

VOLSUNG: INVERSE MODELLING AND UNCERTAINTY ANALYSIS USING PEST

Peter Franz¹, Jonathon Clearwater¹

¹Flow State Solutions Ltd, New Zealand

Peter.Franz@FlowStateSolutions.co.nz

Keywords: *Volsung, Geothermal, Simulation, Inverse Modelling, PEST, Cloud Computing*

ABSTRACT

The PEST suite of tools has long been established as a standard toolkit in groundwater and other geoscience modelling. Its inverse modelling and uncertainty analysis capabilities allow modelers to greatly shorten model development time, improve model calibration, quantify uncertainty in model parameters and determine confidence in model predictions.

PEST tools have been used in the geothermal context, in particular in combination with TOUGH2; another noteworthy tool in this arena has been iTOUGH2. However the use of these tools has been greatly hampered by the complexity in setting up and running these types of simulations and the nature of TOUGH2 models. In particular, extraction of model observation data has been a really difficult task; in order to match real-world observations up to four dimensional interpolation has to be used. Since TOUGH2 does not store grid geometry this difficult task was left to the individual user. Further, parallel run management, i.e. the deployment of "workers" on different computational nodes, was also left to the individual modeler.

In this work we introduce a universal coupling between the Volsung suite of reservoir and wellbore simulators and the PEST suite. It is universal in the sense that any model parameter and any model observation is accessible to the inverse modelling process. It fully supports data interpolation, i.e. the modeler can simply enter data by location/type and time. The modeler can select the adequate PEST tool for the task and use the graphical user interfaces to setup the inverse modelling problem. Run management is fully automatic and can create clone workers on cloud service platforms like Amazon Web Services (AWS). This reduces barriers to greater use of inverse modelling and uncertainty analysis in the geothermal industry.

1. MOTIVATION

Calibrating complex non-linear models to data available through field surveys can be a laborious and expensive task. The modeler needs to select a parameter set, prepare model input files, run the model and generate adequate output charts. After analyzing the model outputs, she must make a guess for an improved parameter set based on human intuition. This process is repeated several times over until the output of the model matches the field data adequately. The process could be sped up by running multiple parameter sets in parallel; however it is hard for the human mind to keep oversight when quickly switching between various model outputs and parameter sets.

With modern advances in artificial intelligence, it is plausible that the human modeler could soon be replaced by an artificial modeler who would be cheap to run and who would never require to sleep or rest. This is of course not currently possible, but nor would it be desirable to blindly trust a solution an artificial machine would come up with.

Hence in the real world we must aim for a somewhat smaller target: We want to have an automated process which can improve solutions in a cost-effective way. The human modeler is still required; however, the human's task is now to keep oversight over the calibration process rather than performing repetitive menial tasks.

In mathematics, the field of inverse modelling techniques is used for this purpose. The principle is simple: Calculate a quantity (objective function) describing the mismatch between a real-world observation and its modelled value then select an adequate algorithm which changes the model's input parameters in such fashion as to minimize this objective function. A variety of such optimization algorithms exist, for example the gradient-based Levenberg-Marquardt method being one of the best-known and probably most widely used [Levenberg 1944, Marquardt 1963].

However, inverse modelling techniques go beyond simply finding an optimal parameter set for a model. They can provide parameter sensitivities, i.e. how much a model output changes for a change in given input parameters. They can also be used to provide constraints on model outcomes, i.e. in what range a modelled observation can change in a scenario for a given accuracy of the field data on which the calibration is based. This predictive uncertainty analysis can be a key model output for geothermal operators who want to minimize risk in geothermal developments and gain confidence in model predictions.

2. BACKGROUND

Inverse modelling techniques have been under-utilized in the geothermal modelling context. The reasons for this are manifold. Chiefly, traditional reservoir simulators like TOUGH2 and TETRAD run very slowly. Since inverse modelling requires that a model is run many hundreds or thousands of times it is easy to see that the inverse modelling runtime is prohibitive if a single model run takes in the order of many hours to even days.

Secondly, it has been very hard to set up inverse modelling problems. Take, for example, a reservoir temperature survey along a welltrack. Extracting comparable data from TOUGH2 output would require the modeler to interpolate modelled temperature from the grid block outputs. Since TOUGH2 does not store any grid information the modeler needs to write auxiliary scripts to perform

this 3D interpolation. Once this is done an additional interpolation over time is required if the temperature survey falls between print times.

For TOUGH2 an inverse modelling package called iTOUGH2¹ has been available. iTOUGH2 can help with setting up inverse models in TOUGH2; however its text based input files can be cryptic and it requires specialist knowledge and training to utilize. iTOUGH2 has had a modest amount of use in the geothermal industry [e.g. Moon et al., 2014].

