Proceedings World Geothermal Congress 2020+1
Reykjavik, Iceland, April - October 2021

Waiwera: A New Parallel Open-Source Geothermal Reservoir Simulator

Adrian Croucher[1], Michael O'Sullivan[1], John O'Sullivan[1], Angus Yeh[1], John Burnell[2] and Warwick Kissling[2]

[1]The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand

[2]GNS Science, Private Bag 30-368, Lower Hutt 5040, New Zealand

a.croucher@auckland.ac.nz

Keywords: geothermal energy, reservoir models, numerical modeling, flow simulator

ABSTRACT

Waiwera is a new multi-phase subsurface flow simulator designed particularly for modelling geothermal reservoirs. Based around the PETSc library, it features scalable parallel execution on multi-core desktop machines and compute clusters, enabling large models to be run efficiently. Revised algorithms for handling multi-phase flows (particularly regarding the gravity term, and phase transitions), together with the advanced parallel linear solvers and preconditioners made available by PETSc, give improved performance for geothermal simulations, particularly the convergence of natural state models. A new flexible system for handling complex arrangements of source terms facilitates the modelling of geothermal steamfields.

Waiwera also features a clearly structured object-oriented code-base, modern software development principles (e.g. unit testing), use of standardised efficient formats for input and output, and an open-source license.

In this paper we describe the capabilities of Waiwera and demonstrate its performance on benchmark problems and real geothermal reservoir models.

1. INTRODUCTION

Geothermal applications place particular demands on reservoir simulation software, calling for robust numerical methods capable of modelling high-temperature fluids that boil and condense during the simulation. Of the relatively few codes with this capability, the TOUGH2 simulator (Pruess et al., 1999) has been considered by many to be the de-facto industry standard for geothermal reservoir simulation. A fork of TOUGH2 developed at the University of Auckland, known as AUTOUGH2 (Yeh et al., 2012), added other features specifically designed for geothermal modelling, including improved phase transition behaviour and additional generator types.

To approximate pre-exploitation conditions, a geothermal model is usually first run to a steady-state (or "natural state") solution, by running it to a large simulated time (and time-step size). This is often the most difficult and time-consuming part of a geothermal reservoir simulation, as the simulator may fail to converge to a steady state. Some of AUTOUGH2's modifications improved its steady-state convergence behaviour, but slow convergence and failures still occur in some situations.

There has also been an increasing demand for larger and more complex geothermal models, concomitant with increases in available computing power. This places ever greater demands on the simulation software, particularly in terms of computational speed. For very large models, currently the only viable remaining route to increased speed is parallelisation. The TOUGH2-MP (Zhang et al., 2013) and more recently TOUGH3 (Jung et al., 2017) variants of TOUGH2 are parallelised, but lack the geothermal-specific features of AUTOUGH2.

Parallelisation of the AUTOUGH2 code would in principle be possible, but difficult. Both AUTOUGH2 and TOUGH2 are based on the earlier TOUGH (Pruess, 1987) and MULKOM (Pruess, 1983) simulators, and the code design has changed very little since the 1980s. The code itself is written largely in Fortran 77, is not clearly structured and takes almost no advantage of the newer features introduced into more modern versions of the Fortran language, making it difficult to modify and maintain. In addition, code optimised to run efficiently in parallel usually needs to be designed differently from efficient serial code, so adding parallelism to existing serial code is generally not a straightforward process. Some aspects of the TOUGH2 workflow are also problematic, such as the custom-formatted text-based input and output files, which require auxiliary software such as PyTOUGH (Wellmann et al., 2012) for automated pre- and post-processing and are not efficient for large-scale models.

We therefore took the decision to develop a completely new geothermal flow simulator, named "Waiwera" (which means "hot water" in the Māori language), building on our experience with developing and using the TOUGH2 and AUTOUGH2 codes. However, Waiwera starts afresh with a new parallel code-base, using modern structured, object-oriented code design and taking advantage of the features of the latest Fortran 2003/2008 language standards.

2. METHODS

2.1 Numerical formulation

Like TOUGH2, Waiwera solves time-dependent mass and energy conservation equations with multiple mass components. (e.g. water, CO_2). The flow domain is discretised using a finite volume method, solving in each mesh cell for the average thermodynamic state, which is represented by a small set of "primary variables" (one for each conservation equation being solved). These primary variables depend on the mass components present and on the phase conditions in the cell.

Time evolution of all the primary thermodynamic variables over the mesh is calculated by time stepping. Waiwera allows different numerical methods to be used for time stepping, with the default "backward Euler" method and the second-order multi-step "BDF2" methods currently implemented.

As in TOUGH2, gradients of pressure and temperature across the mesh cell faces are evaluated using a simple two-point approximation (i.e. differencing the adjacent cell values across each face). This approximation relies on the mesh satisfying the "orthogonality criterion", i.e. that the line joining any two cell centroids is perpendicular to the face between them (Croucher and O'Sullivan, 2013). The mass and energy phase fluxes through each face are "upstream weighted" for numerical stability.

2.2 PETSc library and parallelisation

Waiwera makes extensive use of the PETSc library (Balay et al., 2018), a widely-used, open-source, high-quality computational tool-kit providing ready-made parallelised building blocks for the implementation of large-scale simulation codes. The PETSc "DMPlex" data type is used for handling the underlying unstructured mesh, including "domain decomposition", in which the mesh is divided up and distributed amongst the different parallel processes. It also handles mesh input from a variety of file formats, and the creation of parallel vectors (e.g. for rock and fluid properties, and the solution vector) and matrices over the mesh. PETSc handles most of the inter-process parallel communication, via the Message Passing Interface (MPI).

