PROCEEDINGS, 44th Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, February 11-13, 2019
SGP-TR-214

Working with Multi-million Block Geothermal Reservoir Models

John O'Sullivan, Adrian Croucher, Joris Popineau, Angus Yeh and Mike O'Sullivan

Department of Engineering Science, University of Auckland, Private Bag 92019, Auckland, New Zealand

jp.osullivan@auckland.ac.nz

Keywords: geothermal reservoir simulation, Waiwera

ABSTRACT

Waiwera is an open-source, parallelized geothermal reservoir flow simulator which gives the ability to reliably run and calibrate complex geothermal reservoir models made up of millions of blocks. However, running such large models brings a number of new challenges that must be addressed in order to realise the benefits of the increased computational power.

This paper describes these new challenges and our team's approaches for handling them. The challenges can be broadly categorised as either model setup or model post-processing. There are a small number of issues relating to simulation control that are also discussed. For model setup, our techniques for the following are discussed:

- Grid definition and generation
- Distributing rock properties to the model
- Representing faults and fractures
- Defining wells and their feedzones
- Specifying boundary conditions including deep inputs
- Upscaling/downscaling between different grid resolutions

For model post-processing we make extensive use of python scripting tools and graphical user interfaces such as Leapfrog. We describe our techniques for extracting information from very large transient simulation output files and then presenting results in a range of meaningful formats to provide stakeholders with tools to support their decisions.

Both model setup and model post-processing are demonstrated using examples from current real-world geothermal reservoir modelling projects.

1. INTRODUCTION

Reservoir modelling has become an integral part of managing geothermal resources. It provides invaluable information to assist with the decision-making process and as a result there is a requirement to run increasingly complex reservoir models. Partly because increasingly complex questions are being asked of the models require high resolution models (Burnell et. al., 2015). And partly because geoscientific tools for characterising geothermal systems continue to improve and so conceptual models are becoming more detailed and refined. Creating reservoir models that are closely coupled to the conceptual models developed from the geoscientific information therefore requires building more complex, higher resolution reservoir models. An important benefit of creating closely coupled models is that it enables experts from a range of disciplines to interact and communicate using a single source of information that includes all available field data and reservoir modelling results (Popineau et. al., 2018). This in turn enables experts to communicate more effectively and consistently with other stakeholders such as management and community groups.

Using our new geothermal simulator Waiwera we now have the capability to run very large models efficiently (Croucher et. al., 2018, O'Sullivan et. al., 2017). However, for full-scale geothermal reservoir models our experience has shown that as the resolution increases the model run times increase more rapidly than proportionately to the increase in model size. A typical example is shown in the Table 1 below for three different resolution dual porosity models of a short period of production history for a geothermal system. The models cover the same domain, were set up using the same conceptual model and have the same layer structure.

Table 1: Comparison of performance of different resolution models.

Model	Horizontal model resolution of blocks in production zone	Number of blocks in the model	Maximum time step size achieved	Number of time steps required	Number of CPUs	Run time	CPU hours
Coarse	200 m x 200 m	140,000	2.1×10^5 s	115	80	109 s	2.4 hr
Medium	100 m x 100 m	550,000	9.5×10^4 s	238	160	726 s	32.3 hr
Fine	50 m x 50 m	2,300,000	4.5×10^4 s	479	320	5056 s	449.4 hr

The results show that as the resolution increases the maximum time step achieved decreases. This means that simulations take longer because not only there are more blocks but also more time steps are required to achieve the same total time. This is not surprising because as the model resolution increases and model block volumes become smaller, the thermodynamic conditions in each block can change more rapidly hence requiring a higher resolution in time to solve each time step. For this example, the number blocks increases by a factor of 4-5 with increasing spatial resolution and the number of time steps required for the simulation increases by a factor of 2. Combining these should result in overall increases in computational effort of approximately 10. However, the table shows that the actual increases in computational effort were a factor of 13-14. Previous benchmarking has shown that Waiwera scales well with increasing number of CPUs for a given model (Croucher et. al., 2018, O'Sullivan et. al., 2017) so the discrepancy cannot be attributed to the overhead associated with more processors. Instead it is due to the fact that thermodynamic conditions in smaller block volumes not only require to shorter time steps solve their non-linear system of equations but also that those non-linear systems of equations are more complex leading more poorly conditioned linear systems of equations that must be solved. This has an additional computational cost at every time step and pushes the factors in this example from approximately 10 up to 13-14.

The consequence of these findings is that there is a significant computational cost to increasing the resolution of full-scale geothermal reservoir models. Using Waiwera as the simulator that computational cost can be offset by using more CPUs but often the computational resource is not unlimited and it also has a financial cost associated with it. Our solution to this problem has been to use multiple models with different resolution for a particular system with the model resolution selected based on the model forecasts required. In addition, when the highest resolution models are required, we have used lower resolution models to help speed up the model calibration process and reduce the computational cost.

When using high-resolution models of full-scale geothermal systems, the large number of blocks means that different approaches are required when setting up and post-processing the models. The approaches we use are described in the following sections. Using multiple models with different resolutions for the same system means that the model setup is more complicated as the approach needs to ensure the models generated are consistent. The implications of using a number of multiple models for each of model setup sections are also discussed. Note that the approaches presented apply only to regular rectangular grids because their simple structure allows relatively simple methods to be used for consistently generating multiple models. The high resolution models also overcome many of the shortcomings of regular rectangular models.

2. MODEL SETUP

2.1 Grid definition and generation

There are many tools available for generating the grids used in geothermal reservoir models (Croucher et. al., 2018, Popineau et. al., 2018, Yeh et. al., 2013) and most should handle increasing the number of models blocks into the millions without any trouble. We most commonly use either pyTOUGH scripts (Wellmann et. al., 2012) or Leapfrog Geothermal both of which handle large grids easily and can be used to apply the topography surface to the tops of the model. However, when using multiple models for a system we have found that the most consistent model surfaces can be created by generating a very high resolution grid first and mapping the topography on to it. Each of the lower resolution models can then calculate their surface topography by downscaling the topography from the very high resolution model through an averaging process. The averaging process is straightforward for regular grids but can also be applied to irregular grids. By its nature this approach does have a smoothing effect when downscaling to lower resolution models but we have found it to more consistent pressure distributions from one resolution model to the next. Examples of the grid surfaces generated are shown in Figure 1.