Another inverse modelling package which has been used both in groundwater and geothermal contexts is PEST² (Parameter ESTimation) [Doherty et al., 1994]. PEST is a free software package designed to work with any numerical simulator through a template scheme. Model input parameters are replaced in ASCII input files by parameter tags; PEST will then automatically replace the tags with actual numbers according to its internal algorithms. Model output observations are analyzed via a keyword scheme from the model output files. While this template scheme is very flexible it has probably also been the major hurdle for using PEST with TOUGH2; it typically required the modeler to write additional intermediate scripts to extract, interpolate and format data from the TOUGH2 output files. Nonetheless, there have been successful studies using PEST with TOUGH2 [e.g. Pearson et al., 2015].

For our work here the goal was to enable greater use of uncertainty quantification and parameter estimation with the Volsung Geothermal Simulator Package [Clearwater and Franz, 2019; Franz et al., 2019]. The Volsung package consists of multiple applications used in geothermal numerical simulations: Brynhild for reservoir simulations; Gudrun for wellbore simulations; Sigrun for pipeline simulation; and Swanhild for numerical pressure transient analysis. The Volsung reservoir simulator has been optimized by utilizing parallel computing methods on both CPU and Graphical Processing Unit (GPU) architectures [Franz et al., 2019]. By doing this it reduces model run times by about an order of magnitude when compared with a standard TOUGH2 model, making it a far better candidate for use as the forward model simulator in inverse modelling applications. Prior to this work only Swanhild had built-in inverse modelling functionality [McLean et al., 2020], so the focus of the current work was to add this capability to Brynhild and Gudrun.

Given PEST's functionality it was an ideal tool to couple with Volsung to enable the required inverse modelling capability. Unfortunately, development on PEST has slowed down in recent times. At present there are at least two alternatives which use similar principles, PEST_HP² and PEST++³. PEST_HP input files are not fully compatible with PEST, and our inverse modelling tests of PEST++ were not successful. This means that PEST can't be relied upon to provide a future-proof platform; however, since at its current development status it provides a stable and reliable tool with good functionality for optimization and predictive uncertainty analysis we decided to adopt it as our base tool. At the same time, we are working on a platform which can either completely replace PEST in the future or provide all required functionality not accessible through PEST. It is noteworthy that PEST also has additional features which make it useful as an inverse modelling tool basis. A parallelized version called BEOPEST enables inverse modelling in parallel using distributed "agents" which can be located anywhere on a network. This allows good scaling for problems with a high number of variable input parameters. Further, PEST/BEOPEST has an in-built predictive uncertainty analysis mode.

3. IMPLEMENTATION

3.1. General

A key development goal for the inverse modelling tools was that they needed to be flexible enough to work with both the Volsung reservoir simulator Brynhild and the wellbore model Gudrun, plus any other possible future simulators. Further, the new tools should be universal, i.e. allow the modeler to use any parameters and observations present in the model. We achieved this by creating an internal abstraction layer; the actual inverse modelling user interfaces derive directly from this abstraction layer but are only thin software wrappers. The two applications added to Volsung are Gunnar (which works with Brynhild) and Atli (which works with Gudrun). Visually these two applications are nearly identical; internally they only differ by knowing how to run their associated forward model simulators and how to extract data from the output. The following discussion of features hence applies to both applications equally.

3.2. Model Input Parameters

The inverse modelling tool needs to vary certain user-defined model input parameters. In Volsung, these input parameters are now flagged as "parameter tags" in the forward simulation's GUI. In practice, this means that the user clicks on an input field with the mouse and selects to change the value over to a parameter tag for which she must give a name, e.g. "myparameter". The PEST naming convention is adhered so there is a 12 character limit.

Any floating-point input parameter to the model can be changed from a fixed value into a parameter tag, with only a handful of exceptions. For example, parameters relating to times (like print times, flow rate changes in tables) or relating to grid geometry can't be used as parameter tags. This is a sensible restriction since varying these would make the simulator output incomparable to the output from other parameter sets.

¹ <https://itough2.lbl.gov/>

² <https://pesthhomepage.org/>

³ <https://www.usgs.gov/software/pest-parameter-estimation-code-optimized-large-environmental-models>

3.3. Model Observations

Predicting the modeler's requirement for extracting model output relating to field data at first appears daunting since the modeler could in principle ask for anything possible. However, on second thought we realized that the vast majority of real-world field observations fall in some narrow classes:

- Observations of reservoir properties (such as pressure, temperature) along a welltrack
- Generic reservoir properties given in 3D space and time
- Flowing fluid properties inside a wellbore
- Wellhead deliverability curves
- Flowing fluid properties in a surface network, i.e. through a pipe or separator
- 2D field surveys like micro-gravity

Volsung was already able to store the above data from field surveys inside its model files and display them in charts in its GUIs. We hence created observation groups based on the above. The modeler only needs to fill in field survey data into the adequate data tables in the Volsung forward model files, then create a corresponding observation group in the inverse model. The observation groups are fully equipped to deal with extracting and interpolating the model data, both spatially and temporally, to match the given field data points.

Any observations which can't be described by the above can be addressed by writing a small Python script which handles the data extraction and formats the data so it can be recognized by the system.

