converging to a local minimum (m1). Then, a random increment of the parameter values from the minimum is made (dashed black lines) and the minimum mode following method used to climb up (red lines) the objective function surface to home in on first order saddle points (sp1). Then an increment of parameter values along the mode with negative curvature is made and minimization carried out (blue lines) leading to a new minimum (m2). The process is repeated at m2 and as a result both the global minimum and the local minima were found.*

Because the position of the global minimum is known, we can assess the accuracy of the result reported by the algorithms by calculating how large the deviation is from the global minimum. All the calculations successfully find the global minimum within a certain margin. The computational effort will be reported in terms of the number of objective function evaluations (OFE) needed, since they are by far the dominant part of the calculations. An evaluation of the objective function requires running the TOUGH2 software to evaluate the forward model. The number of OFE for these global optimization methods is, of course, significantly larger than for the local minimization calculations shown in Fig. 3.

The simulated annealing algorithm was allowed run for 200 iterations and the simulation was repeated 12 times. The global minimum was always identified and the average number of OFE was 3404. This turned out to be the most computationally demanding method. The progress of one of the calculations is shown in Fig. 4.

The differential evolution algorithm was first allowed to run 30 iterations with a predefined population size of 20 individuals. For these conditions it converged only approximately to the global minimum, requiring 623 OFE. In an attempt to increase the accuracy, the number of iterations was increased to 35, 40, 45 and 50 while keeping the same population size of 20 individuals. The number of OFE then increased to 723, 823, 923 and 1023, respectively. In all cases, the end result was the same except for the run with 45 iterations where the global minimum was identified more precisely. Overall the results do not seem to depend strongly on the number of iterations. The effect of increasing the size of the population was then tested by increasing it to 25, 30, 35, 40 and 45 while number of iterations was kept at 30. The number of OFE was then equal to 778, 933, 1088, 1243 and 1398, respectively. The size of the population had a stronger effect on how closely the simulation got to the global minimum. Overall the best choice in terms of accuracy turned out to be 30 iterations and a population size of 25 individuals. Then the number of OFE was 778. The progress of that calculation is shown in Fig. 5.

In calculations using the harmony search algorithm, the number of iterations was set to 200 but the following values of the 'harmony memory' were tested: 15, 20, 25, 30 and 35. The calculations required 544, 580, 569, 459 and 481 OFE, respectively. The global minimum was most accurately identified for harmony memory of 25. The progress of that calculation is shown in Fig. 6.

In the case of the GOUST algorithm, an evaluation of the objective function requires running iTOUGH2, which then runs the TOUGH2 software to evaluate the objective function and its gradient. The GOUST algorithm was evaluated in three cases configured to run the same number of iterations, same stopping and convergence criteria for saddle points and minima searches but different number of SP searches at each minimum. The simulation was repeated 12 times. In all cases the global minimum was accurately identified. For one, two and three SP searches per minimum the average number of iTOUGH2 evaluations of the objective function and gradient was 212, 264 and 358, respectively. Since the gradient is evaluated here by finite differences, the number of OFE is three times larger. An implementation making use of automatic differentiation to obtain the gradient without finite differences could be more efficient. The increase in the number of SP searches per minimum does not affect how close the results are to the global minimum because the convergence criterion is based on a tolerance in the gradient. The progress of one of the calculations is shown in Fig. 7. From the initial guess of the two parameter values, a local minimization is first conducted using the conjugate gradient minimization method. This reveals the local minimum at (-13.06, 0.32). From there, a small random displacement was generated and a SP search conducted using the MMF method. A SP was located at (-12.8, 0.31). To continue, a small displacement was made along the minimum mode and a new local minimization conducted to converge onto a new minimum located at (-11.7, 0.30). The same process was repeated for the newly found minimum. Two searches are conducted, one of which converges onto a SP already visited, but a new SP is discovered at (-11.6, 0.12). At this stage, the second SP is selected and the process is repeated until no new SPs are located below a given threshold value of the objective function. Finally, a map of the minima is generated and the global minimum reported as the lowest minimum found. This corresponds to the best estimated parameter set, which in this case is log (Permeability) equal to -11.7 and initial gas saturation equal of 0.3.