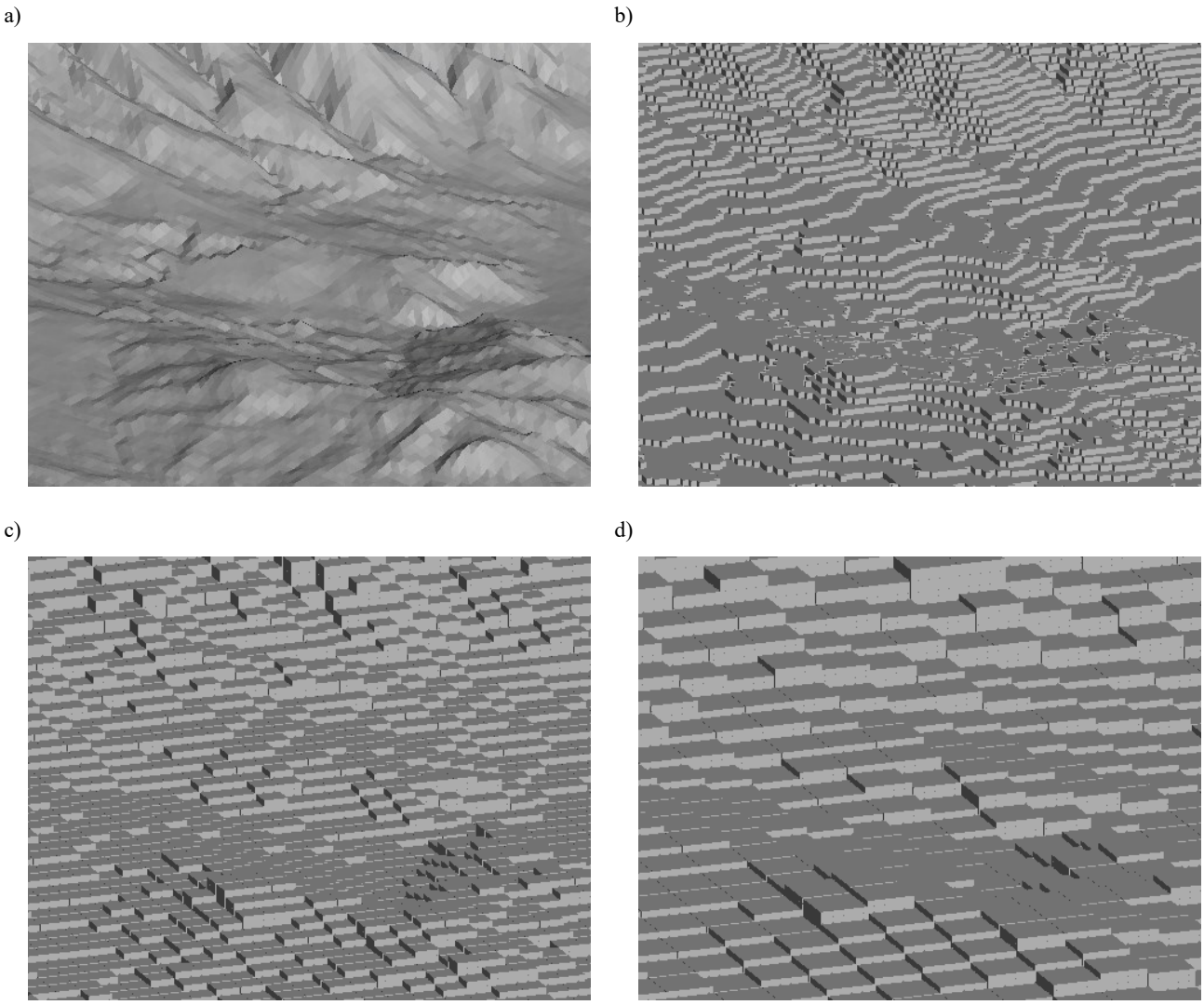

Figure 1: Comparison of surface topography represented by (a) a high resolution surface mesh, (b) a 12.5m x 12.5m resolution block model, (c) a 50m x 50m resolution block model and (d) a 100m x 100m block model.

2.2 Distributing rock properties to the model

There are also several options for distributing the rock properties to a reservoir model grid depending on the simulator that is to be used. For TOUGH2 type simulations rock types groupings is a common approach (Preuss et. al., 1999) though geostatistical approaches have been used in some studies (Omagbon et. al., 2016). In the past, models that used the rock type approach distributed the rock types to the grid by either manually editing model input files directly or using a graphical user interface to manually assign rock types visually. More recently scripting tools have been used to assign rock types is specified zones automatically (O'Sullivan et. al., 2013). These approaches can all be applied to high resolution reservoir models in general and Waiwera models in particular but they can be prohibitively time consuming. Even using scripting can be very challenging for distributing complex geological models to a high-resolution grid. We have found Leapfrog Geothermal to be a very effective tool for overcoming these challenges (Popineau, et. al., 2018) and it has the added benefit of enabling the automatic redistribution of rock types to the reservoir model if the geological model is updated. When multiple models are being generated we again select a very high resolution model and then use Leapfrog to distribute the rock types. We then downscale the rock types onto our different resolution models using statistics from the rock type distribution on the very fine model. An example of the resulting model geologies is shown in Figure 2.

