PROCEEDINGS, 41st Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, February 22-24, 2016
SGP-TR-209

# Calibration of a Geothermal Reservoir Model using a Global Method based on Surrogate Modeling

Ariel Vidal[1,2] and Rosalind Archer[1,2]

[1]Department of Engineering Science, The University of Auckland, [2]The Geothermal Institute, New Zealand.

avid587@aucklanduni.ac.nz

**Keywords:** Calibration, geothermal modelling, surrogate models, global search

## ABSTRACT

Calibration of a multi-phase, geothermal flow model is a crucial step for matching model outputs to available data and hence selecting a good set of parameters for a certain model. In general, it requires a large number of evaluations of an expensive computer simulator. Due to the high cost of each flow model evaluation it is essential that the optimization method selected is able to converge quickly to a good solution. It is well known that nonlinear models may present several local minima, and hence it is possible to get stuck in one of these low-quality solutions using derivative-based optimization techniques only. Global optimization methods are not very efficient, mainly due to the extremely high number of evaluations required for convergence. In this paper we explore the use of a global, derivative-free, optimization method linked with a surrogate model for the calibration of a geothermal reservoir model. The use of a surrogate model allows us to explore the input space in a faster way than using the high fidelity flow simulator. The surrogate model is subsequently trained with new samples suggested from the global method. This workflow is tested in an actual geothermal reservoir model which is complex enough to simulate conditions observed in real life problems. The use of this approach allowed us to calibrate a highly parameterized numerical model using a robust methodology which could be applied to other similar subsurface flow problems.

## 1. INTRODUCTION

Numerical models of heat and mass transfer in the geothermal industry are characterized by a strongly nonlinear relationship between the input parameters and model output, which makes the calibration process a difficult task. The availability of a calibrated model is fundamental for its posterior use as a predictive tool for assessing the behavior of a particular system and in the decision-making process. Additionally, it is ideal if the calibration converges quickly to a good solution. Most of the initial efforts for calibrating geothermal models are based on the use of manual techniques (manual calibration) and lately it is also a normal practice to use automatic tools that implement derivative-based methods, such as gradient based methods which are available in some current software. One important limitation with the use of derivative-based methods is that nonlinear models may present several local minima and hence it is possible to get stuck in one of these low-quality solutions using derivative-based techniques only, unless the initial candidate is close enough to the global solution or if the uncertainty range for the initial parameters has been constrained enough and it only includes one minimum. The previous limitation could be overcome using derivative-free methods, such as global optimization algorithms, but normally these methods require a very large number of function evaluations, which limits their use, even with current computational resources.

In the present work we study the problem of calibration of a highly-parameterized geothermal reservoir numerical model using a different approach from standard methods. In this case we rely on the use of a statistical approximation (surrogate model) to the expensive flow simulator, which will be used for the large number of evaluations required for the global method in its search for the minimum through the input space. As our function is expensive to evaluate, the central idea is to find a combination of input parameters that produces a value that is close enough to the global minimum using only a limited number of function evaluations. This approach is useful where the objective function is continuous, multimodal and expensive to evaluate, additionally the simulator is treated as a black box and derivatives are not available The method will be tested with a geothermal numerical model published by O'Sullivan et al. (2013) for a geothermal prospect in the north of Chile and a set of different combinations of uncertain input parameters will be tested and compared with the results obtained using the derivative-based software PEST (Doherty, 2009).

Similar approaches have been used in engineering applications as well as in the oil and groundwater industry for history matching (Goodwin, 2015). Espinet and Shoemaker (2013) showed an assessment of different local and global optimisation algorithms for parameter estimation of a Tough2 (Pruess, 2004), multi-phase flow model in a geological carbon sequestration context and found that standard derivative-based calibration software as PEST were more efficient when calibrating simpler models, with few parameters, than when calibrating more complex, realistic models with a large number of parameters where surrogate-based optimisation methods appear to be more efficient. These authors concluded that when the number of simulations is limited, surrogate-based algorithms perform best on multi-modal, irregular objective functions, which is expected to be the case for most realistic applications with multi-phase flow models such as those for $CO_2$ sequestration or geothermal reservoir modelling.

## 2. SURROGATE MODELS

Surrogate models, response surface model, meta-models or emulators are basically statistical approximations to the output of a difficult to evaluate phenomena or function, which may represent the value of a physical quantity recorded in the laboratory or the output of a complex numerical model. The principal idea behind this technique is to use previous evaluations of the function under study to build an approximation to the original function with the aim to obtain good and fast estimates of the function under study for non-assessed combination of input parameters. In general, these methods assume some degree of smoothness and correlation for points that are relatively close in the input space for the function under study and use these properties for the estimation of new locations based on data already available. In simple terms, most of surrogate models may be implemented following the next steps:

1. Select an initial set of points where to evaluate the expensive function (Initial Experimental Design).

2. Proceed to evaluate the expensive function at the points selected in Step 1.

3. Fit a surrogate model to the points generated in Steps 1 and 2.

4. Use the surrogate model to predict the value of interest for unsampled points.

### 2.1 Radial basis function interpolation method

In the present work we rely on the use of Radial Basis Functions (RBF) as the interpolation method (surrogate model), but other methods such as Gaussian processes, neural networks or some forms of polynomials could have been used as well. For examples of surrogate modelling in a geothermal context please refer to Vidal and Archer (2014 and 2015) for the use of Gaussian processes and Quinao et al. (2015) for the use of polynomials.

