RECENT DEVELOPMENTS IN THE AUTOUGH2 SIMULATOR

Angus Yeh, Adrian E. Croucher, and Michael J. O'Sullivan

Department of Engineering Science, University of Auckland
Private Bag 92019
Auckland 1142, New Zealand
e-mail: cyeh015@aucklanduni.ac.nz

ABSTRACT

For over a decade, the geothermal modeling group at the University of Auckland has been improving the capability of AUTOUGH2, a modified version of TOUGH2. This paper describes these additional features and the reasons they were introduced.

One of the main changes with AUTOUGH2 is that all EOS modules are compiled with other subroutines into one executable, and the user specifies appropriate EOS flags in the input file. Several new EOS modules have also been added. The input and output file names are also specified by the user directly at run time.

Many internal data array sizes, such as the number of rock types, have been increased. More recently, dynamic allocation of most of the major arrays has been implemented to improve the efficiency of memory use.

The most important change made in AUTOUGH2, however, has been the addition of many new generator (sink and source) types. For example, a recharge well type was added to ease the implementation of recharge boundary conditions. Another new well type was implemented to allow new production and injection wells to be added automatically to meet a specified steam flow or mass flow target as reservoir conditions change over time. These well types were introduced to aid modeling studies of geothermal power projects conducted at the University of Auckland.

INTRODUCTION

The geothermal modeling group at the University of Auckland started working on the MULKOM code in the early 1980s and this work has continued for more than three decades as the TOUGH and TOUGH2 simulators (Pruess et al., 1999) were introduced. The main aims of this continuing effort were to increase execution speed and introduce additional input/output options and graphical utilities (Bullivant, 1990; O'Sullivan and Bullivant, 1995). This customization has accumulated to form the present-day AUTOUGH2.

One of the earliest changes was the introduction of iterative conjugate gradient linear equation solvers to replace the direct solver (MA28) used in MULKOM (Bullivant et al., 1991). (Subsequently, similar solvers have been added to standard TOUGH2 (Moridis, et al., 1998).) Also, faster and more accurate versions of the thermodynamic routines were implemented pre-1990. The combining of all equation of state (EOS) modules into a single executable was implemented at an early stage in the 1990s, along with minor input/output changes. Many new generator types were coded during this time. More generator types were added, and some old ones were modified during the 2000s as new requirements came up. Changes in array size/memory management have been introduced very recently.

Most changes made to TOUGH2 leading to AUTOUGH2 were also implemented in iTOUGH2 to form AUiTOUGH2. Thus, most of the core code is shared between AUTOUGH2 and AUiTOUGH2.

The aim in the development of AUTOUGH2 has been to improve the ease of use, efficiency, and capability of TOUGH2.

RUNNING AUTOUGH2 – EASE OF USE

One of the most obvious changes from a user's perspective between AUTOUGH2 and TOUGH2 is the handling of different EOS

modules. AUTOUGH2 compiles all the EOS modules together with the rest of the TOUGH2 code into a single executable. The choice of EOS is achieved via an additional SIMULator block in the input file. This allows the user to choose the desired EOS as well as allowing backward compatibility with earlier versions of the code, such as MULKOM, TOUGH and TOUGH2.

Upon execution, the user is allowed to specify names of the model input file, incon file, and save file. This relatively small change made to the original TOUGH2 has proved to be both user-friendly and flexible. It makes it easy to run and maintain a series of model files in a single directory. This is found to be quite useful when lengthy calibration is carried out on a model, and all versions of the model need to be kept and tracked, with no external file/version control system needed.

OUTPUT MODIFICATION

In order to deal with the inclusion of all EOS modules in a single executable, some modifications to the output format for TOUGH2 were required.

Internally, AUTOUGH2 uses a piece of centralized "format control" code to print the simulation results appropriately for each EOS. The locally developed pre-/post-processing software MULgraph (Bullivant et al., 1995; Bullivant and O'Sullivan, 1998; O'Sullivan and Bullivant, 1995) uses the same piece of formatting control code to ensure that results for all EOS modules can be dealt with correctly. The consistency of output formatting across different EOS modules also helps facilitate machine-readability for post-processing.

In MULKOM, the only output options were results at a single element for every time step, or results at all elements every so many time steps. We introduced a SHORT output option so that results can be produced at every time step for a number of nominated elements, connections, or wells. (Later versions of TOUGH2 introduced the FOFT, COFT, and GOFT modules that provide similar functionality, via additional output files.)