3.4. Observation Weights

The inverse modelling algorithm requires observation weights which indicate how much emphasis should be given to each individual observation data point. They can be critical to the success of an inverse modelling attempt; however, it can be quite hard for the modeler to define good values. We have simplified this by calculating the individual weights from a three step process.

In the first step the modeler specifies an error; this error should relate to the measured accuracy of a data point. For example a temperature survey may have a typical absolute error of 1°C. The weight of a datapoint is the inverse of this error; a high measurement error results in a low weight. The advantage of this method is that surveys of different quantities – like pressure and temperature – can get combined using relative weights, avoiding objective functions which are dominated by one quantity simply because its numerical value (for instance pressures in pascal) is higher than other quantities (e.g. temperature in degrees Celsius).

The second optional step allows a correction by the number of observations in a group. Consider two temperature surveys with a vast different number of datapoints simply because one survey had a much higher sampling frequency. The basic information in these surveys may be the same but the objective function would be dominated by the survey with the higher sampling frequency. Dividing the weights from the first step by the number of observations in the group makes the two surveys comparable in their information importance.

The third optional step is to multiply the weights by a user-defined factor α . If there are a large number of one kind of survey – say temperature surveys – but only a few surveys of another kind – e.g. pressure – then the objective function would be dominated by the first while largely ignoring the second kind. Giving the second kind more weight can be useful to stress its importance as a dataset.

The above steps give the user a clearly defined choice for the weights which makes their selection process straightforward and transparent. An acceptable strategy might be to try using only the error weighting first; if the inverse model then matches some observation groups much better than others the second and third step options can be employed to improve the match. Note that changing the weights may not require a full re-run of an inverse model. It gives more emphasis to a downhill direction for the objective function which may have been previously masked by numerical errors given by the numerical calculation of the derivatives; the simulation can be restarted at the previous solution and will often converge quickly towards the final solution from there.

3.5. Run Management

In the simplest case the inverse model is run in serial, i.e. by using the serial PEST executable. However, employing parallel forward model runs via using BEOPEST is much preferred to speed up the process.

In order to employ the parallel mode the following steps must be taken:

- A copy of the model directory containing all files necessary for running the forward simulation must be made for each parallel process. Depending on the situation these folder copies can be on the same local machine or on a remote instance.
- A “manager” process must be started on the local machine. This process is lightweight, i.e. even an underpowered computer can run it without problems.
- One or more “agent” processes must be started in the model directory copies. They connect to the manager via the network; the manager then asks each agent to work on a different set of input parameters. Once an agent has finished its job it reports the observation values back to the manager.

The agent processes can either run on the local machine (if it is powerful enough) or be started on remote instances in the network to which the modeler has access. For example, it would be conceivable that the modeler makes use of her colleague’s computers outside of normal business hours. The process of transferring the data to these remote instances and starting the agent processes there is fully automated; the only thing the user needs to do is to perform the initial setup of the remote instances by installing the required software there.

A more convenient and highly scalable method is to use remote instances provided by a cloud computing service provider like Amazon Web Services (AWS)⁴. In this case the modeler sets up one base instance by installing the required software. Volsung can then transfer the forward simulation model to this base instance and create an arbitrary number of “clones” from it. Once the agents on the cloned instances have performed their work the instances are automatically deleted. Hence the cost for employing a large number of computational nodes is kept to the bare minimum of the actual runtime of their service.

4. EXAMPLES

4.1. Inverse Modelling of a Geothermal Reservoir

In this example we created a small model of a geothermal reservoir as shown in Figure 1. It consists of ~10,000 grid blocks using a refined Voronoi grid. The main reservoir consists of two rock types with different porosities – 5% in the western part of the model, 10% in the eastern part. The reservoir is capped by a low permeability clay cap with a small leak area. The top of the reservoir is an atmospheric boundary condition (fixed pressure and temperature). At the bottom of the reservoir a hot upflow with an enthalpy of 1200kJ/kg provides fluid flow into the system. Two wells, one in each compartment of the reservoir, extract a fixed amount of fluid during a 50 year production period. The model has a small two-phase region at steady state conditions which grows in size during the production period.

For the inverse modelling example, we selected the two porosities and the upflow enthalpy as “unknown” parameters. The forward model runs the model to a steady state solution (10^{15} s), then starts the 50 year production period.

As observations we chose the temperature in the reservoir along the two wells at steady state time, i.e. before production commences. As a third observation set, we selected microgravity, this being a complex observation type which is not included in output from other simulators (TOUGH2, TETRAD) but is implemented as a standard feature in Volsung for arbitrary polyhedron cells using analytical geometric gravity factors [Singh and Guptasarma, 2001]. We prepared two data sets to test the inverse modelling process: One set consisted of the ideal, i.e. exact, model results from the forward model run with the correct parameters. For the second set we added random noise to the observations using a normal distribution with 1°C standard deviation for the temperature profiles and 1μGal for the microgravity observations. The weights for the observations were set using these standard deviations as “measurement errors” and applying the correction for the sampling frequency.

