Experiments with Waiwera, a new geothermal simulator

John O'Sullivan[1], Adrian Croucher[1], Angus Yeh[1] and Mike O'Sullivan[1]

[1]Engineering Science, University of Auckland, Auckland, New Zealand

jp.osullivan@auckland.ac.nz

Keywords: *Reservoir models, numerical modelling, flow simulator, Ngawha*

ABSTRACT

In a companion paper we discuss the development of our new geothermal simulator, called Waiwera, and present results of tests on benchmark problems. In this paper we discuss the application of Waiwera to natural state simulations with models of Ngawha (New Zealand) and System B. These two models were chosen because they both involve mixtures of water and a non-condensible gas (carbon dioxide for Ngawha and air for System B) and they both have large contrasts in permeability. As a result the models run badly with a standard version of TOUGH2.

Recently we have made changes in Waiwera and AUTOUGH2 (our version of TOUGH2) to the way some aspects of two-phase flow are handled. We found that these changes greatly improved the performance of the Ngawha and System B models.

TOUGH2 and Waiwera solve the nonlinear mass and energy balance equations by using the Newton-Raphson (N-R) iterative procedure. At each N-R iteration a large system of linear equations has to be solved. Typically in a difficult natural state simulation as the time step increases the condition number of the linear equations degrades and the N-R process may struggle to converge. With Waiwera we have found, by experiment, that careful selection of linear solvers and preconditioners can greatly improve numerical performance. Results for the Ngawha and System B models using different linear solvers and preconditioners are discussed below.

1. INTRODUCTION

1.1 Waiwera

A new geothermal simulator, Waiwera, has been developed as a collaboration between the University of Auckland and GNS Science. The aim is to produce a replacement for the industry standard, TOUGH2 (Pruess et al. 1999). The Waiwera code is written in object-oriented Fortran 2003 and parallelised using the PETSc library for scientific computation (Balay et al., 2016), and will be released under an open-source license.

In a companion paper (Croucher et al. 2017) and elsewhere (Burnell et al. 2015; Croucher et al. 2015, 2016) the development of Waiwera and tests on several benchmark problems are described.

1.2 Test cases

The present paper discusses the application of Waiwera to simulations of the natural state of two geothermal systems, namely Ngawha (New Zealand) and System B. These two models were selected because they both run poorly with a standard version of TOUGH2, i.e. they do not converge well to a stable steady state. Even after many time steps the size of the time step and the total time are not as large as desired. A well-performing natural state model will achieve a time step of 10^{13} - 10^{14} s and a total time of 10^{15} - 10^{16} s in less than 500 time steps. With a standard version of TOUGH2 both the Ngawha and System B models struggle to achieve this.

The features of the the models of Ngawha and System B that make them run slowly are: they both involve mixtures of water and a non-condensible gas (carbon dioxide for Ngawha and air for System B) and they both have large contrasts in permeability. We have found with other models that these characteristics can give poor convergence to a natural state.

2. SIMULATOR IMPROVEMENT

2.1 AUTOUGH2 and Waiwera

Over the years we have made several modifications to AUTOUGH2 (the University of Auckland version of TOUGH2) to improve its performance and to make it more flexible for running future scenarios (Yeh et al. 2012). Most of these modifications have been incorporated into Waiwera, and conversely, some innovations from Waiwera have recently been introduced into AUTOUGH2. The most important of these modifications are discussed below.

2.1.1 Phase transitions

Phase transitions in complex geothermal reservoir models are a common source of time step size reductions and hence slow progress to natural state. AUTOUGH2 incorporates a method for improving phase transition behaviour, based on computing properties of absent phases (O'Sullivan et al., 2013), which alleviates the problem to some extent. However, it introduces some extra computational cost, and complicates the equation of state code, particularly for more complex EOS modules such as those for mixtures of water and non-condensible gases (NCGs), and for supercritical fluid.

In Waiwera we have developed a new approach based on a reformulation of the multi-phase gravity term (Croucher et al., 2016). Our experience is that this works at least as well as the absent-phase approach, is much simpler to implement and also works with more complex EOS modules.