FASTER THERMODYNAMICS

A series of changes was made to the original thermodynamics subroutines COWAT and SUPST. These two subroutines included a number of variables raised to integer powers as part of the calculations. We re-implemented these calculations using repeated multiplications, which resulted in a 5–10 fold increase in the speed of calculation of the secondary thermodynamic variables.

SUPERCRITICAL AUTOUGH2

Standard TOUGH2 and AUTOUGH2 both use the IFC-67 formulation (IFC, 1967) for thermodynamic properties of water. However, they both omit the equations dealing with supercritical conditions at high pressures and temperatures, a fluid state that is becoming of more interest as deep geothermal systems are being studied (Fridleifsson and Elders, 2005).

In addition, the 1967 formulation has since been superseded by the IAPWS-97 formulation (Wagner et al, 2000), which is faster, more accurate, more consistent and has a simpler representation around the critical point. This formulation has been implemented in an extended version of AUTOUGH2 with supercritical capability for EOS1, EOS3, and EOS4 (Croucher and O'Sullivan, 2008).

LINEAR SOLVERS

Back in the 1980s, we wished to set up models with more elements than the limit of around 500 imposed by MA28, the linear equation solver in MULKOM, and the memory limitations of the hardware of the day. To overcome this problem, we introduced iterative conjugate gradient solvers (Bullivant et al., 1991), which have been used in AUTOUGH2 ever since. Similar solvers were soon introduced in TOUGH2 (Moridis et al., 1998).

MORE EFFICIENT MEMORY USE

In MULKOM, TOUGH, and TOUGH2, all major arrays are statically allocated to predetermined fixed lengths, a restriction from early versions of the FORTRAN language and compilers. To use memory efficiently, it is necessary to recompile the code with array

dimensions set appropriately for the size of model being used.

In recent years, the size of models has grown, and the restriction of static memory allocation has become an important issue. On many operating systems, static memory allocation has a smaller size limit than memory allocated dynamically at execution time (Yeh et al., 2011).

The AUTOUGH2 code has been restructured to use dynamic memory allocation. At execution, the input file is processed first to determine the problem size. All major arrays are then allocated to their appropriate sizes. If the machine does not have enough free memory, the user is informed. In AUTOUGH2, all main arrays are dynamically allocated if their lengths depend on the numbers of rock types, elements, connections, generators or generation table entries.

NEW GENERATOR TYPES

Perhaps the most important change in AUTOUGH2 is the introduction of many more generator types. Some new generator types are used to deal with boundary conditions, while some are extensions of those provided by the original TOUGH2 code. Many others were designed with the motivation of simulating complicated real world geothermal production, injection, and power generation scenarios.

Table 1. Featured new generator types

Type	*Description*
Deliverability	
DELG	DELV with extended features
Boundary Condition	
HLOS	Heat loss boundary condition
RECH	Recharge boundary condition
Make-up production/injection	
TMAK	Total demand of make-up wells
DMAK	Make-up production wells
FINJ	Fixed injection
PINJ	Fixed portion injection
RINJ	Remained portion injection
IMAK	Pressure controlled injection

The following table provides a list of the main new generator types introduced. There are others not listed here, as they are minor variations of the listed ones or of the original set available with TOUGH2. Details of some of these new generator types are given in the following sections, with examples illustrating their use.

Deliverability Wells

TOUGH2 includes a DELV generator type, which allows the user to represent a geothermal well that may be partly throttled when it is first brought online. The user is allowed to limit the flow by specifying a maximum steam flow rate, which is calculated using a user-specified separator pressure.

One of the early additions to AUTOUGH2 was the DELG generator type, an extension of DELV. DELG allows users to specify a time-dependent productivity index in the form of a standard TOUGH2 well table. Alternatively, the user can specify an enthalpy-dependent bottom-hole pressure. There is also a very useful feature within DELG that automatically calculates the productivity index using the initial model pressure and a user-specified bottomhole pressure. This enables the model to start a simulation with each well at a specified flowrate, and then to vary the flowrate when the pressure changes over time. Thus, DELG can ensure a smooth flow-rate transition from a previous run to a subsequent one. A limit can be set with DELG to restrict mass or steam flowrate.

A typical use of DELG is for production wells in a future scenario. In the production history part of the model, a well may use a MASS generator, with the mass flow rate obtained from the actual production data. The same well can appear in the future scenario model as a DELG well. This allows the well to produce or cease production according to reservoir conditions.