Because the mass and energy conservation equations are non-linear, determining the updated primary variables at each time step must be carried out using a non-linear equation solver. Waiwera uses PETSc's "Scalable Non-linear Equation Solver" (SNES) for this purpose, using a parallelised Newton-Raphson iterative method. The Jacobian matrix for the Newton-Raphson method is currently also computed automatically by PETSc, using finite differencing.

At each iteration of the Newton-Raphson method, a large sparse system of linear equations must be solved, and for many models this takes up the largest part of the total computation time. Waiwera uses PETSc's suite of "KSP" parallelised iterative linear equation solvers and pre-conditioners, which includes options suitable for the very ill-conditioned linear systems often found in geothermal reservoir models, particularly in steady-state simulations.

2.3 Thermodynamics and equations of state

Waiwera implements the IAPWS-97 (Wagner et al., 2000) thermodynamic formulation for computing the properties of water and steam, as well as the older IFC-67 formulation (mainly for comparison with TOUGH2 results).

Equation of state (EOS) modules are currently included for isothermal water, non-isothermal water, and non-isothermal water plus a non-condensible gas (either air or CO_2). The EOS modules for water/air and water/ CO_2 are both derived from the same parent EOS for a general non-condensible gas, sharing as much code as possible to ensure maximum consistency.

All non-isothermal EOS modules use variable switching, so that the primary thermodynamic variables for water are pressure and temperature under single-phase conditions, but switch to pressure and vapour saturation under two-phase conditions (since pressure and temperature are no longer independent variables on the saturation line). For the non-condensible gas EOS modules, the third primary variable is always gas partial pressure. In all cases, the primary variables are also non-dimensionalised to improve numerical behaviour.

2.4 Phase transitions

As in TOUGH2, when a phase transition occurs in a particular cell, its trajectory in primary variable space is temporarily halted on the boundary between the old and new thermodynamic regions (e.g. liquid water and two-phase, during a boiling transition). The switched primary variables are then computed and the simulation proceeds in the new region. For example, for a cell containing liquid water, a boiling transition occurs if the pressure drops below the saturation pressure corresponding to the temperature in that cell. In this case the primary variables are reset to pressure and vapour saturation at a point on the saturation line.

However, there are various possibilities for the choice of pressure on the boundary. TOUGH2 uses the saturation pressure at the new temperature, but this sometimes leads to phase transition failures and hence reductions in time step size. Waiwera uses a modified scheme which interpolates between the old and new fluid states to find the boundary pressure (Croucher et al., 2017). Because the saturation line is non-linear, this requires a non-linear root finding algorithm. However, this has only a small computational cost and is often effective for avoiding expensive time-step reductions. A similar interpolation process is also used for other types of phase transitions (e.g. condensing).

2.5 Gravity term

The phase fluxes through the faces between adjacent cells contain a gravity term, found by multiplying the local gravitational acceleration g by the effective phase density on the face. TOUGH2 computes this face phase density using a simple arithmetic average of the phase densities in the two adjacent cells. This gives a discontinuous change in phase density at the face when there is a phase transition in either of the two cells, often resulting in poor numerical performance of the Newton-Raphson non-linear solver (which assumes that the mass and energy balance equations are differentiable functions of the primary variables).

AUTOUGH2 introduced a workaround for this problem which involves computing properties of phases which are not physically present (O'Sullivan et al., 2013). This has subsequently been adopted by some versions of TOUGH2 and by TOUGH3. For Waiwera, we considered the force balance over the two cells and derived a modified multi-phase gravity term, which uses a saturation-weighted average of the cell phase densities. This removes the discontinuity and gives improved phase transition performance without having to compute any additional phase properties.

2.6 Source terms

Simulating production scenarios in geothermal reservoirs often requires complex combinations of source and sink terms, some of which depend on reservoir conditions and/or each other. AUTOUGH2 added approximately 20 new generator types to accomplish this, for simulating different kinds of wells on deliverability, make-up wells and re-injection.

Waiwera uses a new modular approach, in which sources and sinks themselves are kept simple, but have their flow rates and enthalpies determined by separate "source controls". Multiple source controls can be chained together to simulate more complex behaviour. Source controls implemented so far include pre-specified tables of flow rates, enthalpies and other parameters with respect to time, deliverability, separators and limiters.

2.7 Modelling fractured media

Like TOUGH2, Waiwera uses the Multiple INteracting Continua (MINC) method for modelling flow in fractured media (Pruess and Narasimhan, 1985). Additional finite volume cells are effectively nested inside the original mesh cells to represent flow between fractures and the rock matrix in between them.

Waiwera carries out all MINC processing of the mesh internally at run time. This is performed in parallel so that the computational time required is kept small, even for large models. Instead of having to prepare a separate MINC mesh, as in TOUGH2, the user only needs to specify which parts of the mesh are to have MINC processing applied to them. Multiple MINC zones can be specified with different MINC parameters if desired.

2.8 Input and output

Waiwera uses standard file formats for input and output to facilitate pre- and post-processing via existing tools (e.g. scripting). The main input file is in JSON format, a flexible, hierarchical data interchange standard. This also makes it easy to extend the input format when new features are added to the software. The simulation mesh is stored in a separate file which may be in any of the mesh formats supported by PETSc (GMSH, ExodusII and others).

Output is written to two separate files, a results file and a log file. The results file contains the simulation results ("heavy data") and is in HDF5 format, which is designed for large, hierarchical scientific data sets. The log file contains diagnostic messages output during the simulation and is in YAML format, which can be easily parsed using standard scripting tools (e.g. Python).

2.9 Software development and availability

In developing Waiwera we endeavoured to follow software engineering best practice. The code is written in a structured way and is version controlled. A test-driven development process was followed, with unit testing integrated into the development workflow using a new Fortran 2003 unit testing library called Zofu, released as a separate open-source project (https://github.com/acroucher/zofu). Functional testing of the code as a whole is carried out using a suite of automated benchmark tests.