The PEST inverse modelling process was started by setting both porosities to 3% and the upflow enthalpy to 1100kJ/kg. We used 4 agents on a consumer grade PC with an AMD Ryzen 9-3900X CPU. In the first instance we used observation group weights of $\alpha=1.0$ for all groups; we noted that while this resulted in very good matches for the temperature profiles the match for the microgravity was less convincing. As discussed above we hence increased the α weight for the microgravity first to 2.0, and then to 5.0 which improved the match significantly. Figures 2 and 3 show the temperature and microgravity matches for the “noisy” data set for the 1.0 and 5.0 weight factors, respectively. Figure 4 shows the improvement in micro gravity found by PEST for a single representative station. Figure 5 shows the improvement in temperature match for the well in the eastern compartment; the match in the western compartment is similar and not shown here. Table 1 summarizes our findings; it is evident that for the 5.0 weight factor the inverse modelling process found the correct parameters for both the ideal and the noisy observation data sets.

| | Ideal Observations | | | Noisy Observations | | |
|-------------------------------|--------------------|--------------|--------------|--------------------|--------------|--------------|
| | $\alpha=1.0$ | $\alpha=2.0$ | $\alpha=5.0$ | $\alpha=1.0$ | $\alpha=2.0$ | $\alpha=5.0$ |
| Obj.function reduction | 4.6E-03 | 4.8E-03 | 3.8E-07 | 6.0E-03 | 1.7E-02 | 8.6E-04 |
| Iterations | 15 | 12 | 19 | 14 | 11 | 13 |
| Model Calls | 112 | 86 | 138 | 100 | 76 | 97 |
| Time [s] | 7606.5 | 6295 | 7419.7 | 7947.3 | 6988 | 4665.5 |
| upenthalpy [kJ/kg] | 1201.7 | 1203.7 | 1200.1 | 1201.9 | 1195.6 | 1200.1 |
| poreast [%] | 6.50 | 9.67 | 9.97 | 7.53 | 2.71 | 10.00 |
| porwest [%] | 2.10 | 2.00 | 4.96 | 2.07 | 19.83 | 4.99 |

Table 1: Summary of the inverse modelling results for the reservoir model. The inverse modelling managed to establish the correct upflow enthalpy for all runs; however, for the algorithm to correctly match the porosity the weight of the microgravity observation group had to be increased to 5.0.

⁴ Other service providers, e.g. Google Cloud or Azure, are possible but not implemented yet.