DELG can also be used to simulate certain types of hot springs. Placing DELG at depth, it can simulate a localized direct link between the surface and the reservoir below the cap rock. Enthalpy-dependent bottom-hole pressure can be set to encourage high flow rates when the reservoir's boiling increases.

Boundary Conditions

With the original TOUGH2, modelers often needed to use special techniques to simulate an open boundary, for example by connecting a boundary block to a very large inactive block, to allow flow into or out of the model when the pressure in the boundary blocks changes. The RECH and HLOS generators were introduced into AUTOUGH2 to make this easier to implement.

RECH generators have behavior similar to that of a deliverability well, but can act as a source (inflow) or sink (outflow) depending on changes in reservoir pressure (unless the "no inflow" or "no outflow" options are used). The reference pressure can be specified manually or determined automatically from the initial conditions.

For example, at the bottom boundary of a model where natural hot upflow is represented by standard MASS generators, RECH generators can be added into a production/transient model. If the modeled pressure drawdown reaches the bottom boundary during production, the RECH generator will introduce additional upflow, at a specified enthalpy. In contrast, if brine injection raises the pressure at a boundary block above its initial state, a RECH generator can allow some fluid exit out of the model at a rate depending on how much overpressure exists.

The HLOS generator is similar to RECH, except that heat, instead of mass, is added or taken out.

Make-up Production/Injection Wells

Often geothermal reservoir models are used for investigating various future production/injection scenarios. In such simulations, it is useful to allow future "make-up" wells to be put into production when needed, and taken out when they become unproductive or are no longer needed.

At the core of the group of generators introduced for future scenario modeling are the TMAK and DMAK generators. The user provides a list of make-up wells, labeled with DMAK, in order of priority. This is followed by a TMAK well, which specifies the desired total mass flow and/or total steam flow (with a choice of separator pressure). The DMAK/TMAK option will automatically turn on new make-up wells if the production cannot meet the total mass or steam flow target. Individual DMAK wells have behavior similar to that of the DELG generators described above.

For the disposal of separated geothermal water and steam condensate, the TMAK/DMAK option keeps track of the amount of both separated geothermal water and steam. This can be reinjected via the new generator types FINJ, PINJ, RINJ, and IMAK, which are designed to deal with a wide range of injection configurations. All injection wells can be set to inject either the steam or separated water, and can be listed in any priority order.

FINJ injects a fixed amount. PINJ injects a fixed proportion of the total demand. RINJ injects a proportion of the remaining amount. IMAK behaves much like DMAK, using an injectivity calculation against a specified cut-off pressure, with an option to cap the amount injected.

Combinations of these generator types can be used to simulate the complicated scenarios commonly specified by field operators (O'Sullivan and Yeh, 2011). This is particularly useful for geothermal production in countries like New Zealand, where field operators need to optimize their operations for profitability, while conforming to environmental resource consent restrictions such as mass take limits, pressure constraints, etc.

Scenario Example with Make-up Wells

This is an example taken from a series of modeling studies for Contact Energy's proposed Tauhara II development near Taupo, New Zealand (O'Sullivan and Yeh, 2011). It is modified to demonstrate more clearly the application of make-up production and injection wells, and therefore does not reflect actual operation of the field.

Scenario specification

The specification included total steam demand for the power station, contributions from various parts of the field, and guidelines for disposal of used geothermal water:

TAKE

47,3000 t/day LP steam flow at 2.5 bar absolute, model to calculate total mass take

- 15% North Tauhara
- 40% East Tauhara
- 20% Centre Tauhara
- 25% South Tauhara

DISCHARGE

- 60% of steam flow injected as condensate in shallow East Tauhara Area
- 40% of steam flow lost to atmosphere from cooling tower
- Separated geothermal water (SGW) split equally between North, East, Centre, and South Tauhara

Building the scenario

To construct this scenario run, a list of potential production model blocks was prepared and used to form a series of DMAK generators, grouped into four respective production zones. Each zone of DMAK generators ends with a TMAK with the corresponding steam demand.

Each DMAK generator has its own bottom-hole pressure, depending on the depth of its model block. External wellbore simulation data were used to build the table of enthalpy-dependent bottom-hole pressures. This allows the well to flow with a lower cut-off pressure in two-phase or steam conditions. Note that a single DMAK generator here does not necessarily correspond to a single well in the field. It simply represents a portion of the production reservoir. A suitable productivity index was used and a limit on flow rate was set. A separator pressure of 2.5 bar absolute was set for all DMAKs.