Waiwera (and some EOS modules in AUTOUGH2) also recently introduced an improved variable-switching scheme for phase transitions, in which pressures and temperatures are interpolated on to the saturation curve (Croucher et al., 2017). This also helps to avoid phase transition failures and time step reductions.

2.1.2 Convergence criteria

The basic numerical task carried out by TOUGH2 and Waiwera is to solve a large system of nonlinear equations arising from the conservation equations for mass, energy and possibly non-condensible gas (NCG) (and/or other chemicals). The primary variables being solved for in each

cell are typically pressure, temperature (or vapour saturation) and mass fraction or partial pressure of gas.

In both TOUGH2 and Waiwera, this system of nonlinear equations is solved using an iterative Newton-Raphson method. In TOUGH2, the iterative solution process is considered to be converged when the balance equations in all cells have been satisfied to within a specified tolerance.

Many nonlinear solvers, such as the PETSc 'SNES' solvers used in Waiwera, allow an alternate convergence criterion: when the relative sizes of the updates made to the primary variables are smaller than a second specified tolerance, the Newton-Raphson method can also converge. AUTOUGH2 also introduced a similar alternate convergence criterion, which can become important when the model is near steady state (O'Sullivan et al., 2014).

2.1.3 Other improvements

Various other improvements have been made to Waiwera, some with a view to improving natural state model convergence. For example, Waiwera recently switched to using non-dimensionalised primary variables in the Newton-Raphson solution process, and a more efficient and accurate method for evaluating the polynomials in NCG equations of state. More details can be found in Croucher et al. (2017).

2.2 Linear solvers and preconditioners

At each iteration of the Newton-Raphson method used to solve the nonlinear mass and energy balance equations, a system of linear equations must be solved. The number of unknowns can be large ($\sim 10^4$ for the models considered here) but the system is sparse because the balance equations in each cell involve unknowns from only a few other cells (typically 21 unknowns from 7 cells for a regular 3D grid and a water / NCG EOS).

For solving large sparse systems of linear equations, iterative conjugate gradient methods are appropriate. The conjugate gradient solvers GMRES and BiCGSTAB were introduced into AUTOUGH2 (Bullivant et al. 1991). They were subsequently were added into the standard version of TOUGH2 (Moridis and Pruess, 1995, 1997), together with a modified option BiCGSTAB(*l*) (Sleijpen and Fokkema, 1993) which sometimes works better on difficult problems. The PETSc library (Balay et al., 2016) used by Waiwera offers a wide range of parallelised linear solvers and options for tuning their performance. For example, the PETSc GMRES solver has an option to use the 'restarted' GMRES algorithm (Saad and Schultz, 1986), in which the standard GMRES solution process is repeatedly restarted after a specified number of iterations. The TOUGH2 / AUTOUGH2 GMRES solvers do not offer this option.

These linear solvers generally work well but they are iterative, producing only an approximate solution. This is usually not a problem but with a difficult natural state simulation the matrix may become poorly conditioned and the quality of the solution may degrade. The poor conditioning of the matrix can be dealt with to some extent by applying preconditioning.

The standard preconditioner available in AUTOUGH2 and TOUGH2 is incomplete LU factorization. With this method an approximate LU factorization of the matrix **J** is used as a preconditioner. Various versions of ILU are available, depending on the level of approximation of **L** and **U**, the lower and upper triangular matrix factors of **J**. If the sparsity pattern of **J** is used in **L** and **U** then the method is referred to as ILU(0). However the effectiveness of preconditioning can be increased by allowing the sparsity pattern of **L** and **U** to match the sparsity pattern of $\mathbf{J}^{N+1}$. In this case the method is referred to ILU(N) and N is referred to as the level of fill-in (Chan and van der Vorst, 1997).

The preconditioners offered by the PETSc library and used in Waiwera are parallelised. In this case the high-level parallel preconditioner is distributed amongst the processors, each of which has its own local "sub-preconditioner", running in serial. Typically ILU is used for the sub-preconditioner, and the level of fill-in can be specified (this option is not available for the ILU preconditioners used in TOUGH2 or AUTOUGH2). However, ILU cannot readily be used for the high-level parallel preconditioner, because the ILU algorithm does not scale well in parallel. For Waiwera we have experimented mainly with two of the parallel preconditioners, Block Jacobi and Additive Schwarz, provided by PETSc.