Following the description of Powell (1992) the RBF method needs a set of points $x_1,...,x_n \in \mathbb{R}^d$ and the function values on these points $f(x_1),..., f(x_n)$ are assumed to be known, the RBF algorithm used in this work has the general form:

$$s_n(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) + p(x), \qquad x \in \mathbb{R}^d \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm, $\lambda_i \in \mathbb{R}$ for $i = 1,...,n$, $p(x)$ is a linear polynomial with $d$ variables and $\phi$ has the cubic form: $\phi(r) = r^3$, although other forms could have been used such as Gaussian, thin plate spline or multiquadratics. According to Regis and Shoemaker (2013) the cubic form has been used successfully in previous similar studies.

Additionally it is needed to define a matrix $\Phi \in \mathbb{R}^{nxn}$ by: $\Phi_{ij} = \phi(\|x_i - x_j\|), i, j = 1,...,n$ and the matrix $\mathrm{P} \in \mathbb{R}^{n\times(d+1)}$ so its $i$th row is $\left[1, x_i^{\mathrm{T}}\right]$. The RBF model that interpolates the points $(x_1, f(x_1)),...,(x_n, f(x_n))$ will be obtained by solving the next system of equations:

$$\begin{pmatrix} \Phi & \mathrm{P} \\ \mathrm{P}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} \mathrm{F} \\ 0_{d+1} \end{pmatrix}, \tag{2}$$

where $\mathrm{F} = (f(x_1),..., f(x_n))^{\mathrm{T}}, \lambda = (\lambda_1,...,\lambda_n)^{\mathrm{T}} \in \mathbb{R}^n$ and $c = (c_1,...,c_{d+1})^{\mathrm{T}} \in \mathbb{R}^{d+1}$ consists of the coefficients of the linear polynomial $p(x)$.

## 3. SURROGATE-BASED GLOBAL OPTIMIZATION

Global optimization methods were proposed with the aim of finding the global minimum (or maximum) of a function for problems were it is not efficient to obtain the gradient of the function, either because the irregularity of the function or for the presence of several local minima. With this objective many different methods have been proposed in the literature such as pattern search or heuristic methods (simulated annealing, genetic algorithms or particle swarm). For a more exhaustive list with some examples and references please refer to the work of Regis and Shoemaker (2013). However, many of these methods are not very efficient for expensive functions, such as those found on reservoir modelling because they may require a very large number of function evaluations to obtain a relatively good solution. An alternative solution to this problem is the use of a surrogate model instead of the expensive function, so all the intensive calculations are carried out in this interpolator. These types of methods were proposed some time ago (Kushner, 1962) but they have become popular only recently and many successful applications have been reported for several fields of research during the last years. A popular technique that has been widely used is known as efficient global optimization (EGO; Jones et al., 1998) which is based on the kriging interpolator and introduces the concept of expected improvement for the selection of the next points where to evaluate the expensive function. Vidal and Archer (2015) used the EGO algorithm for the calibration of a synthetic geothermal reservoir finding that it works well for problems with a moderately number of dimensions. Regis and Shoemaker (2007 and 2013) present a good overview of many of these methods.

Following the steps mentioned in Section 2 for the implementation of a general surrogate model the next steps may be added for the use of a general surrogate-based global search method:

5. From a group of candidate points select the next point/points for function evaluation based on a certain user-defined criteria.

6. Evaluate the function at new points from Step 5.

7. Update the surrogate model with the new set of points derived from steps 5 and 6.

8. Repeat until the number of function evaluations is exhausted or until any of the other user-defined stopping criteria is reached

Step 5 is very crucial and it is where most of the surrogate-based global methods differ. How and when the potential next candidates to evaluate are defined is very important. For example, in some implementations the user needs to declare a priori a set of points where the method will choose the next iterates, contrarily there are methods where the next iterates are selected from a group of points which are defined after each iteration depending on the location of the best point evaluated until now. The criteria used for assessing the different potential candidates is very influential for the method's convergence and efficiency. Some common methods under use will select the point or points which maximize the expected improvement quantity (or some other quantity) as mentioned before, other methods select the points that minimize the value of the current surrogate model for each iteration or criteria based on the distance between previous evaluated points and new ones could be used. Selection of one or a combination of criteria will depend on the problem at hand and on the numerical implementation of the method.