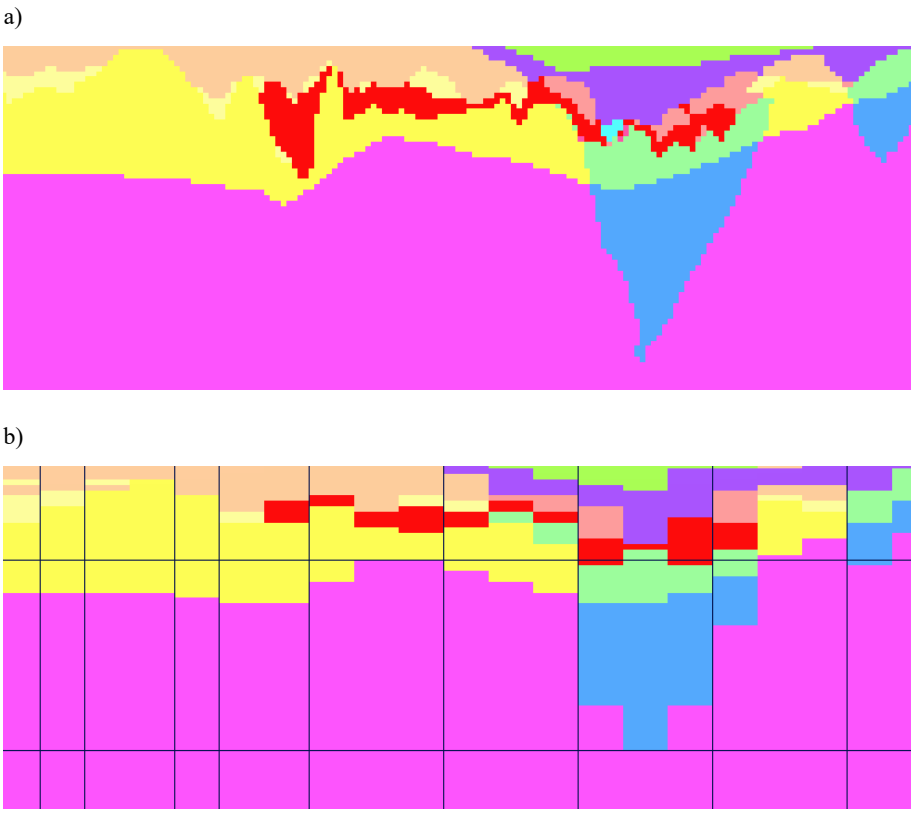

Figure 2: Comparison of model geologies on (a) a very high resolution grid mapped using Leapfrog Geothermal and (b) downscaled onto a lower resolution grid.

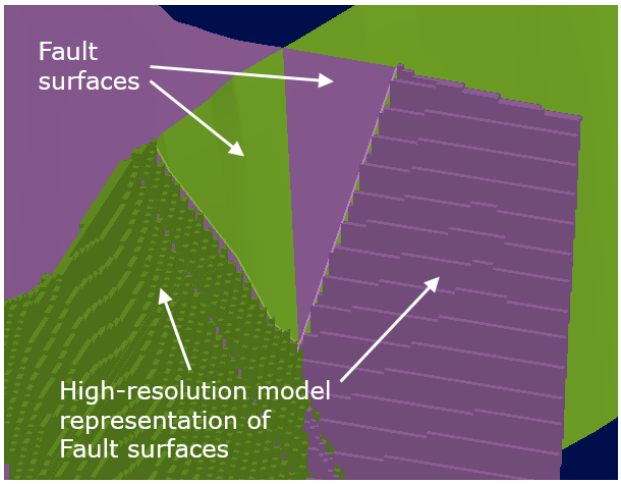

Figure 3: Comparison of faults represented using surfaces and once they have been mapped onto a high resolution model grid.

2.3 Representing faults and fractures

Our models now include explicit representation of the faults and fractures as standard practice (Ratouis et. al., 2017a, Ratouis et. al., 2017b, O'Sullivan et. al., 2013, O'Sullivan et. al., 2016, O'Sullivan et. al., 2018). We often use Leapfrog geothermal for assigning the explicitly represented faults and fractures to the blocks in our reservoir models. The tools in Leapfrog geothermal make this process quite simple and again make it easy to transfer any updates in the structural geological model across to the reservoir model. Leapfrog geothermal also allows dipping faults to be represented and has options that can be used to ensure that the fault is represented continuously along the

strike and up the dip of the fault. When multiple models are being generated for the same system, we do not use Leapfrog to apply explicit structures in each model. We use Leapfrog to apply the structures explicitly to a very high-resolution model and then use our own method for specifying the fault block permeabilities and porosities in such a way that they can be downscaled onto any lower resolution model resulting in a consistent permeability distribution. The method can be explained most easily by considering the examples in Figures 4 and 5.

2.3.1 Case 1: A single fault intersects the block (Figure 4)

The very-high resolution model created by Leapfrog is shown in Figure 4a. The permeability and porosity of the both the faulted and unfaulted rock type are defined on this grid. Note that both can have anisotropic permeabilities. Figure 4b shows an example of a downscaled model grid with four times lower resolution blocks. The faulted blocks in this model are much larger and from the plot it can be seen that the fault defined on the high-resolution grid actually occupies only approximately one quarter of the larger blocks. The permeabilities of the larger fault blocks is calculated by considering the properties of the fault across-strike, along-strike and up-dip:

- Along-strike: Approximate area weighted average of permeabilties aligned with the fault strike
- Across-strike: Minimum of the across-strike permeabilities
- Up-dip: Approximate area weighted average of permeabilties aligned with the fault dip