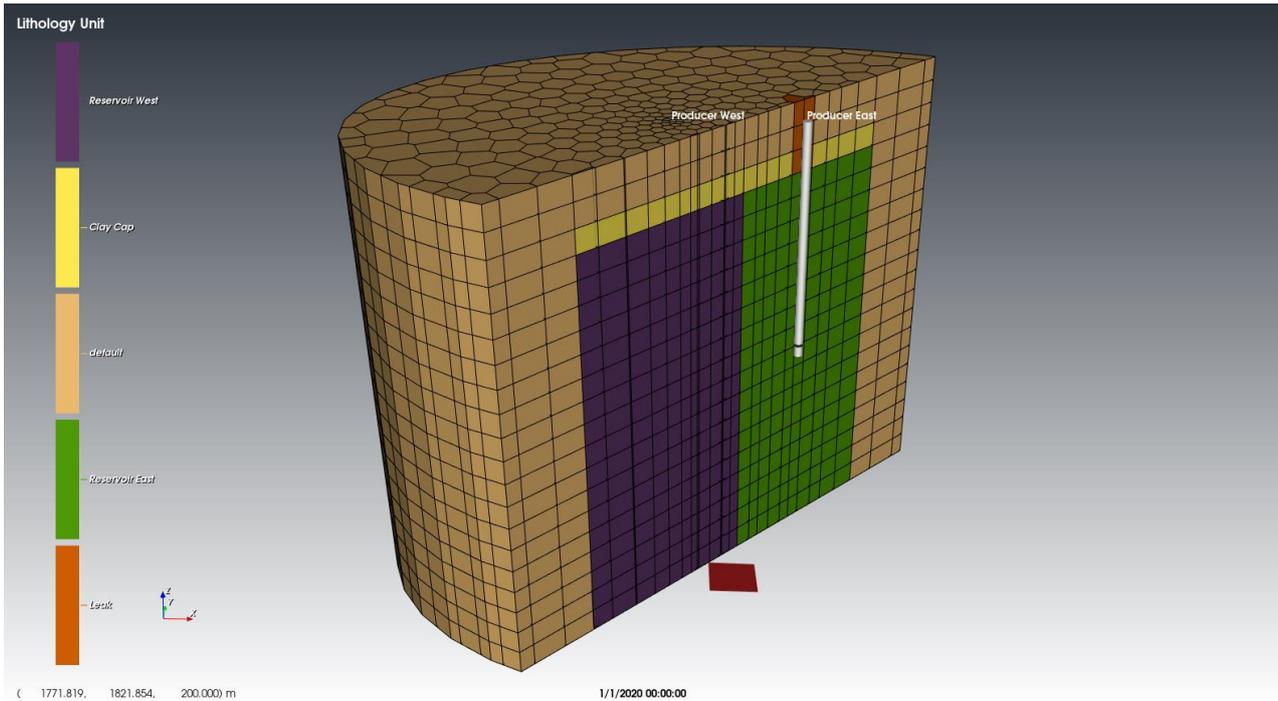


Figure 1: The example reservoir model used for the inverse modelling demonstration. Model grid blocks are coloured by rock type. Porosities in *Reservoir West* and *Reservoir East* units and the upflow (red square at bottom) enthalpy were automatically calibrated to match temperature profiles and microgravity data.

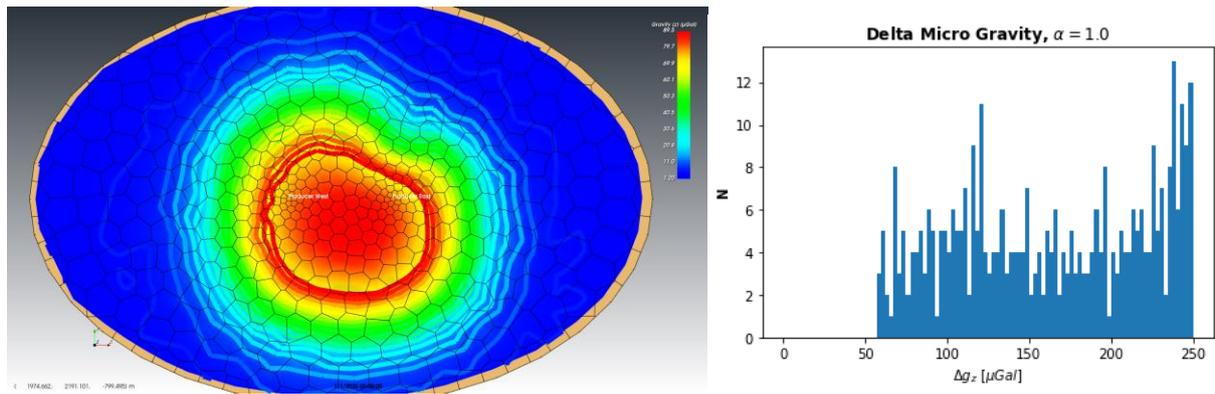


Figure 2: (Left) The match between model and calibration microgravity data for noisy data, $\alpha = 1$. Model microgravity data is shown as the coloured surface over the model and the calibration is shown as coloured contours. Both datasets have the same colour scale so we can tell from this image that the shape of the modelled microgravity change is consistent with the calibration data but the match to magnitude of gravity change is poor. (Right) Histogram showing the difference between the true micro gravity data and the solution found by PEST, $\alpha = 1$.

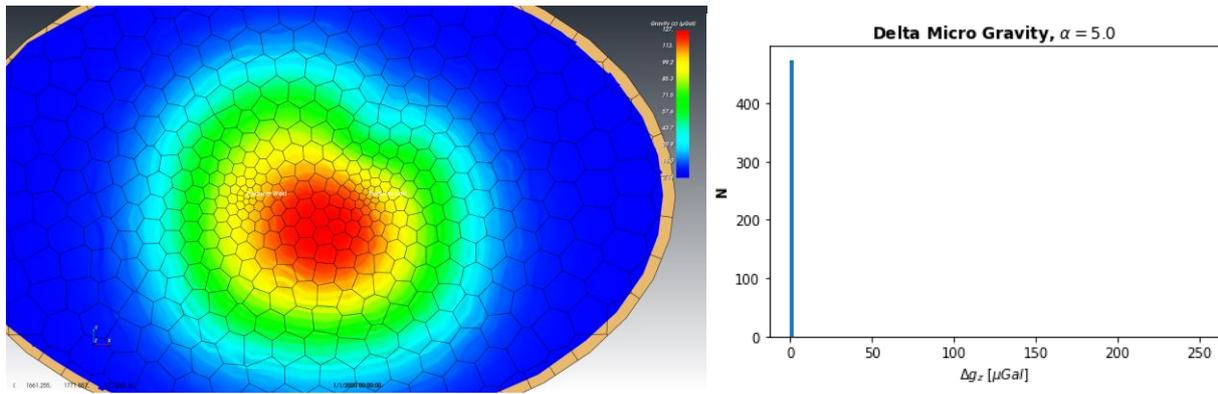


Figure 3: (Left) The match between model and calibration microgravity data for noisy data, $\alpha = 5$. Model microgravity data is shown as the coloured surface over the model and the calibration is shown as coloured contours. Both datasets have the same colour scale; due to the excellent parameter match both surface and contours blend into each other and become nearly indistinguishable. **(Right)** Histogram showing the difference between the true micro gravity data and the solution found by PEST, $\alpha = 5$, using the same scale for the abscissa as used in Figure 2.