A group of injection generators was inserted after the four DMAK/TMAK groups. The condensate injection uses the fixed portion PINJ generator type. The SGW injection was formed by a long list of IMAK generators. Each IMAK has a pressure limit depending on the location, depth and available overpressure from injection pumps. IMAKs from different zones were sorted so that the flow was split evenly into four zones. If a certain zone had difficulty injecting, the excessive SGW would be redistributed to other zones.

Results

Fig. 1 shows the result steam production from the scenario. All targets were met.

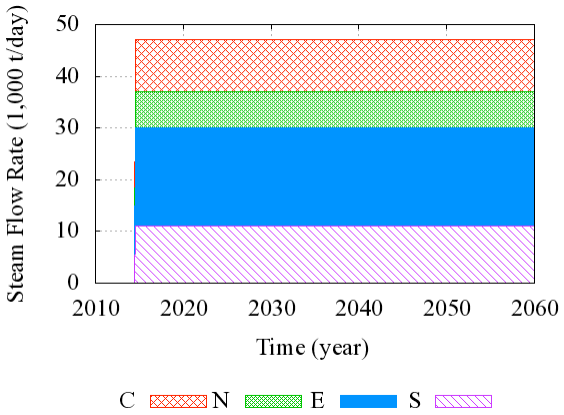

Figure 1. Steam flow rate from different production zones.

For each production zone, enough generators were turned on to maintain the target steam flow rate. With the original implementation of DMAK in AUTOUGH2, a DMAK could only be either completely off or fully open. In this case, if the steam flow only drops a small amount below the demand, a newly opened DMAK may add more than the required amount. This produces “steps” in the resulting mass flow rate, as shown in Fig. 2.

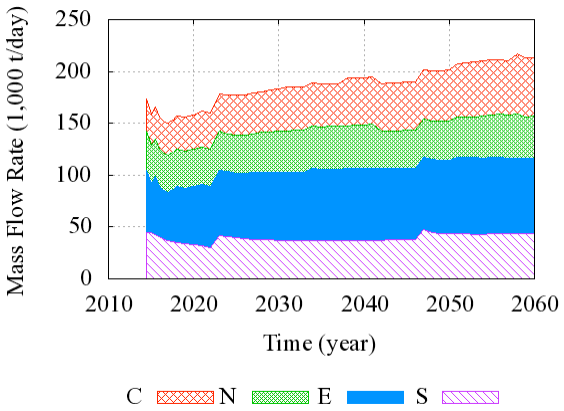

Figure 2. Mass flow rate computed by older code that does not support partial make-up well opening.

In recent years, the DMAK implementation has been modified to allow partial opening in order to meet demand precisely. Fig. 3 shows the result of using the newer code. A smoother increase in mass take can be observed, which was used to produce the precise steam flow rate shown previously in Fig. 1. This is achieved internally by scaling, and the scaling is configurable to be performed on either a single DMAK, or evenly on all opened DMAKs in the same group.

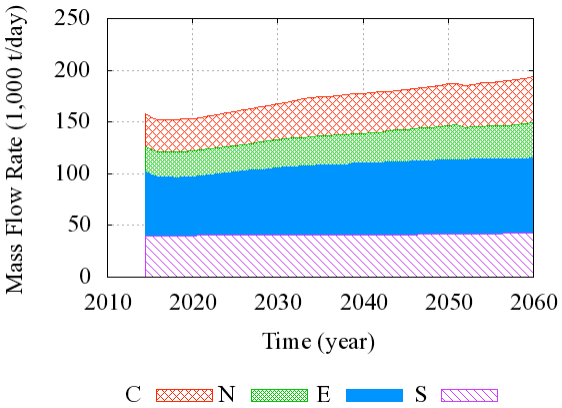

Figure 3. Mass flow rate computed by newer code. It is used for the rest of scenario results presented here. Note it is a lot smoother than Fig. 2.

The result shows some zones having to increase more mass take than others. This is caused by different enthalpy declines in each zone. The average enthalpy is shown in Fig. 4. In this scenario run, the DMAKs were configured to shut down if the enthalpy produced dropped below 850 kJ/kg. Production would then be taken from other DMAKs in the group. Many DMAKs from the North zone were shut down around the year 2050 due to low enthalpy. Production was shifted to other model blocks in the zone automatically. Note that it is possible to have production below target if productive model blocks in a zone are exhausted.