The porosity is calculated using the approximate weighted average of component porosities. This approach assumes that structures can act as pathways along the strike and up the dip of the fault and can act as barriers across the faults strike. Essentially it takes into account that the faulted blocks in the lower resolution models are made up in a large part by unfaulted material. Note that in all three directions it is possible to assign properties that make the fault hydraulically inactive (ie permeabilities equal to the unfaulted rock type). For the grid resolution in Figure 4b the calculations would be as follows:

$$k_{FM} = \begin{bmatrix} \min(k_{UFX}, k_{FX}) \\ {}^{3}/_{4}\, k_{UFY} + {}^{1}/_{4}\, k_{FY} \\ {}^{3}/_{4}\, k_{UFZ} + {}^{1}/_{4}\, k_{FZ} \end{bmatrix}$$

$$\phi_{FM} = {}^{3}/_{4}\, \phi_{UF} + {}^{1}/_{4}\, \phi_{F}$$

Figure 4c shows an example of a downscaled model grid with eight times lower resolution blocks resulting in the following calculations:

$$k_{FC} = \begin{bmatrix} \min(k_{UFX}, k_{FX}) \\ {}^{7}/_{8}\, k_{UFY} + {}^{1}/_{8}\, k_{FY} \\ {}^{7}/_{8}\, k_{UFZ} + {}^{1}/_{8}\, k_{FZ} \end{bmatrix}$$

$$\phi_{FC} = {}^{7}/_{8}\, \phi_{UF} + {}^{1}/_{8}\, \phi_{F}$$

Using the approximate weighted average is equivalent to assuming that the fault is perfectly aligned with the block and has zero dip. This assumption is more accurate with higher resolution grids but in general it is not the case. However, making this assumption greatly simplifies the calculation of the downscaled permeabilities and allows it to be applied generally to regular rectangular grids. The results show in Section X that the permeability distributions that are generated using this approach are equivalent to a large degree.

a) b) c)

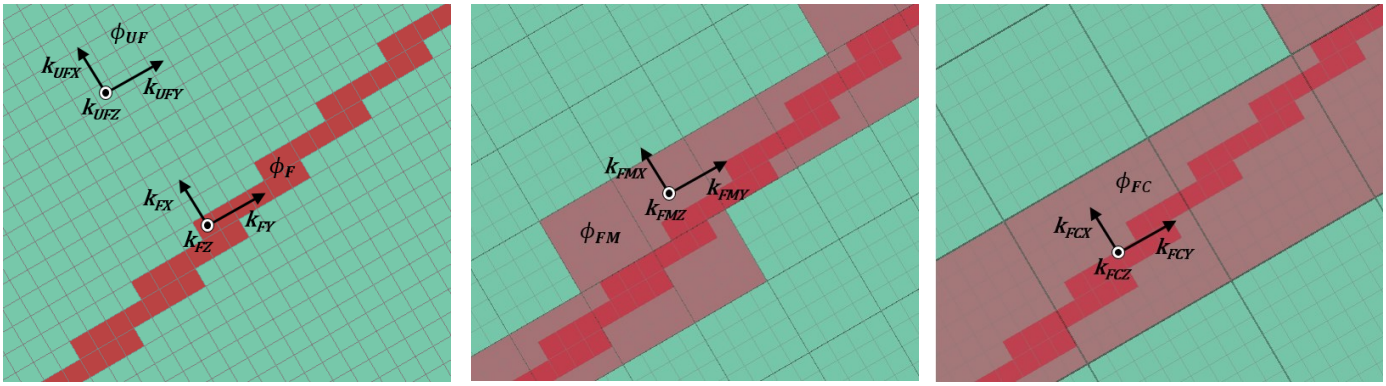

Figure 4: Representing a fault explicitly on three different resolution grids. The fault passes through the red blocks in each and the green blocks are not faulted.

2.3.2 Case 2: Two or more faults intersects the block (Figure 5)

The very-high resolution model created by Leapfrog is shown in Figure 5a with two faults intersecting. Figures 5b and 5c shows the model structure for the two downscaled model grids with four and eight times lower resolution blocks respectively. In this case the permeabilities in the lower resolution models is more complicated to calculate because the faults could both be acting as either barriers or pathways to flow at their intersection. Also, the concepts of along-strike and across-strike become unclear when multiple faults are present. Instead w return to describing the permeability directions in terms of the xyz coordinate system. Our approach is to calculate their properties as follows:

- Horizontal permeabilities: If the permeability of the high-resolution fault intersection block is greater than unfaulted permeability then one of the faults is acting as a pathway in this direction and an approximate area weighted average of permeabilties is used. If the permeability of the high-resolution fault intersection block is less than the unfaulted permeability then one of the faults is acting as a barrier to flow in this direction and the minimum of the permeabilities is used.
- Up-dip: Approximate area weighted average of the vertical permeabilties

The porosity is still calculated using the approximate weighted average of component porosities.

a) b) c)

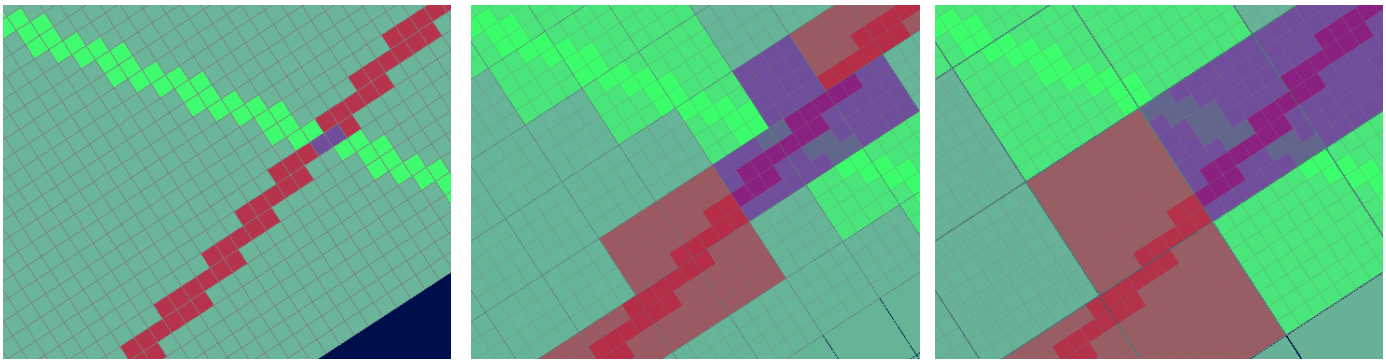

Figure 5: Representing two intersecting faults explicitly on three different resolution grids. One fault passes through the red blocks, the other passed through the bright green blocks and the intersection of the faults is indicated in purple. The darker green blocks are not faulted.