2.3 Equation of state for water / NCG

We have recently developed an equation of state (EOS) module for simulating non-isothermal mixtures of water and generic non-condensible gases (NCGs) in Waiwera (Croucher et al., 2017). Specific EOS modules have been derived from this for water / CO_2 and water / air mixtures. The NCG energy of solution is calculated using a method similar to that used in TOUGH2's "EWASG" EOS, rather than the approaches taken in TOUGH2's other NCG EOS modules, some of which (e.g. the EOS3 and EOS4 water / air modules) assume a zero energy of solution. We have found that including the energy of solution for air often improves natural state model convergence for water / air models.

3. NGAWHA MODEL

A plan view of the grid used for the Ngawha model is shown in Figure 1. It has 1015 cells per layer and 32 layers, with some of the shallow layers being incomplete as the top of the model follows the topography), giving a total number of 29,590 cells. A CO2/water equation of state (EOS) is used because several of the wells produce ~1% mass fraction of CO2. In this version of the model the top is set at the estimated position of the water table, i.e., the vadose zone is not included in the model.

A vertical slice through part of the model, showing the layer structure is given in Figure 2.

This particular version of the Ngawha model was selected because of its poor numerical behaviour with TOUGH2, thus providing a severe test of simulator performance. From a physical perspective the model has some shortcomings, with some of the temperatures at the base of the model being too high, up to 336°C. (Strictly speaking, these temperatures are slightly outside the range of applicability for some of the CO_2 thermodynamic functions used in TOUGH2 and Waiwera, such as the Henry's constant equation which is valid only up to 300°C.)

The grid is similar to that used by Burnell et al. (2016) who ran natural state and production history simulations. In the present study only the natural state is investigated.

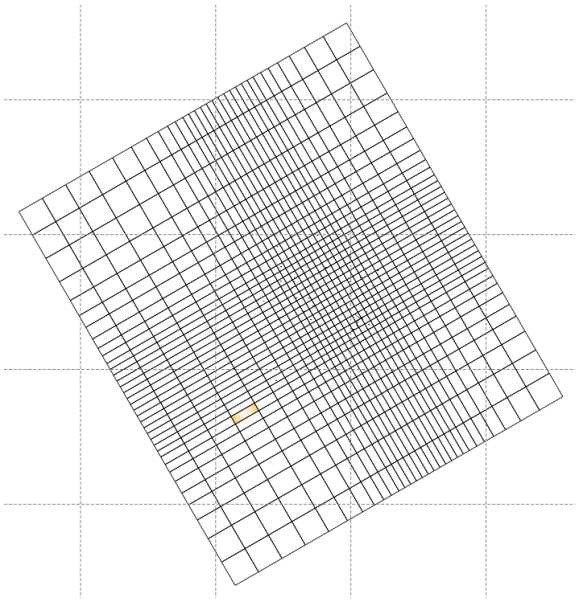

Figure 1: Plan view of grid for the Ngawha model

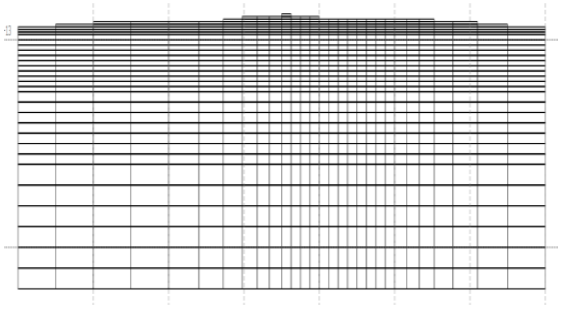

Figure 2: Vertical slice through the Ngawha model

Simulations were carried out with TOUGH2, AUTOUGH2 and Waiwera on a desktop PC with a 12-core Intel Xeon E5-2670 processor. Natural state was considered to be achieved at time 1E+15 s.

Table 1 shows the performance of the TOUGH2 and AUTOUGH2 simulators on this problem. The TOUGH2 run failed after only 6 time steps, with an unrecoverable error in the linear solver. Other linear solvers such as Bi-CGSTAB(*l*) gave similar behaviour. AUTOUGH2 did manage to reach a natural state solution, but it took nearly 400 time steps and over 8.5 hours of run time.