### 3.1 DYnamic COordinate search – DYCORS-LMSRBF method

In the present work we will focus in a method developed by Regis and Shoemaker (2013) for the bound-constrained optimization of high-dimensional, expensive, black-box problems which use RBF as a surrogate model. When using this method the next candidate for evaluation is selected from a set of random trial solutions obtained by perturbing only a subset of the coordinates of the current best solution. The perturbation of the coordinates is normally distributed and uses decreasing perturbation probabilities on randomly selected coordinates, which means that the number of coordinates perturbed tends to decrease as the algorithm progresses. At the beginning, depending on the dimensionality of the problem, all coordinates of the current best solution are perturbed when generating the next candidates and, as the number of function evaluations increases, the probability of perturbing a given coordinate decreases, and so the coordinates of the best solution that are perturbed tend to become fewer. The motivation for this idea came from the experience of the authors when applying manual calibration of watershed models where they found that early in the search perturbing all coordinates is helpful in improving an initially poor solution, but as the solution improves it becomes necessary to perturb only a few coordinates simultaneously so that the current improvement is not lost. This method use a weighted score between two criteria for selecting the new iterate to evaluation in the expensive function: estimated function value from the surrogate model and minimum distance from previously evaluated points. A weighted score of these two criteria is used because it is desirable to seek a trial point with a low surrogate value and that is also far from previously evaluated points in order to improve the current RBF model (Regis and Shoemaker 2013).

This method, like other similar methods, starts by evaluating the expensive function at the points derived from an initial experimental design, which in this case corresponds to a symmetric Latin hypercube design, that have proved to be more efficient than non-symmetric designs for large-dimensional problems (Ye et al., 2000). In addition, DYCORS does not need to solve an optimization problem during each iteration, in contrast to EGO (and other popular methods) which needs to find the global solution for the function that evaluates the expected improvement quantity over the input domain which makes the DYCORS methods more efficient.

## 4. CALIBRATION OF A GEOTHERMAL NUMERICAL RESERVOIR MODEL

The DYCORS-LMSRBF algorithm will be used for the calibration of a geothermal reservoir numerical model of the Pampa Lirima basin in the north of Chile (O'Sullivan et al, 2013). The Pampa Lirima project is one of a series of potential fields where geothermal energy could be harvested in Chile and the conceptual and numerical model were developed as a representative case study of a high Andean hydrothermal system. The Tough2 model was built with available surface and subsurface data and according to the authors it has proven to be a useful tool for improving the understanding of the Pampa Lirima system and may be useful in planning its subsequent development.

The extent of the model was defined to include all the important features of the geothermal system and was positioned so that the lateral boundaries of the model do not have a large influence on the model behavior. Consequently an area of 24x15 km was considered to be part of the model with a depth of 4.7 km. As there is little information about subsurface conditions the aim of this preliminary model was to study the large-scale behavior of the system and to investigate the interaction between two different hot springs areas present in the basin. Consequently a coarse mesh was used with 24, 15 and 37 blocks in the x, y and z directions, respectively (Figure 1). The high number of vertical layers was necessary in order to take into account the large range in topography, while also ensuring that the subsurface flow is accurately represented.

Figure 1 presents two different views of the model geology. The upper section shows the surface geology and the lower cross-section presents a vertical view with the distribution of geological units. In total, the model includes 63 different rock types whose permeabilities values were assigned according to available hydraulic conductivities measurements for the shallower rock units and through a manual calibration process for the deeper rock units. Boundary conditions of the model were determined using available information regarding the regional heat flux, precipitation, surfaces temperature and closed lateral boundary conditions were assumed. For more details about the construction and implementation of the numerical model please refer to O'Sullivan et al. (2013).

Once the model had been set up and calibrated it was used to estimate the steady-state of the system (Figure 2), which agree well with the available data suggesting that the Pampa Lirima system is structurally controlled with a number of faults interacting to provide the mechanisms for both upflow and recharge.