2.3.3 Permeability Directions

There are many advantages to using regular rectangular grids in terms of model setup, model post-processing and grid orthogonality. Nevertheless, a consequence of using regular rectangular grids is that most structures will not align with the grid and they will need to be represented using a "staircase" approach. To ensure that structure behave correctly on all resolution grids the face permeability directions must be adjust along the structure so that fluid can flow through the staircase. Using unstructured grids (such as Voronoi) can alleviate this issue in the horizontal plane as the grid can be mapped to the structure. However, unless fully 3D unstructured grids are used, stair casing must still be applied in the vertical direction for dipping structures. The plot in Figure 6 shows how the permeability directions are modified for a staircased fault.

2.4 Defining wells and their feedzones

Well tracks and their feedzones must be defined in geothermal reservoir models so that source or sink terms can be created to represent fluid flowing into or out of the wells. In the past, the lower resolution of the reservoir models meant that often feedzones were contained entirely within a single model block. This make the process of assigning and handling the wells very simple. However, in multi-million block models each feedzone of a well may span a number of model blocks. We use a system where well information such as the well track, feedzones and recorded field data are stored in json format data files. Each well has its own data file and the files are completely grid dependent. When setting up a model our python scripts read these data files and then sample down the well track in the specified model grid. The sampling approach not only identifies which blocks in the model the well track passes through but it also calculates the proportion of each feedzone that is in each model block.

Figure 7 shows the same well specified on two different resolution grids. In Figure 7a the lower resolution grid means that a total of ten blocks are intersected by the well's feedzones. In Figure 7b the model has double the vertical and double the horizontal resolution. In this case 22 different blocks intersect the same well's feedzones. In some cases the feedzone only intersects a small proportion of the block demonstrating the usefulness of the sampling information. Using the sampling information the flow into or out of the well can be divided up within each feedzone proportionately to amount of feedzone that passes through each block.

This approach not only provides an efficient method for determining feedzone blocks and their flow rates but also it provides the mechanism for generating consistent models of different resolutions using the grid independent well data files.

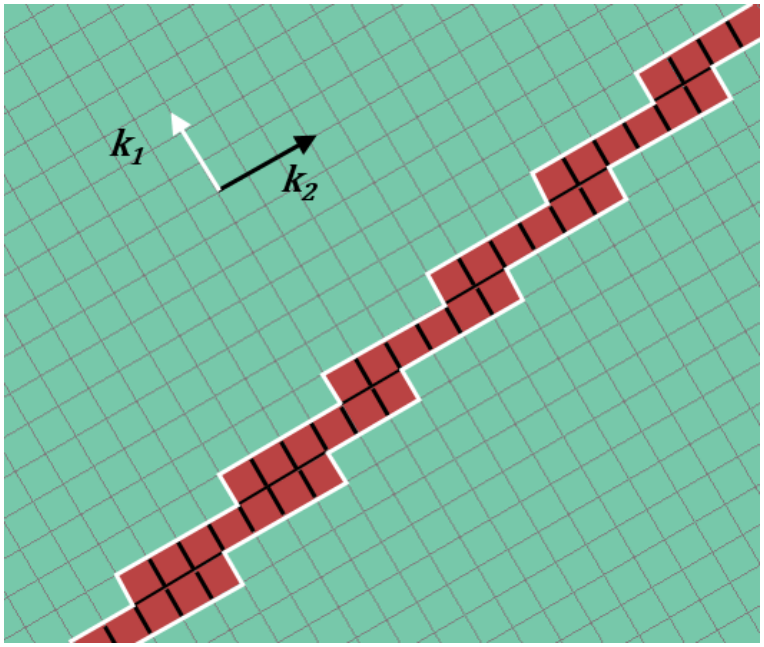

Figure 6: A fault represented in a high-resolution model. The fault passes through the red blocks and the green bloack are unfaulted. White and black lines indicate k_1 and k_2 permeabilities respectively.

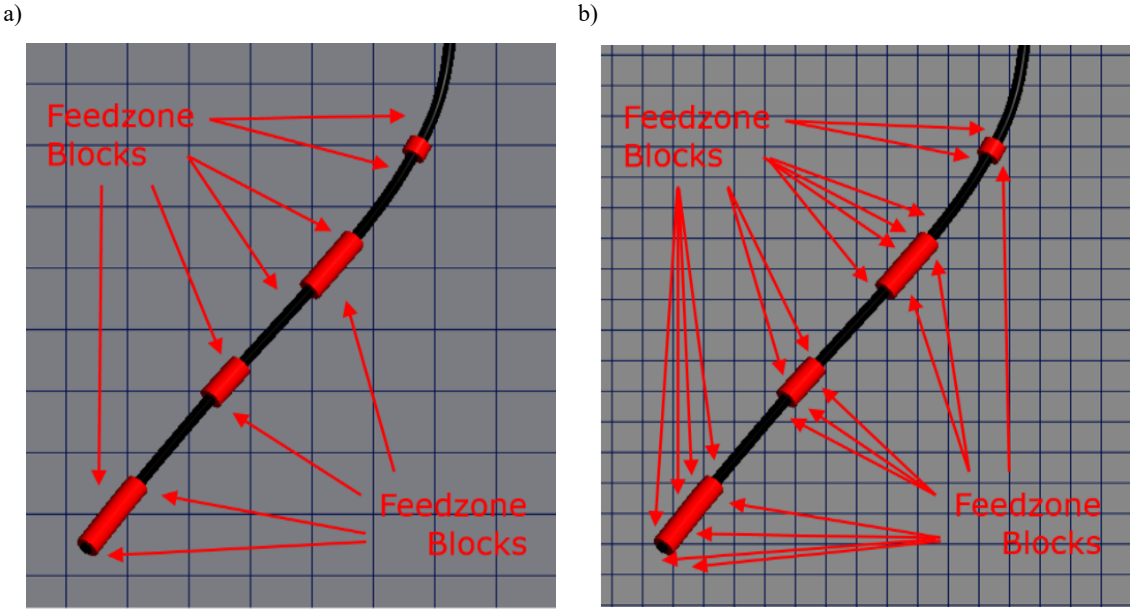

Figure 7: A well with its feedzones indicated mapped onto two different resolution grids. The feedzone blocks for each grid are indicated.