Waiwera is free software, released under the GNU LGPL license, and is available for download from http://waiwera.github.io.

3. RESULTS

Here we demonstrate the performance of Waiwera on a selection of benchmark test problems and a real geothermal reservoir model.

3.1 Model Intercomparison Study problem 4

This test problem, titled "Expanding two-phase system with drainage", is from the 1980 Geothermal Model Intercomparison Study (Molloy, 1981), which established a set of standard benchmark tests for geothermal flow simulators. It is a one-dimensional vertical column model, 2 km deep. The initial conditions are hydrostatic, but with a piecewise linear temperature profile, 10°C at the top, 290°C at the half-way point and 310°C at the bottom. The mesh has 20 equally-sized cells with those in the upper half given lower permeability and porosity (5 mD, 15%) than those in the lower half (100 mD, 25%). Mass is extracted at a constant rate (100 kg/s/km^2) from the bottom of the model over a 40-year production period.

As the title of the model suggests, a two-phase zone develops and expands during production. As a result, there is no analytical or semi-analytical solution available. We compare the Waiwera results with those from AUTOUGH2 and the S-Cubed results from the original study. Figures 1 and 2 show the pressures and liquid saturations at depth 1550m. All three simulators give very closely consistent results. At other depths (not shown here) the level of agreement was also very good. This shows that Waiwera is reproducing the correct behaviour of a small but moderately complex two-phase system.

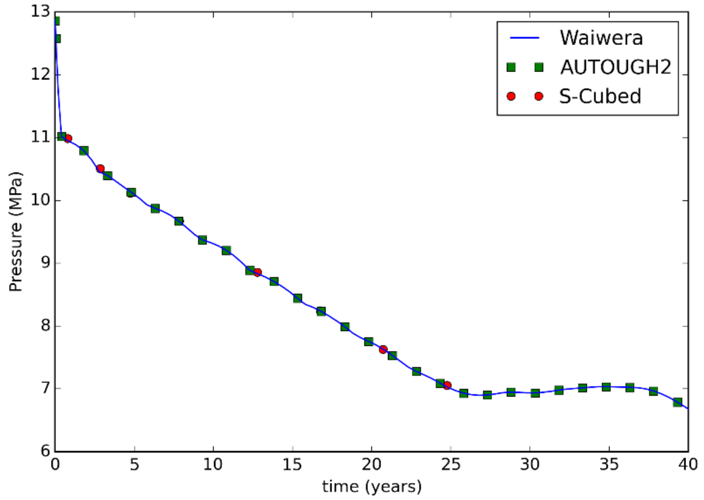

Figure 1: Pressure history at depth 1550 m for Model Intercomparison Study problem 4

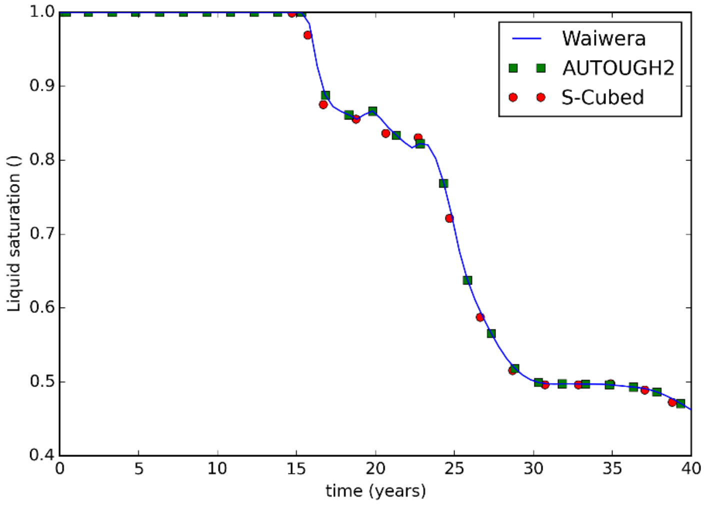

Figure 2: Liquid saturation history at depth 1550 m for Model Intercomparison Study problem 4

3.2 Steady-state boiling column problem

This is also a one-dimensional vertical column model with a two-phase zone. Here, however, the model mimics some of the basic behaviour of a natural-state geothermal model. It is run from cold hydrostatic initial conditions to steady state, and the steam zone develops as a result of hot water injected at the bottom.

The model mesh is 1 km deep and made up of 10 equally-sized cells, 100 m on each side, with atmospheric boundary conditions (1 bar, 20°C) applied at the top surface. The rock permeability is 100 mD throughout and linear relative permeability curves are used. Hot water at 240°C is injected at 10 kg/s into the base of the model.

Figure 3 compares Waiwera's steady-state convergence on this problem to that of TOUGH2 (v. 2.0), TOUGH3 (v. 1.0) and AUTOUGH2 (v.2.42). In each case the time step size history is plotted. From the initial time step size of 10^6 s, each simulator should ideally increase the time step size monotonically until a reliable steady state is achieved (generally when the time and time step size have reached at least ~10^{15} s). In practice, all four simulators have similar behaviour until the steam zone begins to develop at around 0.5×10^9 s. At this point, phase transition failures cause AUTOUGH2, TOUGH2 and TOUGH3 to reduce time step size twice. TOUGH2 subsequently stops completely as a result of other convergence problems, with the time step size only having reached around 10^{11} s. AUTOUGH2 and TOUGH3 however both recover once the steam zone has fully developed, and proceed to steady state in 36 time steps.