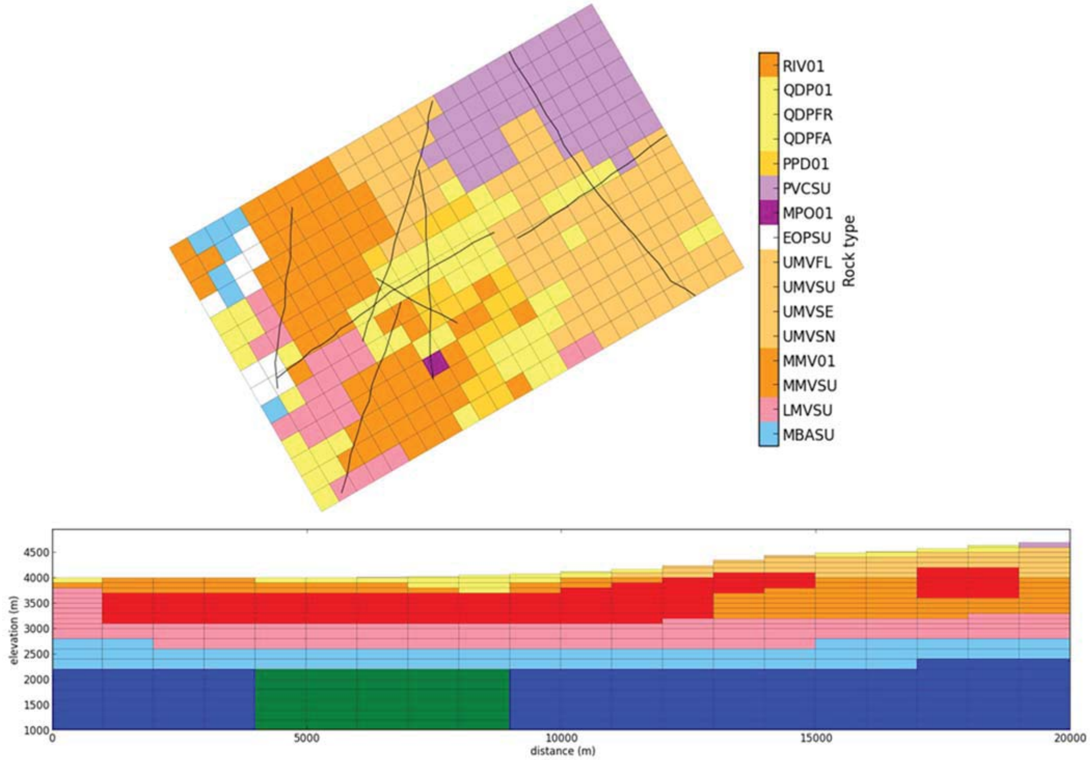


**Figure 1: Two different views of geological heterogeneity in the Pampa Lirima's model. Upper section shows the surface geology and lower section gives a view of a cross section of the model (O'Sullivan et al., 2013).**

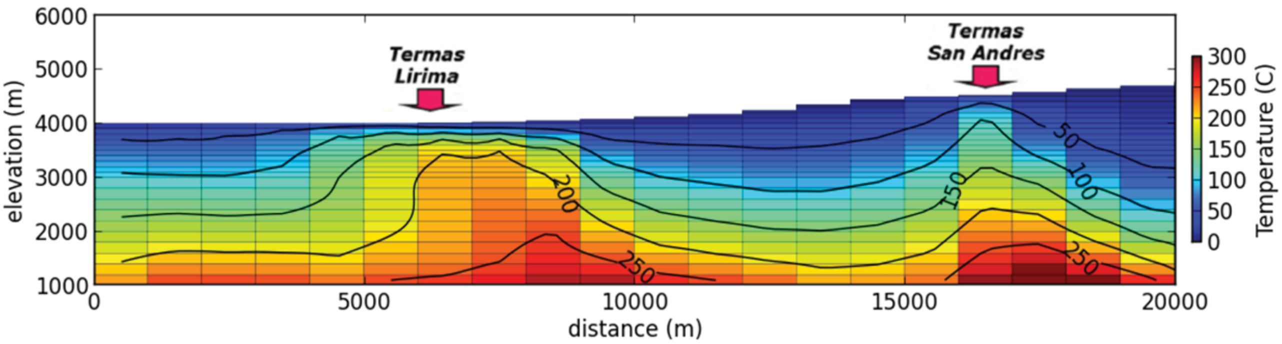


**Figure 2: Cross-section showing temperature distribution of the steady-state of the Pampa Lirima model (O'Sullivan et al., 2013).**

### 4.1 Computational experiments using DYCORS

With the calibrated model presented in the previous section, a series of computational experiments were carried out to test the effectiveness and reliability of DYCORS algorithm in a geothermal context. For the numerical implementation of the method the Python package pySOT, developed by Eriksson et al. (2015), was used in conjunction with the Python library PyTOUGH, which allows the control of Tough2 simulations with the use of Python commands (Croucher, 2011). pySOT is an optimization toolbox for global deterministic optimization problems which includes most of the methods discussed in the previous sections. The main purpose of the toolbox is for the optimization of computationally expensive black-box objective functions with continuous and/or integer variables where the number of evaluations is limited.