Table 1: TOUGH2 / AUTOUGH2 natural state simulations for Ngawha

Simulator	Linear solver	Time steps	Run time (min)
TOUGH2	Bi-CGSTAB	-	Run failed
AUTOUGH2 (v 2.42)	Bi-CGSTAB	397	518

Table 2 shows Waiwera's performance on this problem, for various combinations of linear solvers, preconditioners and numbers of processors (np). In each case, as well as the number of time steps and total run time, the number of linear solver failures is given. For all cases shown, the ILU(0) sub-preconditioner was used.

Our initial experiments used the Bi-CGSTAB(*l*) linear solver and Additive Schwarz (ASM) preconditioner, as this had been an effective combination for other models (e.g. Wairakei), usually giving better performance than GMRES or BiCGSTAB (Croucher et al., 2016). For the Ngawha model this gave natural state run times of approximately 1 – 1.5 hours (depending on the number of processors), i.e. about 6 - 7 times faster than AUTOUGH2, and with fewer time steps but with the increased speed mainly due to the parallelisation. However, the linear solver performance was still not very good, with typically over 100 solver failures per run. Large numbers of failures also often mean that even when the linear solver does converge, it does so slowly. Here the number of failures generally increased with the number of processors, so that the run times were not decreasing as the number of processors increased.

Switching to the Block Jacobi preconditioner improved the performance considerably, with solution times for 8 and 10 processors down to less than 15 minutes and with far fewer solver failures. Unusually, the 6-processor run performed even better, with a 3.5 minute run time, no solver failures and only 28 time steps.

Table 2: Waiwera natural state simulations for Ngawha

Linear solver	Precon-ditioner	np	Time steps	Run time (min)	Solver failures
Bi-CG-STAB(*l*)	ASM	4	249	78.9	95
		6	278	72.7	107
		8	318	81.1	125
		10	299	91.0	117
	Block Jacobi	4	252	69.1	96
		6	28	3.5	0
		8	67	13.2	16
		10	42	8.7	6
GMRES (200)	ASM	4	82	23.2	10
		6	52	9.0	5
		8	51	10.4	4
		10	34	4.7	1
	Block Jacobi	4	50	10.5	4
		6	41	4.0	1
		8	33	2.0	0
		10	31	1.9	0