While the two-parameter Darcy experiment test problem is too small and simple to truly test the performance of the various methods, it illustrates well how different they are. The GOUST method is somewhat systematic in that local minima are identified one after another. The gradient of the objective function is used to navigate on the objective function surface. As a result, the method can deal with a large number of adjustable parameters without a significant increase in computational effort. The most important aspect of the GOUST method is a rather slow increase in computational effort with the increase in the number of parameters.

4. Model 2: Laugarnes geothermal area.

A second illustration of the proposed global optimization method is given by inverse modeling of a geothermal field called Laugarnes, located in Reykjavík, Iceland. This field has been studied extensively (Thorsteinsson and Eliasson, 1970) and only a brief description is presented here. The Laugarnes geothermal area is fed by three aquifers: A, B and C, with water temperature of 110-120, 135, and 146 ºC, respectively. Tuffs and sediments act as aquicludes between the aquifers. The active reservoir underlies an area of 5 km^2 and has a base temperature about 145 ºC (Bodvarsson and Zais, 1978). Prior to exploitation, the hydrostatic pressure at the surface in the geothermal field was 6-7 bars (Gunnlaugsson et al., 2000) and about 10 l/s of 88 ºC water issued in free flow from the hot spring (Thorsteinsson and Eliasson, 1970).

A model of the area was constructed using mainly a hexagonal Voronoi mesh with 38 volume elements, covering an area of 12 km^2 , see Fig. 8. The model extends to a depth of 2235 m in 8 layers. There is a single volume element in layers 1 and 8, both of which are inactive and represent the reservoir top and bottom. Layers 3, 5 and 7 represent aquifers A, B and C respectively. Layers 2, 4 and 6 represent aquicludes and were assigned lower permeability values, see Fig. 9.

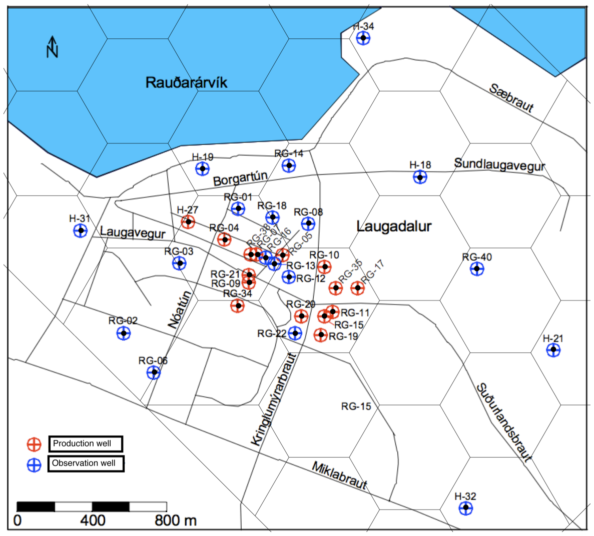

Figure 8. *An aerial view of the central region of the Laugarnes geothermal area. The Voronoi mesh used for the TOUGH2 modeling is shown. Red and blue circles are production and observation wells, respectively.*

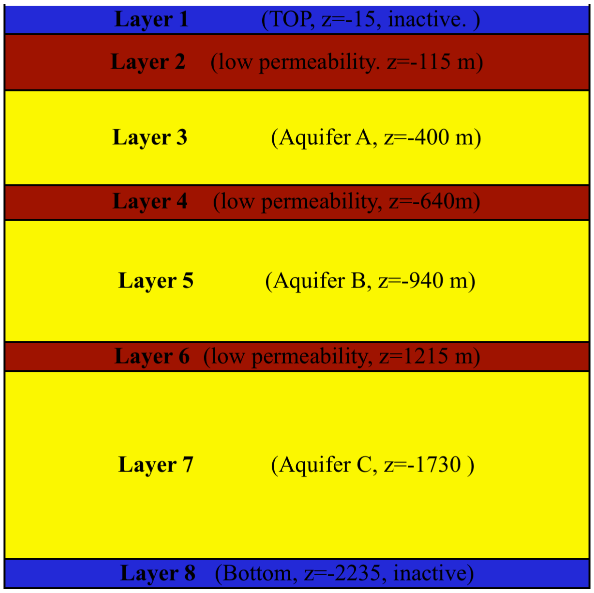
Layer 1 (TOP, z=-15, inactive.)
Layer 2 (low permeability. z=-115 m)
Layer 3 (Aquifer A, z=-400 m)
Layer 4 (low permeability, z=-640m)
Layer 5 (Aquifer B, z=-940 m)
Layer 6 (low permeability, z=1215 m)
Layer 7 (Aquifer C, z=-1730)
Layer 8 (Bottom, z=-2235, inactive)