Using the model from O'Sullivan et al. (2013) as a reference the objective of the experiments is to try to replicate the steady-state temperature distribution of the Pampa Lirima model using only sparse data derived from a set of hypothetical vertical wells distributed in the basin where temperature measurements are available. So, these temperature values from the reference model are taken as the *true* values of temperature of the hydrothermal system and through the use of an objective function the idea is to find the best set of parameters that match these measurements. The objective function used in this case is the squared sum of the residuals between the temperature values in the wells from the reference model and temperature values in the wells from the new runs generated in the calibration process. But before this value can be calculated, the data from the reference model was modified by adding a noise signal to the original temperature values from a normal distribution with zero mean and a standard deviation of 0.25 ˚C to mimic measurement uncertainty in real life applications. The value of this objective function corresponds to the expensive function which we will try to reproduce through a RBF model and then we want to minimize to obtain a set of parameters that produce a good match with observed data (temperature values from vertical wells).

As a way of testing the efficiency of the method a set of PEST runs were initialized for each of the different experiments carried out. We decided to compare both methods based on the value of the objective function after a fixed number of function evaluations, since the time required for one function evaluation is considerably higher than the time used by the optimization algorithm to choose the next point where to evaluate.

4.1.1 189-Dimensional Problem

In this set of experiments the values of permeability in the three main directions x, y and z are considered uncertain in the analysis for each rock type which gives a total of 189 uncertain parameters. The rest of parameters remain fixed and identical to the reference case.

In a first experiment we assumed that the values for each of the 189 parameters should lie in a range of 1 logarithmic unit (when trying to calibrate permeability values is highly recommended to log transform these values), so the input space is a 189 dimensional hypercube of side 1, and the *true* values are considered to lie inside these bounds. The next step before starting the optimization process is to define the initial experimental design which in this case is a symmetric Latin hypercube of size $2\times\left(n\text{dim}+1\right)$, where $n\text{dim}$ is the number of parameters allowed to be modified in the process (dimensionality of the problem). This number of points for the experimental initial design is suggested by pySOT developers (Eriksson et al., 2015). For this experiment a total of 600 function evaluations were allowed, which is a reasonable number taking into account the dimensionality of the problem.

Figure 3 presents the evolution of the objective function against the number of function evaluations; hence we plotted the lowest value of the objective function found up to now versus the total number of simulations carried out. As can be seen in Figure 3(a), both methods converged to similar values of the objective function after 600 evaluations. In the case of pySOT it reached a value of 3,340 and PEST a value of 1,158 after 3 iterations for the objective function, respectively. The stepped section displayed by PEST is due to the calculation of the Jacobian matrix, which requires one function evaluation for each variable (189 in this case) and needs to be calculated for each iteration. It is worth to mention that the value of the objective function for the set of reference parameters is 15.3, so neither of the two methods was able to locate the global solution after 600 forward runs of the Tough2 model. As the initial candidate for the PEST runs presented in Figure 3 we selected randomly one of the points from the initial experimental design of each pySOT run, with the exception of Figure 3(c) where the curve labeled as PEST_2 used as initial point the best solution from a previous pySOT run.