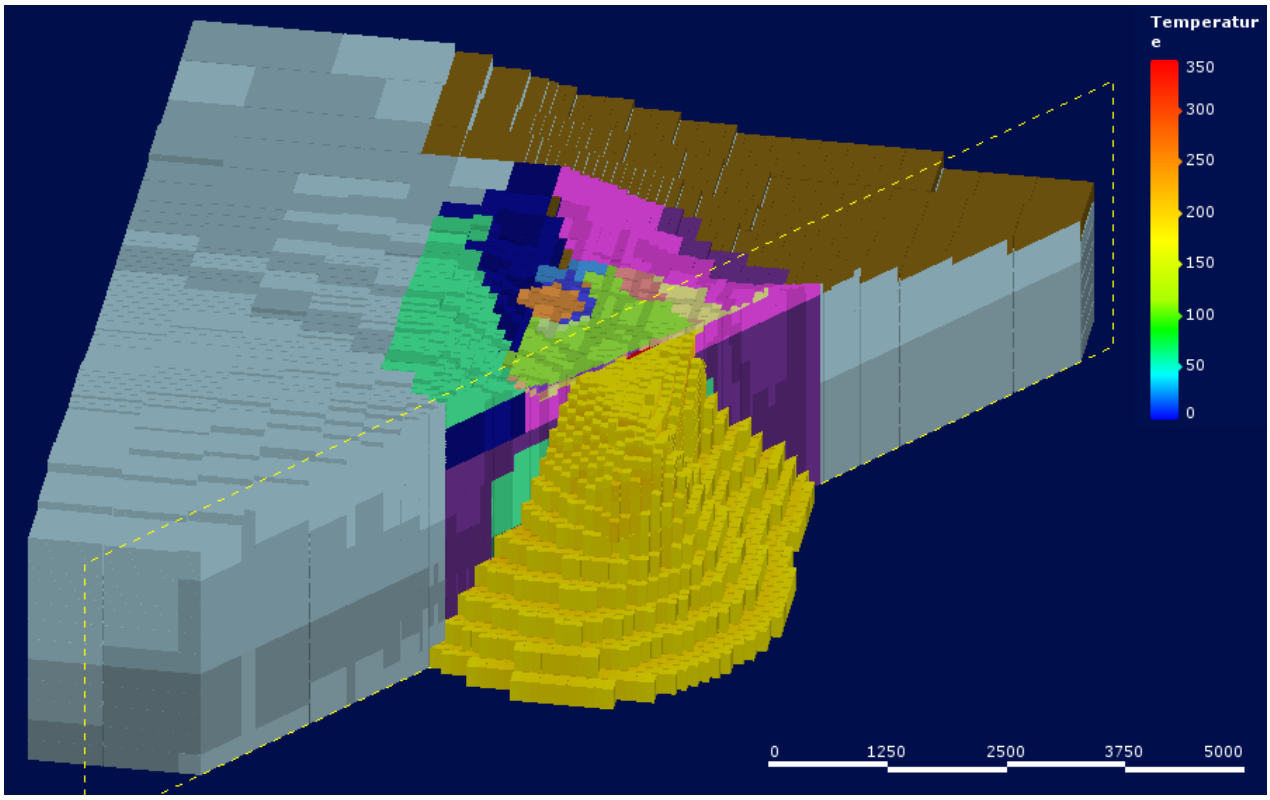

Figure 3: The System B model shown in Leapfrog Geothermal. The back half of the slice shows the model rock type groupings and the front half of the slice shows the blocks with temperatures above 180C.

Following the advice of the PETSc documentation, we then tried using the restarted GMRES linear solver, and experimented with various restart intervals. Our experiments indicated that restarting after 200 iterations gave the best results for this problem, so only results for GMRES(200) are shown here. For GMRES(200) and the ASM preconditioner, the run times were generally 1 – 3 times faster than those for BiCGSTAB(*l*) with the Block Jacobi preconditioner (except when running on 6 cores where the latter combination performed unusually well). However, the best results of all were obtained by combining GMRES(200) with the Block Jacobi preconditioner, particularly for 8 and 10 processors where there were no linear solver failures and the run times were under 2 minutes. For both preconditioners, the GMRES(200) solver performed better as the number of processors increased, with fewer failures and lower time step counts (which was not generally the case for the BiCGSTAB(*l*) solver).

For the cases where linear solver failures still occurred, we also experimented with increasing the level of ILU sub-preconditioner fill-in. However this did not appear to reduce the instance of failures. For this model, the most effective way to achieve good performance was by using the restarted GMRES(200) linear solver.

This problem is clearly extremely ill-conditioned, as the convergence results are very sensitive to the details of the linear solver and preconditioner. However, Waiwera was able to produce a converged natural state solution up to 200 times faster than AUTOUGH2.

4. SYSTEM B MODEL

The grid and rock type distribution used for the System B model are shown in Figure 3. It has 2496 cells per layer and 77 layers, with some of the shallow layers being incomplete as the top of the model follows the topography (on land) and the bathymetry (under the ocean), giving a total number of cells of 127485. An air/water equation of state (EOS) is used as it is important to accurately represent the shallow zone including the unsaturated, vadose zone.

Initial conditions were obtained by interpolating results from a coarser model and then natural state simulations were carried out with TOUGH2, AUTOUGH2 and Waiwera with different choices of preconditioner. The model was run either to convergence or to a total of 500 time steps. The results for TOUGH2 and AUTOUGH2 are given in Table 3 and the results for Waiwera in Table 4.

For TOUGH2 and AUTOUGH2 simulations were carried out on a desktop PC with an Intel Xeon E5-2670 processor. The Waiwera simulations were carried out on the NeSI Pan Cluster at the University of Auckland. They were submitted to the Sandy Bridge architecture cores which have Intel E5-2680 processors and 16 cores per node. The internal network uses QDR Infiniband at 40Gb/s. Natural state was considered to be achieved at time 1E+15 s.