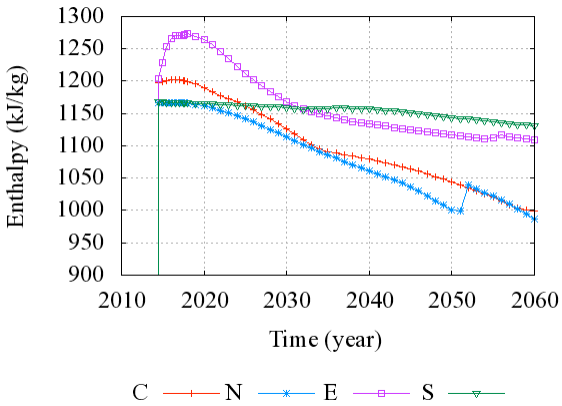

Figure 4. Average enthalpy from each of the production zones.

Fig. 5 shows the limited injection capacity of the South and East zones. Flows had to be redistributed to the Centre and North zones. The relatively small condensate injection is simply a fixed proportion (60%) of the total steam amount, which is not shown here.

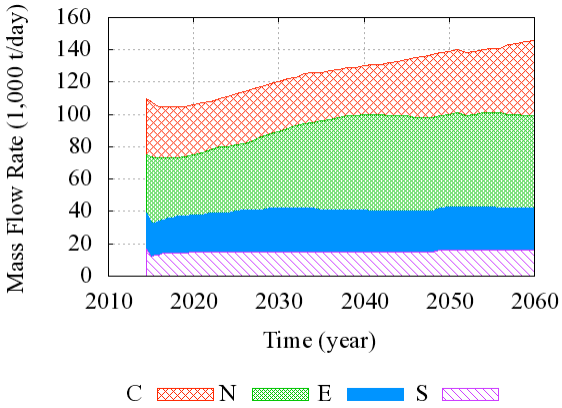

Figure 5. SGW injection rate into each zone. East and South zones are limited in capacity, and excessive SGW was distributed to North and Centre zones.

GRAPHICAL INTERFACE

One of the first software developments we made in association with use of TOUGH2 was the MULgraph graphical interface (Bullivant et al., 1995, Bullivant and O'Sullivan, 1998, O'Sullivan and Bullivant, 1995). We continue to use MULgraph to help with setting up TOUGH2 data files and with visualizing results from simulations, but we have more recently developed the PyTOUGH scripting library (Croucher, 2011; Wellmann et al., 2011) and are using it to assist with model building and visualization. For some of our models, we also use the Leapfrog Geothermal visualization software (Newson, 2012), and have been involved in adding TOUGH2 integration to it, which also makes use of the PyTOUGH library.

The flexibility available with the loose coupling between AUTOUGH2 and MULgraph or PyTOUGH we have found to be essential for carrying out very complex modeling projects, such as our study of the Lihir geothermal system, in which a gold mine is being excavated in the geothermal field. We have used PyTOUGH to run a year-by-year sequence of simulations with a mine pit being excavated at the top of the model (O'Sullivan et al., 2011). This kind of model development would be difficult to achieve with a more tightly coupled graphical interface such as PetraSim (Alcott et al., 2006).

CONCLUSIONS

The changes we have made with AUTOUGH2 have improved the capability of TOUGH2 for geothermal simulation. Some of the most significant changes we have made, such as the introduction of conjugate gradient solvers, were quickly adopted in the LBNL version of TOUGH2.

Many of the modifications we have introduced were aimed at making modeling of real geothermal systems easier, and some of our generator types, such as DMAK/TMAK, are very effective in this regard.

More remains to be done with regard to modeling production histories and future scenarios of geothermal systems. Rather than further modification of generator types, a new "SCENARIO" module is probably required to allow greater flexibility in the specification of a schedule of production and injection wells.

Overall, our experiences with using and modifying TOUGH2 have been very positive. Its longevity is testimony to the skill and judgment of the original developer, Karsten Pruess. There are further improvements we would like to make to AUTOUGH2, some of which we are currently not sure how to achieve. Probably at the top of our priority list is to speed up the approach to steady state in air/water or water/CO_2 models. For example, our models of Ohaaki (Clearwater et al., 2012) and Wairakei-Tauhara (O'Sullivan et al., 2009) both have difficulty reaching a steady state—often the time step will not increase past 1.0E11–1.0E12 seconds, whereas for other models the time step may increase easily up to 1.0E15 seconds.