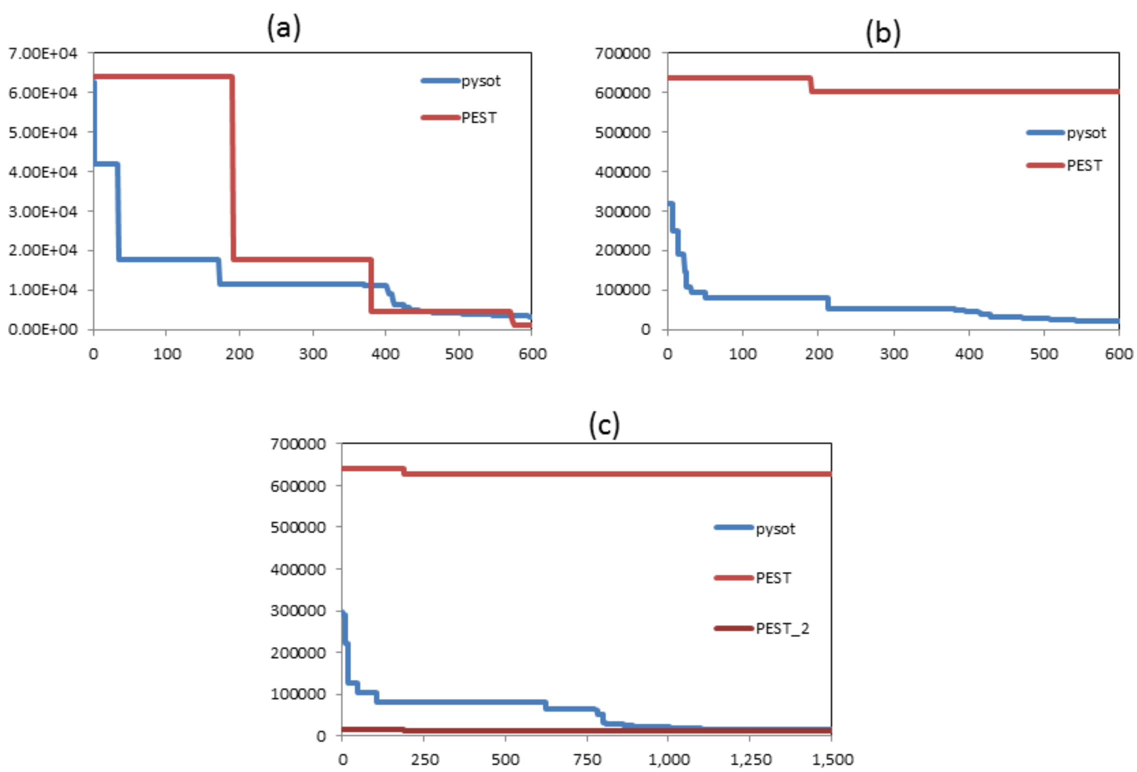


**Figure 3: Evolution of the objective function against the number of function evaluations for the 189-dimensional problem.**

For the experiments presented in Figure 3(b) and (c) we modified the range of uncertainty for each of the 189 parameters, so now instead of 1 logarithmic unit we increase this value to 3 logarithmic units, which could be the case when there is large uncertainty about actual values. As can be seen from this Figure 3(b) pySOT was able to decrease the value of the objective function until it reached a value of 20,500 after 600 function evaluations. PEST on the contrary was only able to decrease slightly this value in the first iteration from 635,974 to 603,627 where it got stuck for successive iterations.

In the last experiment with the 189-dimensional problem, we decided to increase the number of allowed function evaluations to 1,500 with the aim of investigating if the methods could take extra benefit from these additional function evaluations. A second modification was that we changed the size of the initial experimental design in the pySOT run to $4\times(n\dim+1)$ , so we expected a better exploration of the input space. Figure 3(c) presents a similar picture as Figure 3(b), pySOT indeed improved its performance when compared with previous experiments, and was now able to decrease the value of the objective function further to a value of 14,190 and PEST again had some trouble decreasing the value of the objective function from 641,899 to 627,130. As an additional experiment we took the set of parameters from the best solution of this pySOT run as the initial candidate for a new PEST run (PEST_2 curve in Figure 3(c)) with the purpose of assessing the combination of a global and an inherently local method. This PEST run was able to decrease the value of the objective function from 14,190 to 12,986 where it got stuck in a local minimum. This reduction in the value of the objective function is marginal compared with the original reduction that pySOT was able to reached in this problem.

### 4.1.2 126-Dimensional Problem

In this section we present the results of a set of experiments where we considered that only the permeabilities in the x and y direction are uncertain, while in the z direction we used the permeabilities from the reference model, so totalizing 126 uncertain parameters. For these experiments a total of 400 forward evaluations is deemed as sufficient and we considered an initial experimental design of size $2\times(n\dim+1)$ . For these PEST runs instead of using a random point from the initial design as the initial candidate we choose the first point of the initial experimental design, so both methods begins from the same value of the objective function.

Figure 4(a) shows the evolution of the value of the objective function for the first experiment in which the range of uncertainty for each parameter is equal to 3 logarithmic units, representing the *uncertain* scenario. In this example both methods made a good job decreasing the value of the objective function from 305,600 to 13,200 for pySOT and to a value of 18,950 for PEST. However, as can be seen, pySOT was consistently more efficient than PEST, especially for a low number of function evaluations, since only from the third iteration PEST was able to found similar values to pySOT. The need to calculate the Jacobian matrix for some derivative-based methods make it harder to found good solutions when there is a *tight* budget for function evaluations, instead methods like pySOT offer a good alternative when there is no time or resources for a large or very large calibration routine with a considerable number of function evaluations.