Figure 9. *View of layers the model of the Laugarnes geothermal reservoir. Colors correspond to different values of permeability and z is depth.*

A three-dimensional representation of the model is depicted in Fig. 10. Two types of sources have been included. First, a mass source located at the bottom of the reservoir was positioned in the area where the upflow is thought to be located. Second, heat sources were placed at scattered positions on the bottom. Six calibration points are used, as shown in Fig. 10. The red line represents the observation well and it has 4 calibration points at 4 different depths. Starting from the top, point 1 represents the pressure at the top of the reservoir. Points 2, 3 and 4 represent the temperature in aquifers A, B and C, respectively. Points 5 and 6 are in the same location and represent water flow rate from the hot spring and enthalpy of the water, respectively.

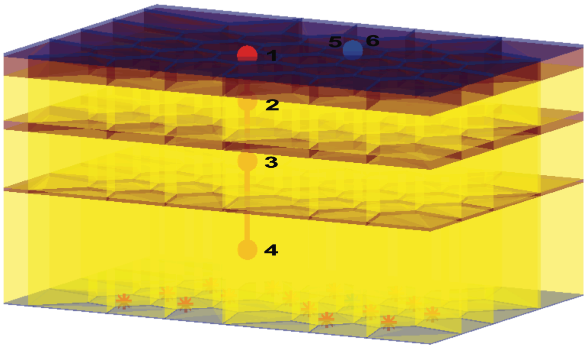

Figure 10. *A 3-dimensional view of the model of the Laugarnes geothermal reservoir. Colors correspond to values of permeability. From the top, points from 1 to 4 in red color and 5 to 6 in blue color are calibration points. The red asterisks in the bottom layer represent the heat sources.*

The model has been constructed to represent a realistic system while being simple enough to serve as a convenient test problem for inverse modeling. The data at the calibration points was artificially generated using iTOUGH2. Gaussian noise was then added to represent observed field data. The inverse modeling is applied to determine the natural state of the system. The fixed parameters in the model, were chosen to have reasonable values consistent with what has previously been reported for this reservoir. By fitting generated data instead of field data, the location of the global minimum is known beforehand, making it easier to analyze the accuracy of the simulated results.

4.1. Objective function

The objective function (see Eq. 5) is chosen to be the squared deviation of the calculated pressure from the observed pressure values at calibration point 1, temperature at calibration points 2 - 4 and water flow rate and water enthalpy at calibration points 5 and 6.

$$S(p_1,p_2)=\sum_{i=1}^{100}\frac{[P_i^*-P_i(p_1,p_2)]^2}{\sigma_{P_i^*}^2}+\sum_{i=1}^{300}\frac{[T_i^*-T_i(p_1,p_2)]^2}{\sigma_{T_i^*}^2} \qquad (5)$$

$$+\sum_{i=1}^{100}\frac{[H_i^*-H_i(p_1,p_2)]^2}{\sigma_{H_i^*}^2}+\sum_{i=1}^{100}\frac{[Q_i^*-Q_i(p_1,p_2)]^2}{\sigma_{Q_i^*}^2}$$

Here, p_1 and p_2 are the two variable parameters: logarithm of permeability and logarithm of flow rate in this case. P, T, H and Q are pressure, temperature, water enthalpy and flow rate respectively. The superscript '*' denotes measured data and σ^2 is the variance.