2.5 Specifying boundary conditions including deep inputs

When defining the boundary conditions for multi-million block models it is not practical to manually edit and adjust mode files. We use a set of python scripts that assign the boundary conditions in a grid-independent manner by specifying physical locations rather than model blocks. The locations can either be specified in zones or they can be specified by physical points or lines in space. Zones are used commonly for specifying the distributed heatflow at the base of a model or the boundary of bodies of water on its surface. Once the coordiates of the zone are specified it is simple to identify the boundary blocks and apply the distributed boundary condition to the model, regardless of the model resolution. A common example of using lines or points to specify boundary conditions is applying deep upflow at the bottom of a model. Rather than specify particular blocks where un upflow flux is to be applied, we use a method where we specify sections along the trace of upflow faults and the total mass that should be applied on that section. Clearly these specifications are grid independent and can be thought of as intrinsic properties of the geothermal system rather than a property of a particular model. Our scripts read the upflow specifications when a model is being set up and then apply the appropriate proportion of upflow to each model block that is intersected by the fault trace in question at the bottom of the model. Examples of the resulting mass flow and enthalpy distribution for the bottom boundary condition of a multi-million block model are shown in Figure 8. Note that in the figures the black lines show all the fault traces present at the bottom of the model and the red lines correspond to the sections of fault where upflow is applied.

Once again, by developing a grid-independent approach we not only have methods for consistently generating models regardless of how large they become but we also have an efficient method for generating multiple models of the same system.

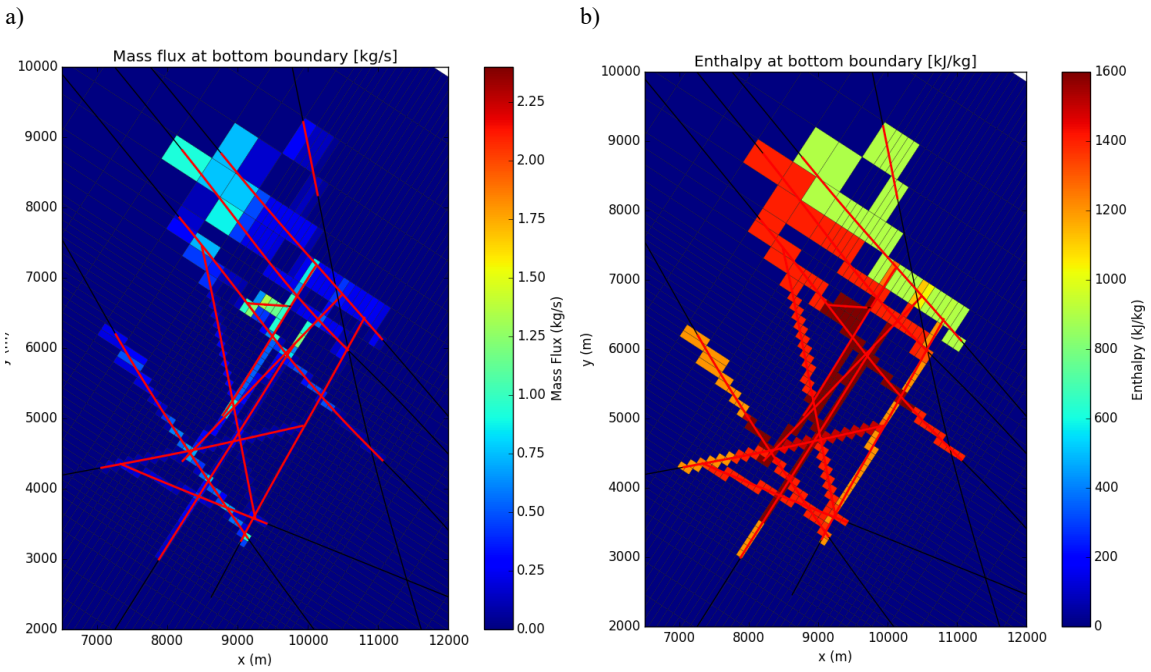

Figure 8: Distributed (a) mass flux and (b) enthalpy bottom boundary conditions for a multi-million block geothermal production model. The faults present at the bottom of the model are indicated with black lines along their traces. Red sections on the fault traces indicate the locations of the upflow.

3. MODEL POST PROCESSING

Post-processing multi-million block models efficiently requires specialised, automated tools. We use python scripts extensively for extracting and collating data from the large, binary format Waiwera output files. Standard plots, such as downhole temperature comparisons, are still important calibration tools and are generated regularly. More sophisticated 2D plans and slices are also generated using python scripts and provide useful visual information of a complex and detailed results. The plots in Figure 9 compare model temperature slices for three different resolution models of the same system. It is interesting to note how similar the results are for this complex production history model despite the model in Figure 10c having almost 20 times as many blocks as the model in Figure 10a while covering the same model domain . This demonstrates the consistency between the multiple models we have set up using out downscaling approach. There are some important differences apparent as well. The plots in Figures 10b and 10c show that the higher resolution models resolve reinjection returns much more sharply than the model presented in Figure 10a.

a)

b)

c)