Table 3: TOUGH2 / AUTOUGH2 natural state simulations for System B

Simulator	Linear solver	Time steps	Max. time step	Run time (min)
TOUGH2	Bi-CG, Lanczos-type PC	500	1.51×10^{13}	1052
AUTOUGH2 (v 2.42)	Bi-CGSTAB	500	2.20×10^{13}	1106

Table 4: Waiwera natural state simulations for System B

Linear solver	Precon-ditioner	np	Time steps	Run time (min)	Solver failures
GMRES(*l*) (200/10)	ASM	8	405	141.2	6
		16	381	63.9	2
		32	363	50.9	1
		64	497	203.6	28
	Block Jacobi	8	403	209.6	4
		16	416	126.3	11
		32	376	86.9	10
		64	778	149.3	13

Table 3 shows that neither TOUGH2 nor AUTOUGH2 achieved the natural state target time within 500 time steps. In both cases the linear solves become very expensive as the time step size increases and the system of equations become more ill-conditioned. TOUGH2's time step size plateaued at approximately 1.5×10^{13} seconds and when the simulation reached 500 time steps it had only achieved a total time of 1.43×10^{14} seconds, 14% of the total target. Using the default linear solver settings in TOUGH2 equates to the Bi-CG Lanczos-type preconditioned method with the number of iterations in the linear solver limited to 10% of the number of equations (~500,000 for this model). Thus the linear solver was allowed up to 50000 iteration to obtain a solution resulting in no linear solver failures but very long solution times. Various options could be applied to attempt to optimise the solution process though at over 17 hours to achieve the first 500 timesteps even the most effective combination of settings is likely to result in excess of two days to achieve a converged natural state.

The results for AUTOUGH2 were quite similar to those obtained by TOUGH2. AUTOUGH2 achieved a slightly higher maximum time step and as a result achieved a total time of 2.32×10^{14} seconds, 23% of the total target. However the larger time steps required more effect in the linear solver and the run time to achieve 500 time steps was longer at over 18 hours.

A number of different combinations of linear solvers and pre-conditioners were tested for the Waiwera simulations of the System B model using a range of different numbers of processors. Following on from the results of Ngawha model GMRES was generally found to more effective than Bi-CGSTAB and Bi-CGSTAB(*l*). Also the accelerated variant GMRES(*l*) (Baker *et. al*., 2005) was found to perform better than the standard GMRES solver and the restart parameter of 200 was found to be effective. As such, detailed timing runs and comparisons of two different types of pre-conditioners were only run using GMRES(*l*).

Table 4 shows that unlike the Ngawha model, the ASM pre-conditioner outperforms the Block Jacobi pre-conditioner. It also shows that in both cases increasing the number of processors from 32 to 64 results in significant increases in the run times. The number of linear solver failures increases as well which not only increases the run times but the number of time steps required to achieve a converged solution. For both the ASM and Block Jacobi pre-conditioners the speedup is good with an increase from eight to 16 processors. Increasing to 32 processor offers smaller gains in both cases.

Comparing the results between Waiwera and TOUGH2 and AUTOUGH2 shows that not only does Waiwera scale well for the System B model but its better implementation of the solution algorithm combined with more effective linear solver options allow a converged solution to be achieved much more rapidly. Results are now available with an hour as opposed to requiring days making calibration of this large model now possible.

5. CONCLUSIONS

The result presented in this paper highlight the challenges presented by simulating natural states of geothermal systems where the effects of NCGs must be included. Solving problems where the true topography is included and air in the unsaturated vadose zone is interacting with a boiling zone are particularly difficult. However, there are many cases where representing the interaction between the NCGs and the water in the system is critical for achieving accurate estimates of the behaviour of the system.

Waiwera provides a step change in our ability to simulate these types of systems. It is fast and shows good scaling properties in terms of parallelisation. Improvements in the NCG equation-of-state algorithms also mean its convergence behaviour is significantly better than both TOUGH2 and AUTOUGH2.

Our studies have shown that the choice of linear solver and preconditioner are important and also problem-dependent. We have found that restarted GMRES a good option to try and increasing ILU fill-in can also be necessary. The PETSc libraries that Waiwera is built on allow access to a wide range of solvers, preconditioners and options that we recommend users investigate for their specific simulations.