Fig. 11 shows the variation of the objective function as the logarithm of the permeability of aquiclude layers and the logarithm of the mass flow rate are varied. The grid search method implemented in iTOUGH2 was used to obtain the data for the figure. This provides the complete topography of the function for this two-dimensional parameter space. However, this method is computationally too demanding for most applications (Finsterle, 2007) where more parameters are typically being varied and is used here only for illustrative purposes. Three minima are present on the objective function surface. The global minimum is known to be located at log(Permeability)=-14.00 and log (Mass flow rate)=1.00 and hereafter will be referred to as m2. Two local minima are also present, one at (-16.46, 1.10), (denoted m1) and another at (12.79, 1.76), (denoted m3). The saddle point between minimum m1 and m2 is labeled as sp1 and the one between minimum m2 and m3 is labeled as sp2.

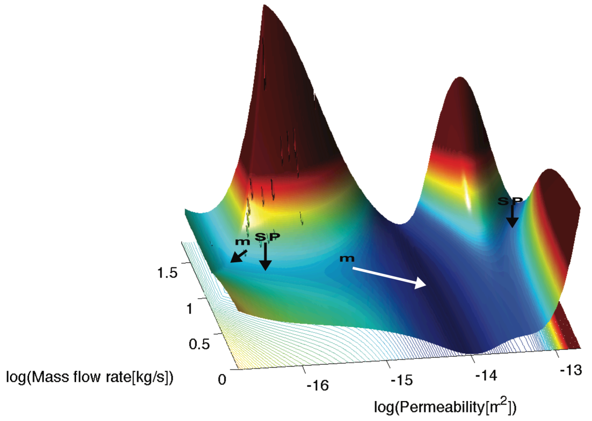

Figure 11. *The variation of the objective function for inverse modeling of the Laugarnes geothermal reservoir as two parameters are varied: logarithm of permeability and logarithm of mass flow rate. For values of log (Permeability) smaller than -15.5 and values of log (Mas flow rate) larger than 1.2, point-like spikes can be seen, possibly because of numerical instabilities in the forward model. The area where these occur is near a maximum, away from the minima and saddle points, thus not affecting the GOUST calculation significantly.*

Both local minima correspond to a significantly higher value of the objective function, but as mentioned earlier they will attract minimization paths started from nearby regions in parameter space.

Fig. 11 shows that point-like spikes are present which may be caused by numerical instabilities in the forward model. It is not clear at this point what causes these instabilities, but since they occur only where the objective function is relatively large, away from the minima and saddle points, the GOUST calculation is not severely affected.

4.2. searching for multiple minima.

Fig. 12 illustrates how the GOUST algorithm finds the global minimum as well as the two local minima for the Laugarnes model. From an initial guess of the parameter values of (-16.38, 1.40), a local minimization is conducted using the conjugate gradient minimization method, converging on a local minimum located at (-16.38, 1.40). From there, a random displacement is generated and a SP search conducted using the MMF method. A SP is found at (-15.96, 0.93). To continue, a displacement along the minimum mode is made and a new local minimization conducted to converge onto a new minimum located at (-14.0, 1.0). The same process is repeated at the new minimum. Two searches are conducted, one of which converges onto the a SP already visited. From the second search, a new SP is revealed at (-13.0, 1.42) which leads in a subsequent minimization to a new minimum at (-12.76, 1.77). In general, the process is repeated from each new minimum until no more SPs are found below a given threshold value of the objective function. The global minimum as well as the two local minima are, thereby, identified.