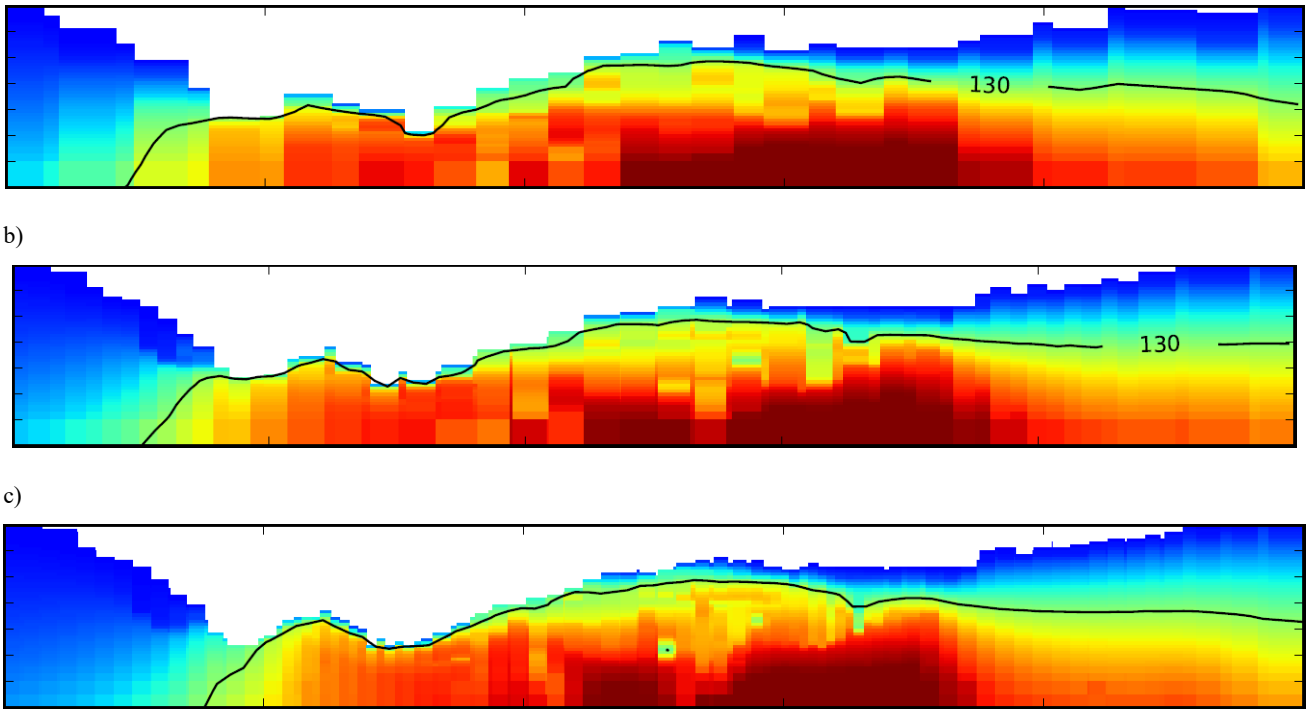

Figure 9: Temperature distribution cross-sections from three different resolution models of the same system.

Alongside our suite of python scripting tools we make extensive use of Leapfrog Geothermal's powerful 3D visualization capabilities. We load in output from multi-million block models in the form of point clouds and the use Leapfrog's radial basis functions to generate complex isosurfaces to provide insight into the simulation results. Leapfrog's movie making capabilities are also invaluable for communicating results from such large, complex models. Figure 10 shows an example of the highly detailed 3D surfaces that can be generated using Leapfrog and Waiwera together.

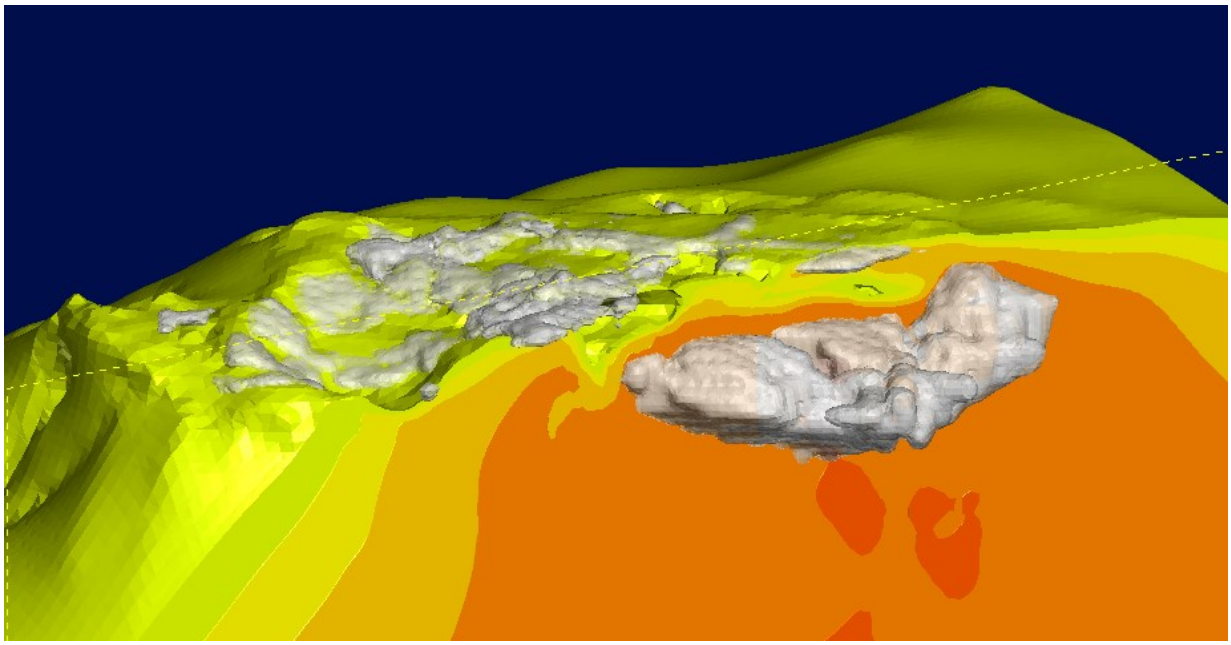

Figure 10: A 3D temperature distribution produced using Leapfrog Geothermal using results from a high-resolution Waiwera simulation. The 3D distribution of the shallow steam zone is also shown.