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

Baker, A.H., Jessup, E.R., and Manteuffel, T.A.: A technique for accelerating the convergence of restarted GMRES. *SIAM Journal on Matrix Analysis and Applications*, 26 (2005).

Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, B., Karpeyev, D., Kaushik, D., Knepley, M., McInnes, L., Rupp, K., Smith, B., Zampini, S. and Zhang, H.: *PETSc Users Manual*, ANL 95/11, Revision 3.7, Argonne National Laboratory (2016).

Bullivant, D. P., O'Sullivan, M. J., & Zyvoloski, G. A.: Enhancements of the MULKOM geothermal simulator. *Proc. 13th New Zealand Geothermal Workshop*, University of Auckland, Auckland, New Zealand. pp. 175-182. (1991).

Burnell, J., O'Sullivan, M. J., O'Sullivan, J. P., Kissling, W., Croucher, A. E., Pogacnik, J., . . . Climo, M.: Geothermal supermodels: The next generation of integrated geophysical, chemical and flow simulation modelling tools. *Proc. World Geothermal Congress 2015*. Melbourne, Australia. (2015).

Burnell, J., Weir, G. and Kissling, W.: Modelling the Ngawha geothermal field. *Proc. 38th New Zealand Geothermal Workshop*, University of Auckland, Auckland. (2016).

Chan, T.F. and van der Vorst, H.A.: Approximate and incomplete factorizations. In: *Parallel Numerical Algorithms*, ed. Keyes, D.A., Sameh, A. and Venkatakrishnan, V. Kluwer, pp. 167 – 202 (1997).

Croucher, A. E., O'Sullivan, M. J., O'Sullivan, J. P., Pogacnik, J., Yeh, A., Burnell, J., & Kissling, W. (2015). Geothermal Supermodels project: an update on flow simulator development. In *Proceedings 37th New Zealand Geothermal Workshop*. Wairakei, New Zealand. (2015).

Croucher, A., OSullivan, M. J., O'Sullivan, J. P., Pogacnik, J., Yeh, A., Burnell, J., & Kissling, W.: Geothermal supermodels project: an update on flow simulator development. *Proc. 38th New Zealand Geothermal Workshop*, University of Auckland, Auckland. (2016).

Moridis, G. J. and Pruess, K.: *Flow and Transport Simulations Using T2CG1, A Package of Conjugate Gradient Solvers for the TOUGH2 Family of Codes.* Report: LBNL-36235, UC-814, 1240, Earth Sciences Division, Lawrence Berkeley National Laboratory Berkeley, CA 94720, pp. 1-118 (1995).

Moridis, G. J. and Pruess, K.: *T2SOLV: an enhanced package of solvers for the TOUGH2 family of reservoir simulation codes.* Report: LBNL-40933, UC-1240, Earth Sciences Division, Lawrence Berkeley National Laboratory Berkeley, CA 94720. pp. 1-36 (1997).

Pruess, K., C. Oldenburg, and G. Moridis, *TOUGH2 User's Guide, Version 2.0*, Report: LBNL-43134, Lawrence Berkeley Laboratory, Berkeley, California. (1999).

O'Sullivan, J. P., Croucher, A. E., Yeh, A., & O'Sullivan, M. J.: Improved convergence for air-water and CO2-water TOUGH2 simulations. *Proc. 35th New Zealand Geothermal Workshop*. Rotorua, New Zealand. (2013).

O'Sullivan, J. P., Croucher, A., Yeh, A., & O'Sullivan, M. J.: Further Improvements in the Convergence of TOUGH2 Simulations. *Proc. 11th World Congress on Computational Mechanics - WCCM XI 5th European Congress on Computational Mechanics - ECCM V 6th European Congress on Computational Fluid Dynamics - ECFD VI*, Vol. 5 (pp. 5929-5940). Barcelona, Spain. (2014).

Saad, Y. and Schultz, M.H.: GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* 7(3), 856 - 869 (1986).

Sleijpen, G.L.G. and D. Fokkema: BiCGSTAB(*l*) for linear equations involving unsymmetric matrices with complex spectrum, *Electronic Transactions on Numerical Analysis* 1**,** 11-32 (1993).