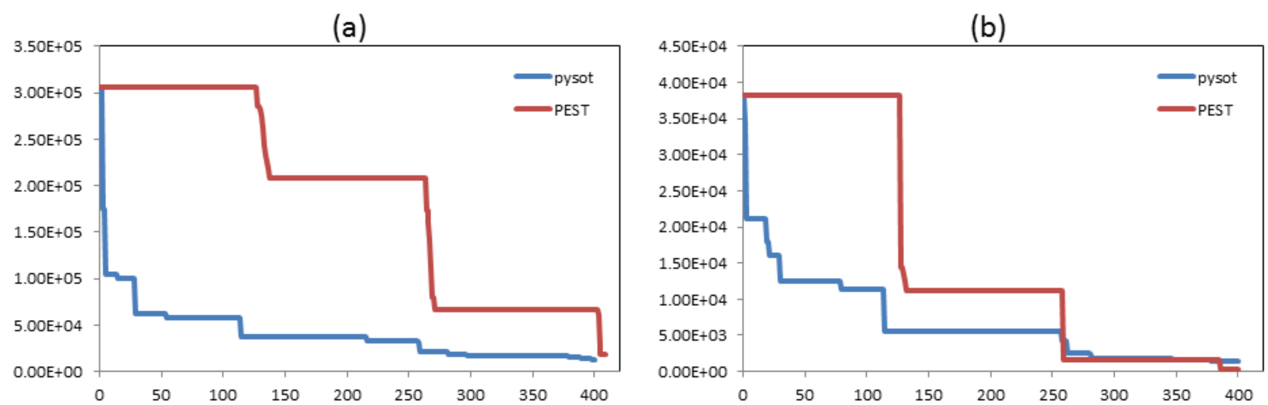


**Figure 4: Evolution of the objective function against the number of function evaluations for the 126-dimensional problem.**

In the second case (Figure 4(b)), the range of variability for each uncertain parameter was reduced to a value of 1 logarithmic unit, in the so-called *constrained* scenario. The rest of parameters are fixed from scenario (a). As has been seen in the previous examples pySOT is able to decrease the value of the objective function from very early in the optimization process to a value of 1,518 after 400 function evaluations. In this case PEST only required 2 iterations for a decrease of the objective function to values comparable with pySOT, from iteration 3 PEST was able to decrease further this value to 210, which is very close to the value using reference parameters (15.3).

### 4.1.3 63-Dimensional Problem

In this last set of experiments only the permeabilities in the x direction are considered able to vary, and permeabilities in the y and z directions are fixed from the reference model presented in section 4. According with this reduction in the number of free parameters only 200 forward evaluations are considered for this analysis. In the same way as the 126-dimensional problem we fixed the size of the initial experimental design to $2\times(n\dim+1)$ and for the PEST runs we choose the first point of this experimental design as the initial candidate for the optimization routine.

Figure 5 presents the results for the two scenarios investigated in the 63-dimensional problem. Figure 5(a) shows the evolution of the value of the objective function for the uncertain case where each parameter is considered to be able to vary in a range of 3 logarithmic units. In this case, in a similar way than the 189-dimensional problem, PEST did not make a good job when decreasing the objective function; instead it got stuck after the first iteration at a value of 137,155. pySOT for its part was able to decrease this value two orders of magnitude until a value of 9,765. In this case pySOT made a considerable reduction of the objective function in the very first iterations and then a long flat section begins from this value, until 98 function evaluations where the objective function start to decrease again.

Figure 4(b) shows the results from the constrained scenario where each variable is considered to lie in a range of 1 logarithmic unit. In this case, as one might have expected, PEST was able to find a very good solution, so after 3 iterations it decreased the value of the objective function to a value of 236, which is the lowest value found until now. For its part pySOT after 200 function evaluations was able to decrease the value of the objective function to a value of 1347. In this example pySOT was more efficient than PEST only in the first iterations (<64) before PEST made the calculation of the Jacobian matrix in the first iteration, from then PEST was more efficient for decreasing the objective function.

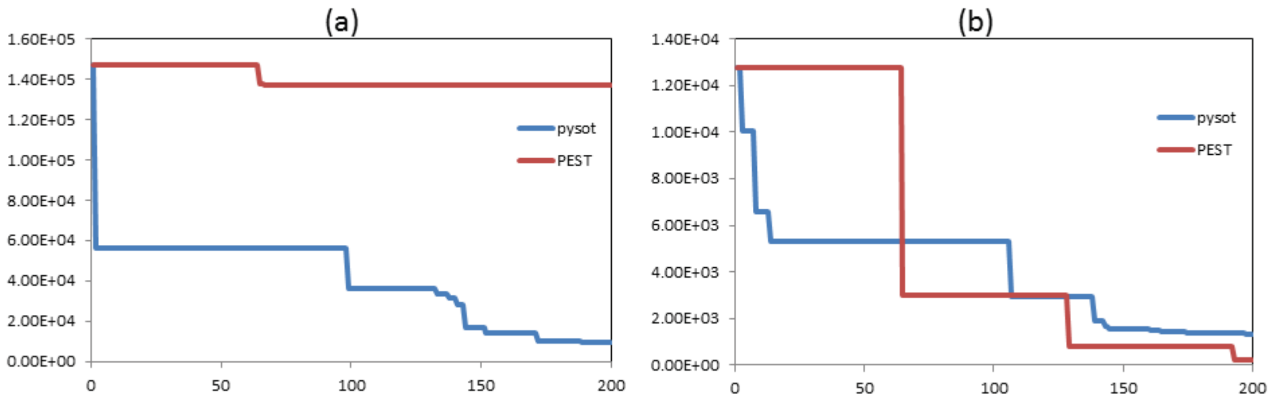


**Figure 5: Evolution of the objective function against the number of function evaluations for the 63-dimensional problem.**