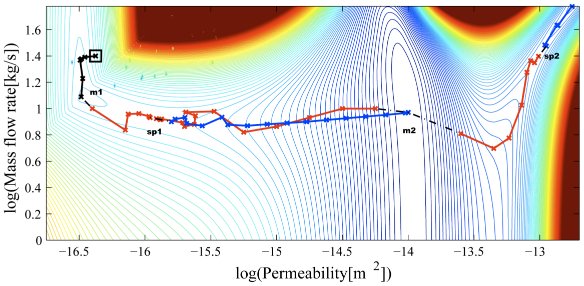

Figure 12: *A GOUST exploration of the objective function for the Laugarnes reservoir inverse modeling. The global minimum as well as two local minima are found by traversing from one minimum to another via first order saddle points. First, starting from an initial guess (square), a minimization is carried out (solid black line) converging to a local minimum (m1). Then, a random increment of parameter values from the minimum is made (dashed black line) and the minimum mode following method used to climb up (red line) the objective function surface to home in on a saddle point (sp1). Then, an increment of the parameter values along the direction of negative curvature away from saddle point is made and another minimization carried out (blue line) converging to a new minimum (m2). The process is repeated from m2 revealing a new saddle point (sp2) and a new minimum (m3).*

The paths in Fig. 12 go through the vicinity of the SPs. The tolerance for convergence onto the SP can be large since the precise value of the objective function there is not important. The fact that the paths taken from one minimum to another go through the vicinity of SPs means that parameter regions with large values of the objective function are avoided. This can be advantageous since unphysical parameter values can lead to convergence problems, ill-defined values of the objective function and large computational effort.

5. Conclusion

The objective functions used in inverse modeling of multiphase flow in porous media can have multiple minima and the task of finding the global minimum can be a challenging one. For objective functions that are continuous and differentiable, the gradient can be used to navigate systematically on the objective function surface so as to move from one local minimum to another. A method we refer to as global optimization using saddle traversals, GOUST, is presented. It is based on climbing up the objective function surface to home in on first order saddle points is used to map out the local minima. This not only gives an estimate of the global minimum (as the lowest minimum found) but also an estimate of the uniqueness of the optimal values found for the parameters.

The applicability of the method to reservoir modeling has been demonstrated by coupling it with the iTOUGH2-TOUGH2 software. The basic features of the method were illustrated with two simple test problems including just two parameters but the method can easily be applied to models involving a large number of parameters. In simulations of atomic scale systems, such saddle point searches are routinely carried out for systems with hundreds or even thousands of parameters (i.e. atomic coordinates) (Pedersen and Jónsson, 2009).

Acknowledgements

This work was supported by a Grant from GEORG (Geothermal Research Group) and the Icelandic Research Fund (RANNIS). We would like to thank Stefan Finsterle for useful discussions as well as providing us access to the latest iTOUGH2 code.

References

Ali M., Pant M., and Abraham A., Improving differential evolution algorithm by synergizing different improvement mechanisms, ACM Transactions on Autonomous and Adaptive Systems **7**, (2), 2012, DOI: 10.1145/2240166.2240170 http://doi.acm.org/10.1145/2240166.2240170, (Article 20 (July 2012), p. 32).

Ali Musrrat and Pant Millie, Improving the performance of differential evolution algorithm using Cauchy mutation, Soft Computing **15**, (5), 2011, 991-1007.

Bodvarsson Gunnar, Zais Elliot. 1978, A field example of free surface testing, Report. https://pangea.stanford.edu/ERE/pdf/IGAstandard/SGW/1978/Bodvarsson2.pdf

Carmichael Gregory R, Adrian Sandu and Potra florian A., Sensitivity analysis for atmospheric chemistry models via automatic differentiation. Atmospheric Environment **31**, (3), 1997, 475–489.

Das S., Abraham A. and Konar A., Automatic clustering using an improved differential evolution algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans **38** (1), 2008, 218-237, DOI: 10.1109/TSMCA.2007.909595.

Finsterle S., 2007. iTOUGH2 User's Guide, LBNL-40040, February; iTOUGH2 Sample Problems, LBNL-40042, February.

Fourie Jaco, Green Richard, and Geem Z.W., Generalised adaptive harmony search: a comparative analysis of modern harmony search, Journal of Applied Mathematics **2013**, 2013, DOI:10.1155/2013/380985, (Article ID 380985, p. 13).

Geem Z.W., Kim J.H. and Loganathan G.V., a new heuristic optimization algorithm: Harmony search. Simulation **76**, 2001, 60.