We have been sharing experiences in using TOUGH2 with colleagues at LBNL for more than 30 years and look forward to continuing this relationship for the next 30 years.

ACKNOWLEDGEMENTS

Thanks are due to all who have contributed to the development of AUTOUGH2, particularly David Bullivant and George Zyvoloski, and many others whose names are hidden deeply in the code.

REFERENCES

Alcott, A., D. Swenson, and B. Hardeman, Using PetraSim to create, execute, and post-process TOUGH2 models, *Proc. TOUGH Symposium 2006*, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2006.

Bullivant, D., Making MULKOM/TOUGH faster and easier to use, *Proc. TOUGH Workshop '90*, Lawrence Berkeley Laboratory, Berkeley, Calif., 1990.

Bullivant, D.P. and M.J. O'Sullivan, Graphics and TOUGH2, *Proc. TOUGH Workshop '98*, Berkeley, Calif., 1998.

Bullivant, D.P., M.J. O'Sullivan, and M.R. Blakeley, A graphical interface for a geothermal reservoir simulator, *Proc. World*

Geothermal Congress, Florence, Italy, p. 2971-2976, 1995.

Bullivant, D.P., M.J. O'Sullivan, and G.A. Zyvoloski, Enhancements of the MULKOM geothermal simulator, *Proc. 13th New Zealand Geothermal Workshop*, 1991.

Clearwater, E.K., M.J. O'Sullivan, K. Brockbank, and W.I. Mannington, Modelling the Ohaaki geothermal system, *Proc. TOUGH Symposium 2012*, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2012.

Croucher, A.E., PyTOUGH: a Python scripting library for automating TOUGH2 simulations, *Proc. 33rd New Zealand Geothermal Workshop*, Auckland, New Zealand, 2011.

Croucher, A.E. and M.J. O'Sullivan, Application of the computer code TOUGH2 to the simulation of supercritical conditions in geothermal systems, *Geothermics*, 37, 622-634, 2008.

Fridleifsson, G.O. and W.A. Elders, The Iceland Deep Drilling Project: A search for deep unconventional geothermal resources, *Geothermics*, 34(3), 269-285, 2005.

IFC, *A formulation of the thermodynamic properties of ordinary water substance,* International Formulation Committee, Düsseldorf, Germany, 1967.

Moridis, G.J. and K. Pruess, T2SOLV: An Enhanced Package of Solvers for the TOUGH2 Family of Reservoir Simulation Codes, *Geothermics,* 27(4), 415 - 444, 1998.

Newson, J.A., W. Mannington, F. Sepulveda, R. Lane, R. Pascoe, E.K. Clearwater, and M.J. O'Sullivan, Application of 3d Modelling and Visualization Software to Reservoir Simulation: Leapfrog Geothermal and TOUGH2, *Proc. Thirty-Seventh Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, Calif., 2012.

O'Sullivan, M.J. and D.P. Bullivant, A graphical interface for the TOUGH family of flow simulators, *Proc. TOUGH Workshop '95*, Berkeley, Calif., 1995.

O'Sullivan, M. J., J.P. O'Sullivan, A.E. Croucher, L. Stevens, M. Esberto, Modelling the evolution of a mine pit in a geothermal field at Lihir Island, Papua, New Guinea, *Proc. 33rd New Zealand Geothermal Workshop*, Auckland, New Zealand, 2011.

O'Sullivan, M.J. and A. Yeh, *Wairakei-Tauhara modelling report*, Uniservices and Department of Engineering Science, University of Auckland, 276 pp. (http://www.contactenergy.co.nz/web/pdf/environmental/P4ReservoirModellingReport.pdf), 2010.

O'Sullivan, M.J., A. Yeh and W.I. Mannington, A history of numerical modeling of the Wairakei geothermal field, *Geothermics*, 38, 155-168, 2009.

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

Wagner, W., J.R. Cooper, A. Dittman, J. Kijima, H.-J. Kretzschmar, A. Kruse, R. Mareš, K. Oguchi, H. Sato, I. Stöcker, O. Šifner, Y. Takaishi, I. Tanishita, J. Trübenbach, J. and Th. Willkommen, The IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, *ASME J. Eng. Gas Turbines and Power*, 122, 150-182, 2000.

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., A. Croucher, and M.J. O'Sullivan, Recent experiences with overcoming TOUGH2 memory and speed limits, *Proc. 33rd New Zealand Geothermal Workshop*, Auckland, New Zealand, 2011.