## 5 CONCLUSIONS

In this work we present some preliminary results on the use of the pySOT toolbox, which is a global optimization method based on the use of a surrogate model, for the calibration of a geothermal Tough2 numerical model. Three different scenarios were studied in the present work: a 189, 126 and 63-dimensional problem respectively, and for each of them two variations were considered: an uncertain case and a constrained case, based on the ranges (lower and upper bounds) that each parameter is allowed to move freely. All these models are based on a reference study taken from a real geothermal field. As can be seen from the examples previously discussed pySOT was able to decrease the value of the objective function, by at least two-orders of magnitude, for each of the dimensionalities considered. Secondly, pySOT results were not affected too much when the variability of each parameter was increased, which places pySOT as a robust and efficient method for the calibration of non-linear, expensive functions.

When compared with a popular, derivative-based software as PEST, pySOT proved its reliability for each of the problems considered, suggesting that surrogate-based global optimization methods are very promising for the calibration of high dimensional problems with a large uncertainty, as could be expected in early stages of development of a geothermal prospect when there is not much subsurface data available. The need for calculating the Jacobian matrix for each iteration of a PEST routine means that methods such as pySOT are relatively efficient as reductions in the value of the objective function can be seen from early stages in these global methods, which could be very beneficial when there is a tight budget for running forward models and there are not so many function evaluations available.

## ACKNOWLEDGEMENTS

The authors would like to thank David Eriksson from the Center for Applied Mathematics, Cornell University for his valuable help on the use of pySOT toolbox and for some discussions about theoretical aspects of its implementation.

## REFERENCES

Croucher, A.E.: PyTOUGH: a Python scripting library for automating TOUGH2 simulations. *Proceedings,* 33th New Zealand Geothermal Workshop (2011).

Doherty, J.: PEST: model-independent parameter estimation. Brisbane, Australia: Watermark Numerical Computing. (2009).

Eriksson, D., Bindel, D and Shoemaker, C. Surrogate Optimization Toolbox (pySOT). github.com/dme65/pySOT. (2015).

Espinet, A. J. and Shoemaker C. A.: Comparison of Optimisation Algorithms for Parameter Estimation of Multi- Phase Flow Models with Application to Geological Carbon Sequestration. *Advances in Water Resources*, 54, 133-148. (2013).

Goodwin, N.: Bridging the Gap Between Deterministic and Probabilistic Uncertainty Quantification Using Advanced Proxy Based Methods. Houston, Texas, *Reservoir Simulation Symposium*. SPE-173301-MS Society of Petroleum Engineers. (2015).

Jones, D. R., Schonlau, M., Welch, W. J.: Efficient Global Optimisation of Expensive Black- Box Functions. *Journal of Global Optimisation*, 13(4), 455-492. (1998).

Kushner, H.: A versatile stochastic model of a function of unknown and time-varying form. *Journal of Mathematical Analysis and Applications*, **5**, 150–167. (1962).

O'Sullivan, J., Vidal, A., Yáñez, G. and Muñoz, J.: Modelling Hydrothermal Fluid Flow in a High Andean Setting. *Proceedings,* 35th New Zealand Geothermal Workshop (2013).

Powell, M.J.D.: The theory of radial basis function approximation in 1990. In:W. Light, ed. Advances in numerical analysis. Vol. 2. Wavelets, subdivision algorithms and radial basis functions. Oxford, UK: Oxford University Press, 105–210, 1992.

Pruess, K.: The TOUGH codes – A family of simulation tools for multiphase flow and transport processes in permeable media. *Vadose Zone Journal* 3(3), 738–746. (2004).

Quinao, J.J. and Zarrouk, S.J.: Probabilistic resource assessment using the Ngatamariki numerical model through experimental design and response surface methods (ED and RSM). *Proceedings,* 37th New Zealand Geothermal Workshop (2015).

Regis, R.G. and Shoemaker, C.A.: A stochastic radial basis function method for the global optimization of expensive functions. INFORMS *Journal on Computing*, 19 (4), 497–509. (2007).

Regis, R.G and Shoemaker, C.A.: Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization* 45 (5), 529-555. (2013).

Vidal, A. and Archer, R.: Bayesian Emulation Of Geothermal Numerical Models: A Synthetic Reservoir Case-Study. *Proc. 36th New Zealand Geothermal Workshop*, Auckland, New Zealand. (2014).

Vidal, A. and Archer, R.: Surrogate-based Modeling and Calibration of a Synthetic Geothermal Reservoir Model: Looking for efficient ways of calibration in highly non-linear problems. *Proceedings,* 17th Annual Conference of the International Association for Mathematical Geology, Freiberg, Germany. (2015)

Ye, K.Q., Li,W., and Sudjianto, A.: Algorithmic construction of optimal symmetric latin hypercube designs. *Journal of Statistical Planning and Inference*, 90 (1), 145–159. (2000).