Geem Z.W., Kim J.H. and Loganathan G.V., Harmony search optimization: application to pipe network design, International Journal of Modeling and Simulation **22**, (2), 2002, 125–133.

Geem Z.W., Tseng C. and Park Y., Harmony search for generalized orienteering problem: best touring in China, Lecture Notes in Computer Science **3412**, 2005, 741–750.

Gunnlaugsson Einar, Gislason Gestur, Ivarsson Gretar and Pall Kjaran Snorri, Low temperature geothermal fields utilized for district heating in Reykjavik, Iceland. In: Proceedings World Geothermal Congress 2000 Kyushu - Tohoku, Japan, May 28-June 10, 2000.

Haario H. and Saksman E., Simulated annealing process in general state space. Advances in Applied Probability **23,** 1991, 866–893.

Henkelman G. and Jónsson H., A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives, *Journal of Chemical Physics* **111**, 1999, 7010.

Henkelman G. and Jónsson H., Long time scale kinetic Monte Carlo simulations without lattice approximation and predefined event table, *Journal of Chemical Physics* **115**, 2001, 9657.

Henkelman G. and Jónsson H., Multiple time scale simulations of metal crystal growth reveal importance of multi-atom surface processes, Physical Review Letters **90**, 2003, 116101.

Joshi R., and Sanderson A. C., Minimal representation multisensor fusion using differential evolution, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, **29** (1), 1999, 63-76, DOI: 10.1109/3468.736361.

Kang S.L. and Geem Z.W., A new structural optimization method based on the harmony search algorithm. Computers and Structures, **82** (9–10), 2004, 781–798.

Karssemeijer L., Pedersen A., Cuppen H. and Jónsson H., Long-timescale simulations of diffusion in molecular solids. Physical Chemistry Chemical Physics **10844**, 2012, 14.

Kim J.H., Geem Z.W., Kim E.S., Parameter estimation of the nonlinear Muskingum model using harmony search. Journal of the American Water Resources Association, **37** (5), 2001, 1131–1138.

Kirkpatrick S., Gelatt C.D., Jr. and Vecchi M.P., Optimization by simulated annealing, *Science* **220**, 1983, 671.

Lampinen Jouni and Zelinka Ivan, 2000. On stagnation of the differential evolution algorithm. In: Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing, pp. 76–83.

Lanczos Cornelious, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of the National Bureau of standards 45 (4), 1950, 255-282.

Lee K.S. and Geem Z.W., A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanical Engineering, **194**, 2004, 3902–3933.

Pedersen A. and Jónsson H., Simulations of hydrogen diffusion at grain boundaries in aluminum, Acta Materialia 57, 2009, 4036-4045.

Pedersen A. and Jónsson H., Distributed implementation of the adaptive kinetic

Monte Carlo method, *Mathematics and Computers in Simulation*, **80**, 2010, 1487.

Pedersen A., Berthet J-C. and Jónsson H., Simulated annealing with coarse graining and distributed computing, *Lecture Notes in Computer Science* **7134**, 2012, 34.

Pedersen A., Henkelman G., Schiøtz J. and Jónsson H., Long time scale simulation of a grain boundary in copper, New Journal of Physics 11, 2009a, 073034.

Pedersen A., Pizzagalli L. and Jónsson H., Finding mechanism of transitions in complex systems: Formation and migration of dislocation kinks in a silicon crystal, Journal of Physics: Condensed Matter 21, 2009b, 084210.

Storn Rainer and Price Kenneth, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization **11** (4), 1997, 341-359, DOI: 10.1023/A%3A1008202821328.

Thorsteinsson, T., Eliasson J., 1970. Geohydrology of the Laugarnes hydrothermal system in Reykjavik, Iceland. In: Proceedings of the U.N. Symposium on the Development and Utilization of Geothermal Resources, Pisa.

Tsallis C. and Stariolo D. A., Generalized simulated annealing. Physica A **233**, 1996, 395–406.

Wang F. S., Jang H. J., 2000. Parameter estimation of a bio-reaction model by hybrid differential evolution. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 410-417.