4. CONCLUSIONS

The recent advances in geothermal reservoir modelling tools mean that we are capable of running much larger and more complex simulations than ever before. With a new generation of reservoir models comes a new generation of challenges associated with managing such large simulations. Using our new geothermal simulator Waiwera we have developed a system where reservoir models can defined through a set of a grid-independent data files that can be used to generate models of any resolution. This not only facilitates the setup and handling of multi-million block models but importantly it provides a framework for creating multiple models of the same system with different resolutions. Our approach ensures that the models are consistent despite having resolutions that can vary by orders of magnitude. Having consistent models with different resolutions allow modellers to choose the appropriate resolution to generate specific forecasts. It also modellers to use lower resolution models to help calibrate the high resolution models therefore speeding up the calibration process.

5. ACKNOWLEDGEMENTS

The Geothermal Supermodels Programme is supported by a grant (C05X1306) from the NZ Ministry of Business, Innovation and Employment (MBIE), and by Contact Energy Ltd.

REFERENCES

Burnell, J., O'Sullivan, M., O'Sullivan, J., Kissling, W., Croucher, A., Pogacnik, J., Pearson, S., Caldwell, G., Ellis, S., Zarrouk, S. and Climo, M.: Geothermal Supermodels: the Next Generation of Integrated Geophysical, Chemical and Flow Simulation Modelling Tools. *Proc. World Geothermal Congress 2015*, Melbourne, Australia, 19-25 April (2015).

Croucher, A.E.: PyTOUGH: a Python scripting library for automating TOUGH2 simulations. *Proc. 33rd NZ Geothermal Workshop*, 21 – 23 November 2011, Auckland, New Zealand (2011).

Croucher, A.E., O'Sullivan, M.J., O'Sullivan, J.P., Pogacnik, J., Yeh, A., Burnell, J. and Kissling, W.: Geothermal Supermodels Project: an update on flow simulator development. *Proc. 37th NZ Geothermal Workshop*, 18 – 20 November 2015, Taupo, New Zealand (2015).

Croucher, A.E., O'Sullivan, M.J., O'Sullivan, J.P., Pogacnik, J., Yeh, A., Burnell, J. and Kissling, W.: Geothermal Supermodels Project: an update on flow simulator development. *Proc. 38th NZ Geothermal Workshop*, 23 – 25 November 2016, Auckland, New Zealand (2016).

Croucher, A.E., O'Sullivan, M.J., O'Sullivan, J.P., Pogacnik, J., Yeh, A., Burnell, J. and Kissling, W.: Geothermal Supermodels Project: an update on flow simulator development. *Proc. 39th NZ Geothermal Workshop*, 22 – 24 November 2017, Rotorua, New Zealand (2017).

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

Omagbon, J., O'Sullivan, M. J., O'Sullivan, J. P., & Walker, C.; Experiences in developing a dual porosity model of the Leyte geothermal production field. *Proc. 38th New Zealand Geothermal Workshop*. Auckland (2016).

O'Sullivan, J., Dempsey, D., Croucher, A. E., Yeh, A., & O'Sullivan, M. J.: Controlling complex geothermal simulations using PyTOUGH. In R. Horne (Ed.), *Proc. 38th Workshop on Geothermal Reservoir Engineering*. Stanford University, Stanford, California, USA (2013).

O'Sullivan, J.P., Croucher, A.E., Yeh, A., and O'Sullivan, M.J.: Experiments with Waiwera, a new geothermal simulator. *Proc. 39th NZ Geothermal Workshop*, 22 – 24 November 2017, Rotorua, New Zealand (2017).

O'Sullivan, J., Arthur, S., & O'Sullivan, M.: A new reservoir model to support environmental monitoring of the Orakeikorako geothermal system. *Geothermics*, 59, 90-106, (2016).

O'Sullivan, M. J., O'Sullivan, J. P., & Nugraha, R. P.: A 3D geological model and natural state simulation of Jaboi geothermal field, Nangro, Aceh Darussalam, Indonesia. *Proc of the New Zealand Geothermal Workshop*. Wairakei (2018).

Popineau, J., O'Sullivan, J. P., O'Sullivan, M. J., Archer, R., and Williams, B.: An integrated Leapfrog/TOUGH2 workflow for a geothermal production modelling. In *7th African Rift Geothermal Conference* (ARGeo). Kigali, Rwanda (2018).

Pruess, K., Oldenburg, C. and Moridis, G.: TOUGH2 user's guide, version 2.0. LBNL-43134, Lawrence Berkeley National Laboratory, Berkeley, California (1999).

Ratouis, T., OSullivan, M. J., O'Sullivan, J. P., McDowell, J., & Mannington, W. I.: Holistic approach and recent advance in the modelling of the Ohaaki geothermal system. *Proc. 39th New Zealand Geothermal Workshop*. Rotorua (2017).

Ratouis, T. M., O'Sullivan, M. J., Alcaraz, S. A., & O'Sullivan, J. P. (2017). The effects of seasonal variations in rainfall and production on the aquifer and surface features of Rotorua geothermal field. Geothermics, 69,165-188

Wellmann, J.F., Croucher, A.E. and Regenauer-Lieb, K.: Python scripting libraries for subsurface fluid and heat flow simulations with TOUGH2 and SHEMAT. *Computers & Geosciences* **43**, 197-206 (2012).

Yeh, A., Croucher, A. E., & O'Sullivan, M. J.: TIM - Yet another graphical tool for TOUGH2. In *35th New Zealand Geothermal Workshop: 2013 Proceedings*. Rotorua, New Zealand (2013).

Yeh, A., Croucher, A. and O'Sullivan, M.J.: Recent developments in the AUTOUGH2 simulator. *Proc. TOUGH Symposium 2012, Berkeley*, California, September 17-19 (2012).