Waiwera does not reduce time step size during the development of the steam zone, and achieves steady state in only 31 time steps. Figure 4 shows the impact of disabling the primary variable interpolation scheme used in Waiwera for phase transitions (see section 2.4), and reverting to the simpler scheme used by AUTOUGH2. In this case, the time stepping behaviour is the same as that of AUTOUGH2, showing that Waiwera's improved steady-state convergence on this problem can be ascribed solely to this phase transition interpolation scheme (rather than other changes such as the modified multi-phase gravity term).

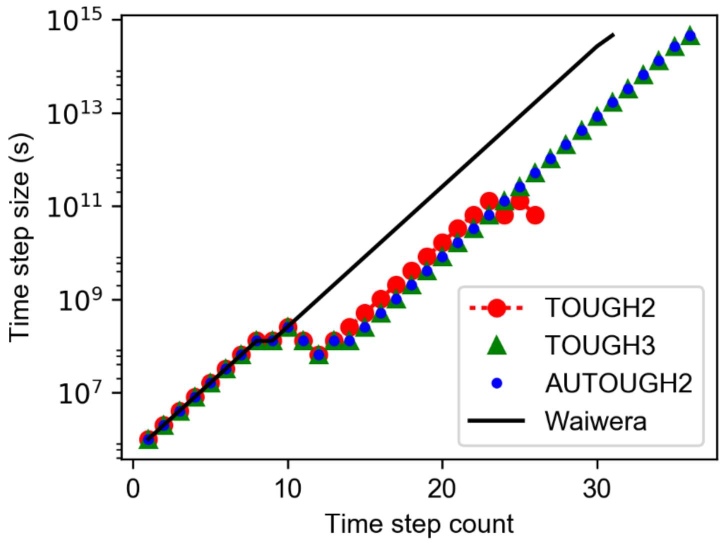

Figure 3: Steady state convergence for steady-state boiling column problem

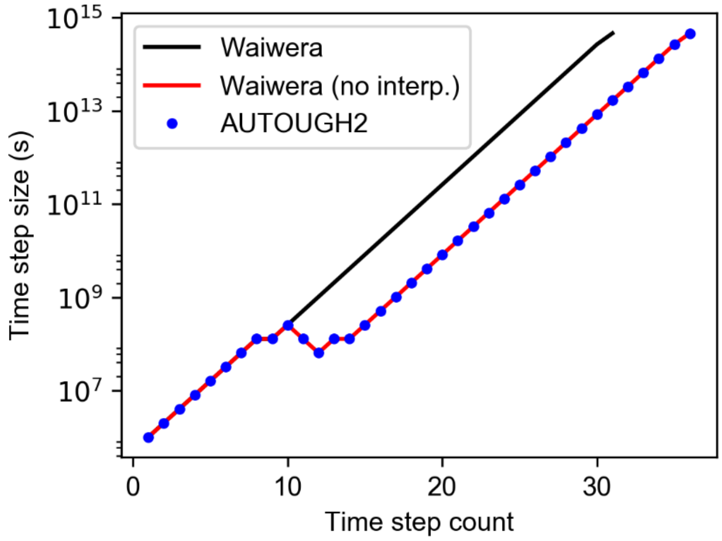

Figure 4: Impact of disabling phase transition interpolation scheme in steady-state boiling column problem

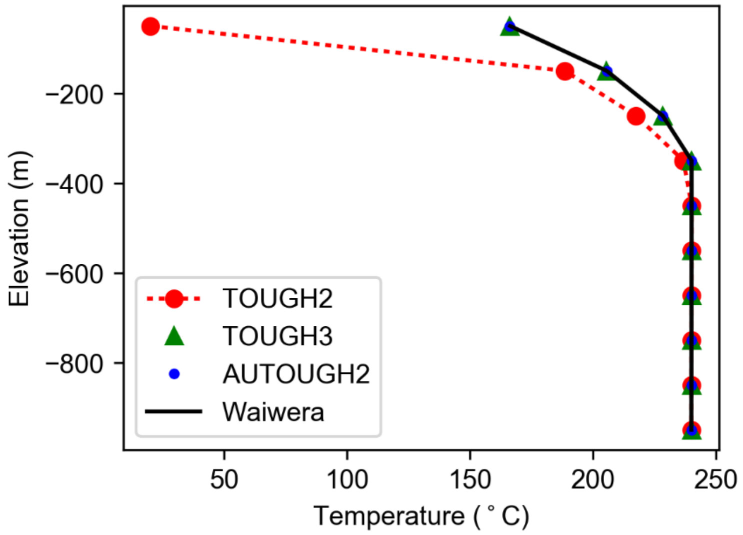

Figure 5: Final temperature profiles for steady-state boiling column problem

Figure 5 shows the final temperature profiles produced by the four simulators. Once again there is good agreement between Waiwera and the other simulators, except TOUGH2 which has clearly not run to a sufficiently large time step size to reach a steady-state solution.

3.3 Model Intercomparison Study problem 6

The sixth benchmark problem in the 1980 Model Intercomparison Study simulates production from a simplified 3-D geothermal reservoir model. The model mesh is 1.8 km deep and covers an area of 5 × 4 km. It is very coarse by today's standards, being made up of only 125 cells each measuring 1000 × 800 m in the horizontal, and with five layers in the vertical, each 300 m thick except for the bottom layer which has a thickness of 600 m.

The top and bottom layers are assigned lower permeabilities, while those in between (which also contain the steam zone) have higher permeability. Corey's curves are used for relative permeability. The initial conditions in each layer have prescribed temperatures (or saturations in the two-phase steam zone) and pressures which approximate hydrostatic conditions. Pressure and temperature boundary conditions are applied over the top and bottom of the model, and on one of the lateral boundaries. The single production well is located near one corner of the model, at a depth of 1050 m. The production rate is piecewise constant, increasing every two years. The time step size is fixed at 0.05 years, and the problem is run for 6.85 years. Boiling occurs in the well after about four years of production and after six years the pressure in the well drops to near zero.