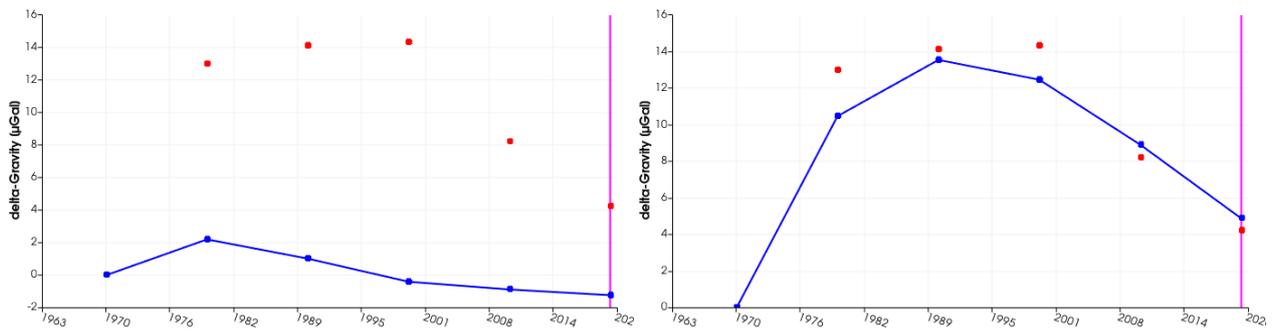


Figure 4: Match between model and noisy calibration microgravity data for a typical survey station. The left chart shows the modelled data (blue) for the initial guess. The right hand side chart shows the final match with the $\alpha = 5$ weight factor for the microgravity observation group.

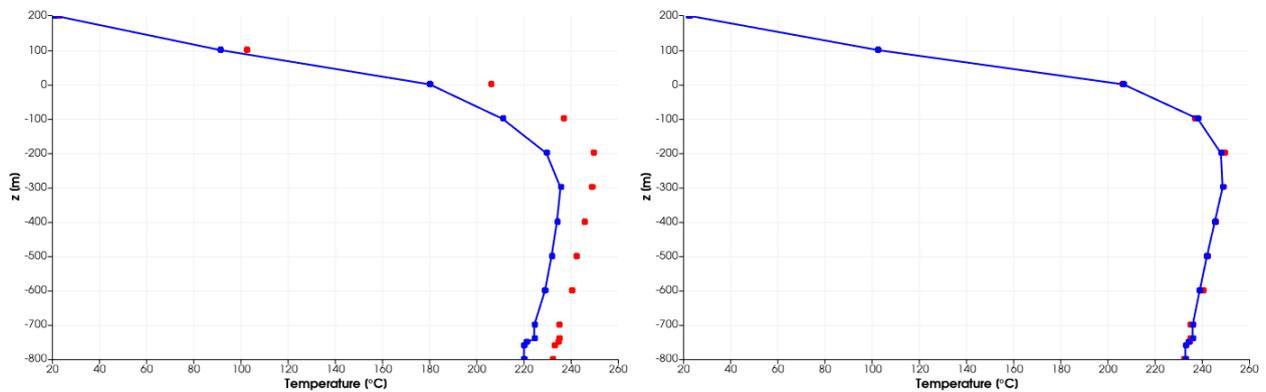


Figure 5: Match between model and noisy calibration temperature data for the well in the eastern reservoir compartment. The left chart shows the modelled data (blue) for the initial guess. The right hand side chart shows the final match with the $\alpha = 5$ weight factor for the microgravity observation group.

4.2. Wellbore Model – Inverse Modelling of a Flowing Pressure/Temperature Profile

Wellbore models of flowing geothermal wells are often calibrated using pressure-temperature-spinner (PTS) surveys. In single phase situations many of the unknown model parameters can be easily established from the pressure, temperature and velocity signal, where the latter is generated from the spinner frequency using cross-plot techniques.

However, it can be very hard to establish fluid properties in the two-phase regions of the well. One aspect is that the velocity/mass rate signal generated from the cross-plots becomes very unreliable. The other difficulty is that the relationship between the fluid temperature and its enthalpy becomes complex; it now depends on pressure, specific enthalpy and non-condensable gas (typically CO₂) content of the fluid. Since the temperature is an observed quantity we need to find representative values for the unknown enthalpy and CO₂ component mass fraction. The situation becomes more complex since the enthalpy of the fluid changes with elevation (potential energy term) in the wellbore, exchange of heat with the surrounding formation, and the acceleration of the fluid (kinetic energy term). Hence the geothermal reservoir engineer rightfully struggles with the analysis of this commonly occurring situation.