The original study compared results from four simulators, but only three of them (from Lawrence Berkeley Laboratory, Geotrans and S-Cubed) achieved reasonable consensus. Figures 6 – 8 show the pressure, vapour saturation and enthalpy results in the well computed by Waiwera, together with the AUTOUGH2 results and the original results from LBL and S-Cubed for comparison. Again, there is good agreement between Waiwera and the other simulators.

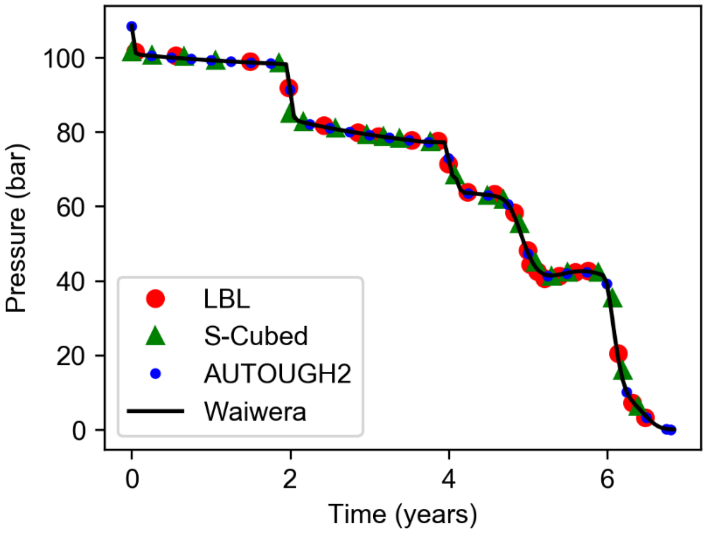

Figure 6: Production well pressures for Model Intercomparison Study problem 6

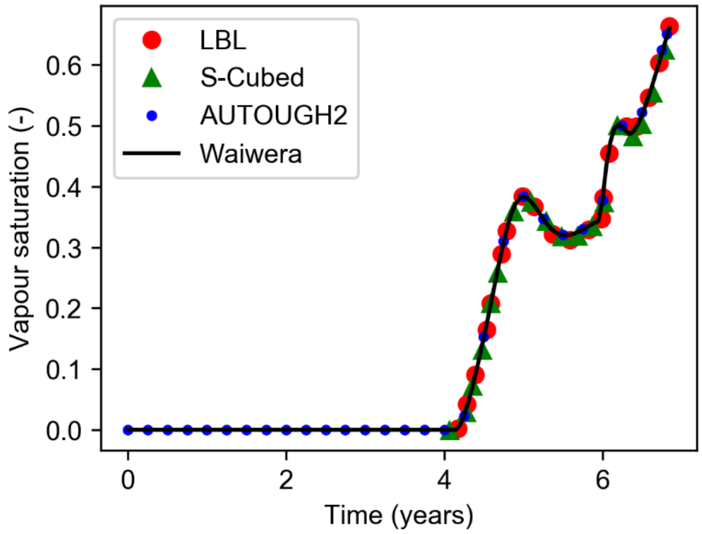

Figure 7: Production well vapour saturations for Model Intercomparison Study problem 6

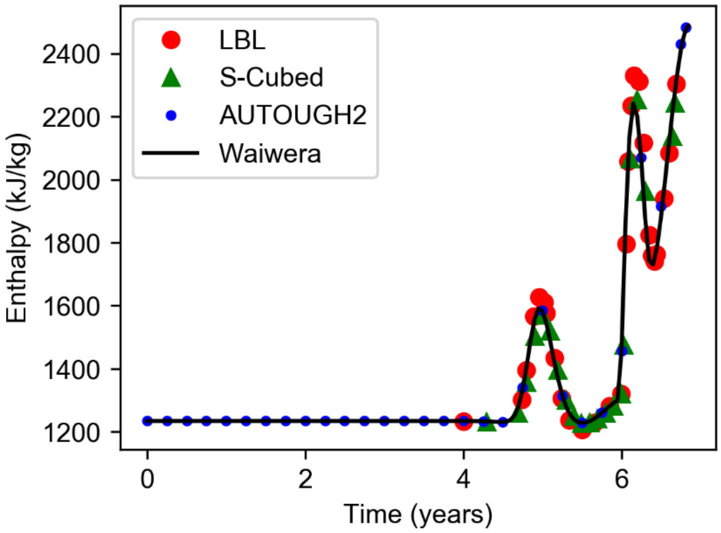

Figure 8: Production well enthalpies for Model Intercomparison Study problem 6

3.4 Parallel strong scaling study

We also created a refined model, loosely based on the original Model Intercomparison Study problem 6 model (see section 3.3), to demonstrate Waiwera's parallel scaling behaviour. A parallel "strong scaling" study involves running the same model on different numbers of parallel processes, and measuring the wall clock time taken in each case. The ideal behaviour is linear scaling, i.e. the computation time is inversely proportional to the number of parallel processes used. In this ideal case, for example, a doubling of the number of processes would result in a halving of the computation time.

In practice, scaling behaviour is usually less than ideal. It depends not only the details of the parallel implementation in the software, and on the parallel computing hardware being used, but also on the problem being solved. Satisfactory scaling can only be achieved if the problem is sufficiently large to benefit from parallelism. Otherwise, the overhead of the data communication between parallel processes may be too great, compared with the computational work being carried out on each process. The PETSc documentation recommends problem sizes of at least 10,000 – 20,000 unknowns per process for good parallel scaling. Even if the model is large enough for this condition to be met for smaller numbers of processes, as the number of processes is increased it will eventually be violated at some point. Hence, for a given model there is always an upper limit on the number of processes for which we can expect good scaling.

For a strong scaling study, it is also important to choose a model which has consistent behaviour as the number of processes is varied. Time-dependent flow simulations are typically used with adaptive time-stepping, in which the time step size is increased or decreased in response to the difficulty of solving the non-linear equations at each time step. This in turn depends on the performance of the linear equation solver used at each non-linear solution iteration. Iterative parallel linear equation solvers typically have performance that does vary depending on the number of processes. Hence, for a strong scaling study it is usually not possible to use adaptive time-stepping. This effectively rules out natural state models as candidates for strong scaling studies. For transient (e.g. production) models, it is necessary to use pre-determined time step sizes, and ensure these are small enough so that no time-step size reductions will occur as the model runs.

To create a model large enough to benefit from parallel execution, we refined the original Model Intercomparison Study problem 6 mesh from its original horizontal cell size of 1000 × 800 m down to 62.5 × 62.5 m, increased the depth of the model from 1.8 km to 3 km, and increased the number of layers from 5 to 160, with the upper 80 layers having thickness 12.5 m and the lower 80 layers being 25 m thick. This results in a mesh with 819,200 cells. As it is a pure water model, with two unknowns per cell, there are approximately 1.6 million total unknowns. Hence if the number of unknowns per process is to remain higher than 10,000 - 20,000 as recommended by PETSc, we cannot expect good scaling beyond about 40 - 80 parallel processes.

Since the object of the model was solely to investigate parallel strong scaling, it was not necessary, or indeed possible, to use the original model parameters. The original initial conditions were set up specifically for the 5-layer mesh and would be far from equilibrium for a more refined mesh. (In fact, they did not correspond exactly to an equilibrium solution even on the original mesh.) We decided to follow more usual practice and run the model to steady state, without production, to obtain an equilibrium initial condition for the production run.

We also altered the original rock properties and boundary conditions in order to give a natural state solution with a single upflow near one corner of the model. We used four rock types with parameters as shown as Table 1, and imposed an inflow condition on the bottom boundary. Water with enthalpy 1250 kJ/kg is injected into the bottom cells at three different flow rates, depending on horizontal position. Inside the square zone $x < 1.5$ km, $y < 1.5$ km, the flow rate is 2.4 kg/km^2/s, while outside the polygonal zone defined by the points (0, 0), (0, 3), (1.25, 3), (3, 1.25), (3, 0) km, the flow rate is 0.04 kg/km^2/s. In between these two zones the flow rate takes the intermediate value 0.64 kg/km^2/s.

Table 1: Rock properties for parallel scaling study model

Location	**Permeability (mD)**	**Porosity**
$-2000 < z < -200$ m	(2, 2, 1)	0.2
$-200 < z < 700$ m	(10, 10, 2)	0.25
$x < 1.5$ km, $y < 1.5$ km, $z > 700$ m	(0.2, 0.2, 2)	0.2
$x > 1.5$ km, $y > 1.5$ km, $z > 700$ m	(0.2, 0.2, 1)	0.2

With this level of mesh refinement it is no longer possible to concentrate all production in one cell, as in the original model. The production must be spread over multiple cells to avoid the pressure in the wells dropping to zero. We used 72 production wells distributed evenly over the production area $0 < x < 750$ m, $0 < y < 750$ m, $-100 < z < 0$ m. The total production rate was 24 kg/s at the beginning of the simulation, increasing to 60 kg/s after 2 years, 96 kg/s after 4 years and 144 kg/s after 6 years. The model was run for 10 years of production with a fixed time step size of 0.05 years.

The model was run on the New Zealand eScience Infrastructure (NeSI) "Māui" Cray XC50 supercomputer, which features Skylake Xeon nodes and Cray Aries interconnect in Dragonfly topology. It runs the Cray Linux Environment (CLE) operating system. We compiled Waiwera on Māui using GCC compilers and ran the model on 32, 64, 128 and 256 parallel processes.

Figure 9 shows a log-log plot of the computation time against the number of parallel processes, with the ideal linear scaling shown for comparison. It can be seen that Waiwera has near-ideal scaling on this problem up to around 100 processes, after which the computation times continue to decrease but at a slower rate, with greater costs exacted by the inter-process data communication. This scaling behaviour meets and exceeds the expected near-ideal scaling up to 40 – 80 processes suggested by the PETSc documentation.

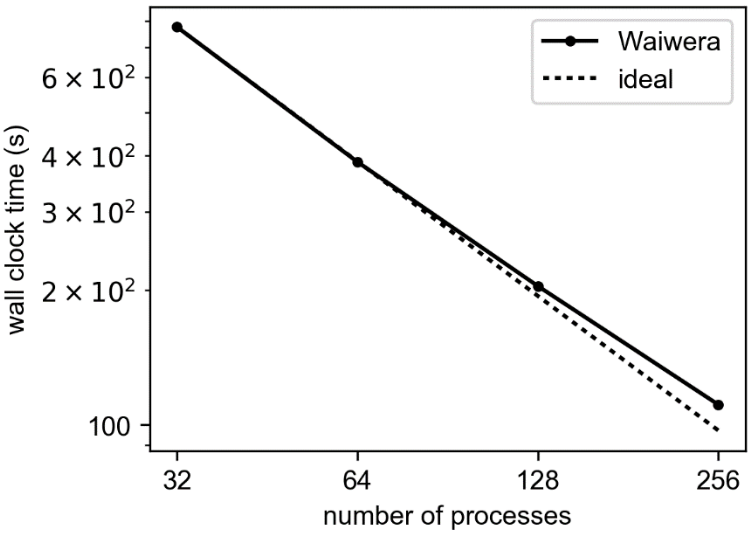

Figure 9: Computation time vs. number of processes for parallel strong scaling study

3.5 Ngawha geothermal reservoir model

The Ngawha geothermal field is located in the northernmost part of New Zealand's North Island, and is the country's only producing geothermal field outside the Taupo Volcanic Zone (Burnell et al., 2016). The Ngawha model used here was originally developed and run using AUTOUGH2. Its mesh has approximately 30,000 cells and it uses a water/CO_2 equation of state, because some of the wells produce approximately 1% mass fraction of CO_2. The top surface of the model is located at the estimated position of the water table. Figure 10 shows the model grid, together with the 200 °C temperature isosurface and the locations of the wells.