As our second example we investigated this situation in a two-phase wellbore model. The wellbore model for this investigation had a single feedzone at a known location with a known mass rate at the time of the flowing PTS survey. This situation is not uncommon during well testing when the production mass rate is measured at surface. For this inverse modelling task the mass rate was held constant and the unknown parameters used for calibration were: fluid enthalpy at the feedzone location, the mass fraction of CO₂, and the rugosities of the production liner and cased sections of the well. Figure 6 depicts the wellbore model.

We first ran the forward wellbore model with known parameters of $h = 1150\text{kJ/kg}$, $X_{\text{CO}_2} = 0.3\%$, rugosity $r_{\text{lower}} = 0.4\text{mm}$ in the lower part of the well and $r_{\text{upper}} = 0.1\text{mm}$ in the upper part. We then used this model output for the pressure and temperature profile as two observation datasets in the inverse modelling process.

Figure 7 shows the comparison between the initial guess (blue) and the correct solution (red). PEST ran this model using 5 agents in 58 seconds using 15 iterations and 117 model evaluations. The final parameters found are $h = 1149.97\text{kJ/kg}$, $X_{\text{CO}_2} = 0.2993\%$, $r_{\text{lower}} = 0.3993\text{mm}$ and $r_{\text{upper}} = 0.0997\text{mm}$. These values are basically indistinguishable from the correct solution of the system.

We note here that while the correct mass rate in the wellbore model is required to determine the rugosities, the specific enthalpy and CO₂ component mass fraction can be uniquely identified when calibrating to pressure and temperature data alone. In other words, while the pressure and temperature profile could be perfectly matched with different rugosity and mass rate combinations, the enthalpy and CO₂ mass fraction are unique since the fluid temperature is a unique function of pressure, enthalpy and component mass fraction. Hence this method of inverse modelling to obtain these parameters will prove to be a valuable tool in a geothermal reservoir engineer's toolbox.

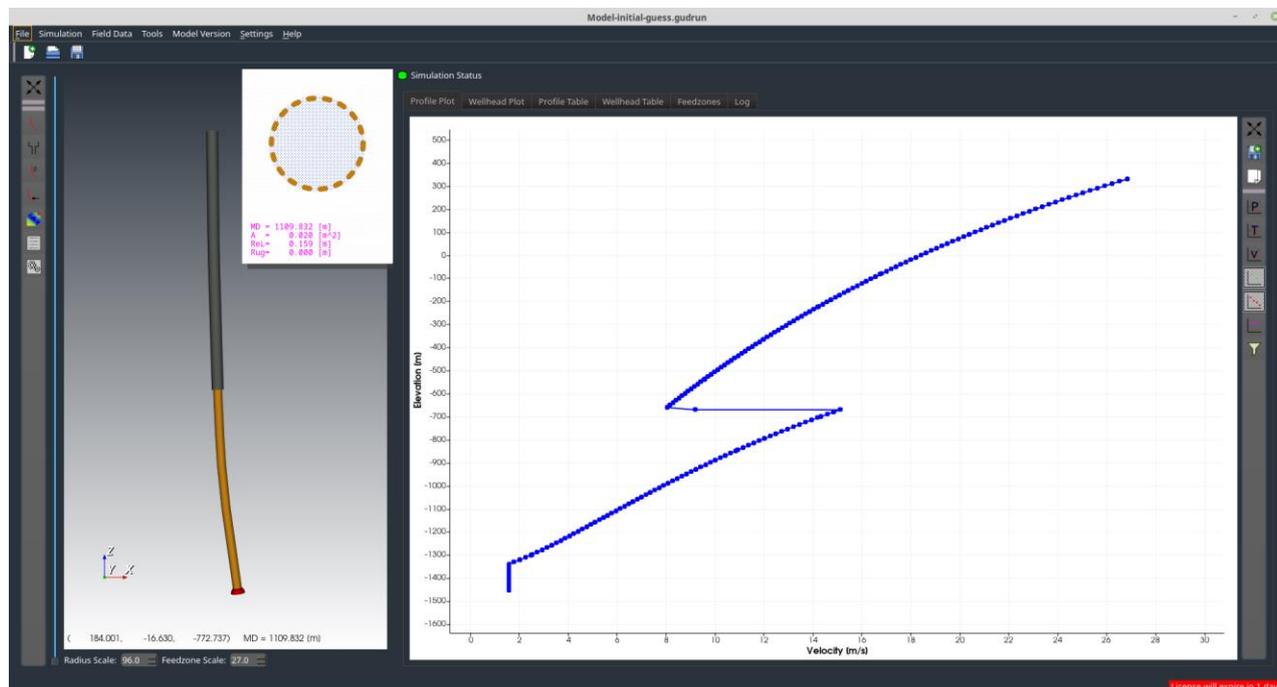


Figure 6: Screenshot of the wellbore model. The left hand view shows the casing/liner section and the feedzone at the bottom of the well. The right hand side shows the mixture velocity with the initial guess parameters; note the “kink” at -650mRL shows the transition of the fluid velocity from the small diameter slotted liner to the large diameter cased section.