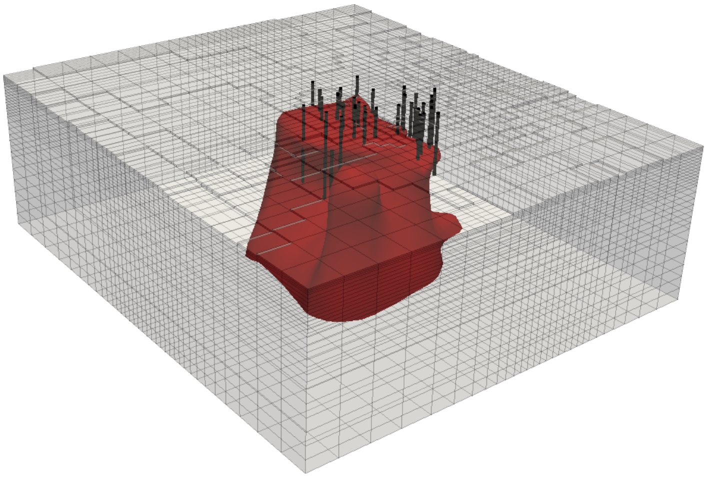

Figure 10: Ngawha model grid with 200 °C temperature isosurface and well locations

O'Sullivan et al. (2017) reported on the conversion of the model to run using Waiwera, and initial results from comparing the performance of a variety of linear solvers and pre-conditioners on the natural-state model. The Ngawha model was chosen as a suitable candidate for this partly because it performs relatively poorly on AUTOUGH2, providing a demanding test for a flow simulator.

Here we describe further experiments with the Ngawha natural-state model, this time designed to investigate the impact of different ways of non-dimensionalising the primary thermodynamic variables. As mentioned above, Waiwera solves for non-dimensionalised primary variables in each cell, to improve numerical behaviour (Croucher et al., 2017). This gives better choices of primary variable increments used in computing the Jacobian matrix via finite differencing, which can otherwise be problematic because the raw primary variables (pressures, temperatures etc.) typically have very different magnitudes.

The initial implementation of this non-dimensionalisation procedure simply applied a fixed scale factor to each variable (Croucher et al., 2017), so that all pressures (including non-condensible gas partial pressures) were scaled (i.e. divided) by a factor of 10^6, and temperatures by 10^2. Subsequently an "adaptive scaling" option was added, in which non-condensible gas partial pressures could be scaled not by a fixed constant but instead by the total pressure in the cell.

In the present work we used the Ngawha natural-state model to compare these two approaches to non-dimensionalising the gas partial pressures, and also investigated the impact of changing the scale factor used for pressure. As in the previous study, all simulations were carried out on a desktop PC with a single 12-core Intel Xeon E5-2670 processor, on which Waiwera was compiled using GCC compilers. The model was considered to have reached natural state when the simulated time reached 10^{15} s.

For reference, AUTOUGH2 was able to produce a converged natural-state solution, but only after 397 time steps and approximately 8.5 hours of run time on the same machine. (It was not possible to run this model on TOUGH3 because it includes geothermal-specific generator types, used to represent hot springs near the surface, which are not implemented in TOUGH3) The fastest Waiwera simulations in the previous study reached a natural-state solution in a little over 30 time steps and approximately 2 minutes run time. The Waiwera natural-state solution agrees closely with the AUTOUGH2 solution, with pressure differences within 0.9% and temperature differences within 0.1%.

To keep the number of cases manageable, here we limit the runs to 6 parallel processes, and use only the restarted GMRES(200) linear solver, which generally gave the best performance on this problem. As in the original study we use the Additive Schwarz (ASM) and Block Jacobi pre-conditioners. For each of these pre-conditioners we ran the model with both fixed gas partial pressure scaling and adaptive scaling, using pressure scale factors of 10^5, 10^6 and 10^7 in each case.

The results are shown below in Table 2. For each case the total number of time steps to reach natural state is shown, along with the run time, the number of time step reductions and the number of linear solver failures.

From these results it can be seen that the optimal choice of scaling method and parameters is dependent on the pre-conditioner used, even if the linear solver type and number of processes are held constant. When the ASM pre-conditioner is used, adaptive scaling performs better than fixed scaling on this problem, with best results using a pressure scale of 10^6. However, when the Block Jacobi pre-conditioner is used, the best results are obtained with fixed scaling, though 10^6 is again the optimal choice of pressure scale. In most cases (apart from ASM with fixed scaling) changing the pressure scale has a significant effect, with variations in run time of up to a factor of two, at least partly caused by changing the number of linear solver failures.

Table 2: Natural state convergence of Ngawha model with different pre-conditioners and primary variable scaling options

Pre-conditioner	Scaling	Pressure scale	Time steps	Run time (min)	Time step reductions	Linear solver failures
ASM	Fixed	10^5	47	11.5	8	6
		10^6	54	13.3	7	6
		10^7	45	10.2	7	5
	Adaptive	10^5	45	10.2	7	5
		10^6	45	6.9	6	2
		10^7	52	11.2	5	3
Block Jacobi	Fixed	10^5	38	5.0	2	1
		10^6	34	4.0	1	0
		10^7	47	8.8	5	4
	Adaptive	10^5	38	5.7	4	2
		10^6	48	8.2	8	5
		10^7	52	8.7	8	5

As a result, we now allow Waiwera users to select the scaling method and parameters in the model input. This gives users another option for improving the behaviour of natural-state and other models. Some experimentation will generally be needed to find optimum settings, as these may depend not only on the problem, but as we have seen, also on the pre-conditioner used, as well as the number of parallel processes and the linear solver type.

It should be remembered that although changing these parameters can have a noticeable impact on the model run time, the variations are small compared with the difference between the Waiwera run times and those needed for running AUTOUGH2 on the same problem (which ran up to 200 times slower). In other words, even if sub-optimal scaling or other options are used, the performance of Waiwera should generally still be significantly better than that of AUTOUGH2.

4. CONCLUSIONS

Waiwera is a new open-source geothermal flow simulator written in object-oriented Fortran 2003 and making extensive use of the PETSc computational library, ensuring parallel scalability for large models. It includes all features commonly needed for modelling geothermal reservoirs including robust phase transition behaviour, equations of state for non-condensible gases, flexible source and sink handling and simulation of fractured media via MINC.

We have demonstrated that Waiwera is capable of solving complex multi-phase benchmark test problems, with solutions in good agreement with previously published results and with the AUTOUGH2 simulator. Waiwera features enhanced natural state convergence behaviour, due partly to improved numerical formulations for the multi-phase gravity term and for phase transitions. We have also shown that Waiwera has near-ideal parallel strong scaling performance on a moderately large model (at least for the geothermal context) of approximately 1.6 million unknowns.

We have also demonstrated Waiwera's applicability to a real high-temperature geothermal reservoir model (Ngawha), and shown that its performance can be tuned for a given model via a range of options. These include the wide range of linear solvers and pre-conditioners made available by PETSc, as well as recently-introduced primary variable scaling options.

5. ACKNOWLEDGEMENTS

Waiwera was developed as part of the "Geothermal Supermodels" project (grant C05X1306) funded by the New Zealand Ministry of Business, Innovation and Employment (MBIE), with additional support from Contact Energy Ltd.

The authors wish to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities as part of this research. New Zealand's national facilities are provided by NeSI and funded jointly by NeSI's collaborator institutions and through MBIE's Research Infrastructure programme.

REFERENCES

Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W., Karpeyev, D., Kaushik, D., Knepley, M., May, D., Curfman McInnes, L., Mills, R., Munson, T., Rupp, K., Sanan, P., Smith, B., Zampini, S., Zhang, H. PETSc Users Manual, Revision 3.10. Technical Report ANL-95/11 Rev 3.10, Argonne National Laboratory, Illinois (2018).

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

Croucher, A.E. and O'Sullivan, M.J. Approaches to local grid refinement in TOUGH2 models. *Proceedings*, 35th New Zealand Geothermal Workshop, Rotorua, New Zealand (2013).

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. *Proceedings*, 39th New Zealand Geothermal Workshop, Rotorua, New Zealand (2017).

Jung, Y., Pau, G.S.H., Finsterle, S., Pollyea, R.M. TOUGH3: a new efficient version of the TOUGH suite of multiphase flow and transport simulators. *Computers and Geosciences*, **108**, (2017), 2-7.

Molloy, M.W. Geothermal reservoir engineering code comparison project. *Proceedings,* Special Panel on Geothermal Model Intercomparison Study, December 16-18, 1980, Stanford University, Stanford, California (1981).

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

O'Sullivan, J.P, Croucher, A.E., Yeh, A. and O'Sullivan, M.J. Experiments with Waiwera, a new geothermal simulator. *Proceedings,* 39th New Zealand Geothermal Workshop, Rotorua, New Zealand (2017).

Pruess, K. Development of the general purpose simulator MULKOM, Annual Report 1982. Technical Report LBL-15500, Lawrence Berkeley National Laboratory, University of California, Berkeley, California (1983).

Pruess, K. TOUGH User's Guide. Technical Report LBL-20700, Lawrence Berkeley National Laboratory, University of California, Berkeley, California (1987).

Pruess, K. and Narasimhan, T. A practical method for modeling fluid and heat flow in fractured porous media. *Society of Petroleum Engineers Journal,* **25**, (1985), 14 - 26.

Pruess, K., Oldenburg, C., Moridis, G. TOUGH2 User's Guide, Version 2.0. Technical Report LBNL-43134, Lawrence Berkeley National Laboratory, University of California, Berkeley, California (1999).

Wagner, W., Cooper, J.R., Dittman, A., Kijima, J., Kretzschmar, H.-J., Kruse, A., Mares, R., Oguchi, K., Sato, H., Stöcker, I., Sifner, O., Takaishi, Y., Tanishita, I., Trübenbach, J., Willkommen, Th. The IAPWS Industrial Formulation 1997 for the thermodynamic properties of water and steam. *ASME J. Eng. Gas Turbines Power*, **122**, (2000), 150–182.

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**, (2012), 197-206.

Yeh, A., Croucher, A., O'Sullivan, M. Recent developments in the AUTOUGH2 simulator. *Proceedings*, TOUGH Symposium 2012, Berkeley, California (2012).

Zhang, K., Wu, Y., Ding, C., Pruess, K. TOUGH2 MP: a parallel version of TOUGH2. *Proceedings*, TOUGH Symposium 2013, Berkeley, California (2013).