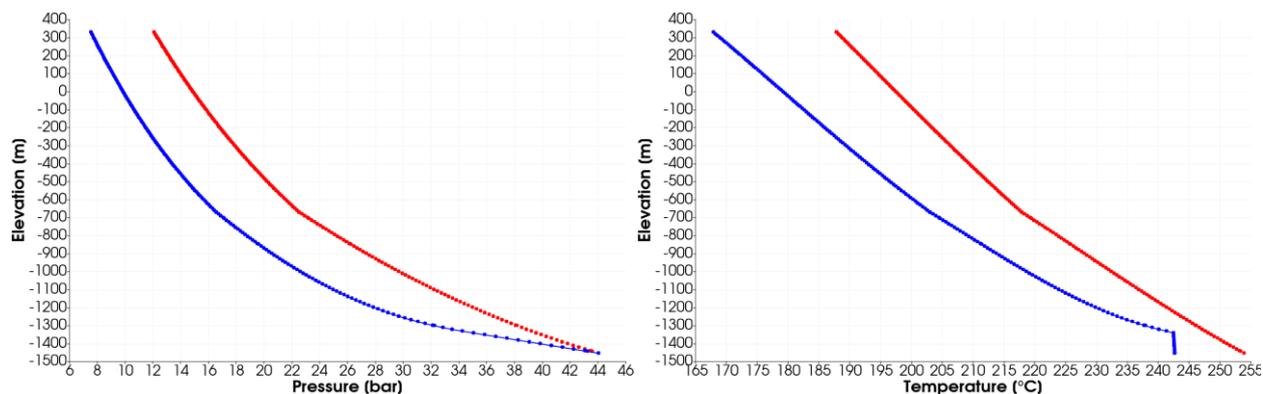


Figure 7: Pressure (left) and temperature (right) in the wellbore; the blue plot denotes the initial guess and red the calibration data. In the final match model and calibration data coincide perfectly, hence we do not show them here again.

5. DISCUSSION AND OUTLOOK

This work presents the coupling of the reservoir and wellbore simulators of the Volsung package to the inverse modelling and uncertainty analysis tool PEST. The coupling is universal and allows access to all input parameter and output observations of geothermal simulations. This opens up the field of autonomous optimization to modelers using Volsung.

In our opinion inverse modelling can be a very useful tool for shortening the development and calibration of numerical models. However, it should not be viewed as a “silver bullet” for calibrating models; it can assist the modeler but not replace her.

Of particular interest is also the infrastructure developed in Volsung to facilitate the running of BEOPEST. Volsung now contains a fully automated system for launching forward models in the cloud. By simply replacing BEOPEST as the run manager it will in short time be possible to use this system for other tasks like running model batches with pre-defined parameters, Monte-Carlo simulations, or any other perceivable algorithm for optimization and uncertainty analysis tasks.

REFERENCES

- Cinco, F., Franz, P., & Menzies, A.: Testing the Volsung Suite as a Reservoir Simulation software by comparing it with the TOUGH2 Tiwi Model. *NGAP Webinar Series: Digital Applications*. (2020).
- Clearwater, J., & Franz, P.: Introducing the Volsung Geothermal Simulator: Features and Applications. *Proceedings 41st New Zealand Geothermal Workshop*. (2019).
- Doherty, J., Brebber, L., & Whyte, P.: PEST: Model-Independent Parameter Estimation. *Watermark Computing*, Brisbane. (1994).
- Franz, P., Clearwater, J., & Burnell, J.: Introducing the Volsung Geothermal Simulator: Benchmarking and Performance. *Proceedings 41st New Zealand Geothermal Workshop*. (2019).
- Levenberg, K.: A Method for the Solution of Certain Non-Linear Problems in Least Squares. *The Quarterly of Applied Mathematics*, 2: 164-168. (1944).
- Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics* 11(2):431-441. (1963).
- McLean, K., Franz, P., & Clearwater, J.: Swanhild: Numerical Pressure Transient Analysis using the Volsung Geothermal Reservoir Simulation Package. *Proceedings 42nd New Zealand Geothermal Workshop*. (2020).
- Moon, H., Clearwater, J., Franz, P., Wallis, I., & Azwar, L.: Sensitivity Analysis, Parameter Estimation and Uncertainty Propagation in a Numerical Model of the Ngatamariki Geothermal Field, New Zealand. *Proceedings of 39th Stanford Geothermal Workshop*. (2014).
- Pearson, S. C. P., Franz, P., & Clearwater, J.: Refining Geothermal Reservoir Models Using Repeat Microgravity Data. *Proceedings World Geothermal Congress 2015*. (2015).
- Quinao, J. J., Franz, P., & Clearwater, J.: Well performance diagnostics and forecasting using the Gudrun wellbore simulator-case studies from Kawerau, New Zealand. *Proceedings 42nd New Zealand Geothermal Workshop*. (2020).
- Singh, B., & Guptasarma, D.: New method for fast computation of gravity and magnetic anomalies from arbitrary polyhedra. *Geophysics*, 66 (2), 521–526. (2001).