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

Volsung: A Comprehensive Software Package for Geothermal Reservoir Simulations

Peter Franz and Jonathon Clearwater

67 Hamon Place, Rotorua, New Zealand

Peter.Franz@flowstatesolutions.co.nz

Keywords: Reservoir modelling, simulation, Volsung, TOUGH2

ABSTRACT

This paper describes the development, testing and application of the new geothermal reservoir modelling software package *Volsung*. Volsung is commercially available and was developed to provide a single platform for all reservoir modelling needs. It includes a new reservoir simulator, a wellbore model, surface network model and a graphical user interface to simplify building, running and visualising models. Volsung is able to run fully coupled reservoir/wellbore/surface simulations which enables geothermal operators to use modelling tools for overall field and plant optimization. Benchmark speed testing shows Volsung to be up to an order of magnitude faster than conventional TOUGH2 with speedup enabled by parallel simulation on CPUs and GPUs. The key philosophy in Volsung is that all tools make use of internally consistent methods and that they are fast, stable and easy to use.

1. INTRODUCTION

Volsung consists of a reservoir simulator, a wellbore model, surface network model and a graphical user interface to simplify building, running and visualising models. The reservoir simulator simulates multi-phase, multi-component flow through porous and fractured media using a finite volume method (FVM) modelled on the very successful and popular TOUGH2 code. While TOUGH2 is a de-facto industry standard in the geothermal industry it is also known for not being very user friendly and lacking key features which are important for geothermal operators. The focus in the development of Volsung has hence been to implement what works from TOUGH2 and to add in new features as required for making better use of reservoir models in geothermal operations. Development will continue and will be driven by user demand for particular features.

In this paper we describe each component of the Volsung software package including results from benchmark testing applications of the software.

2. VOLSUNG SOFTWARE PACKAGE DESCRIPTION

When developing a new software package a good name is essential. We started by writing the reservoir simulator and wanted to name it after a fiery, powerful beast that lives underneath the ground - a dragon came to mind. So *Fafnir* - the name of the dragon from the *Volsunga Saga*[1] – was chosen. Since this saga is rich in diverse characters other modules and the main applications were named after them. In the following subsections we describe: *Fafnir* the reservoir simulator; *Brynhild* the graphical user interface (GUI); *Regin* the wellbore model and *Otter* the surface network simulator.

Before describing the individual components it is worth reflecting on the motivation for including a wellbore model and the capability to run coupled reservoir/wellbore/surface network simulations. When simulating production from a reservoir model the basic approach is to simply extract a certain amount of mass from a block coincident with a well's feedzone. However, without a wellbore model there is no check that the thermodynamic conditions in the block and productivity of the feedzone could actually enable this fluid to flow in a physically plausible way. This motivates the requirement for some type of check that there is sufficient pressure in the block; this functionality is often included in simulators such as the well-on-deliverability option in TOUGH2. But what about the case when a well has multiple feedzones? Take for example a situation when a well has one feed in a deep liquid zone and another feed in shallow steam zone. These two zones may evolve differently during a forecast scenario and the relative contribution from each feedzone may change drastically. If this process is not modelled properly then forecasts can become inconsistent with physics.

Another motivator for the coupled wellbore simulation approach is that operators typically want to forecast how much fuel is available at plant operating pressure. This is very difficult to answer accurately with a reservoir model that only forecasts reservoir conditions. It's important to use a wellbore model to link reservoir conditions to wellhead deliverability curves. This then enables a model to forecast exactly what operators need to know to enable more physically realistic forecasts of generation and make-up well requirements. Coupling a wellbore model to the reservoir model enables deliverability curves for mass flow and enthalpy to be calculated from reservoir conditions, it makes sense to then link these deliverability curves to power plant requirements via a surface network model. This has the advantage that the full feedback loop between reservoir conditions, fuel availability, plant requirements and injection load is accounted for.

The concept of coupling a reservoir model to a wellbore model has been previously investigated by other authors (e.g. Hadgu et al. 1995, Bhat et al. 2005, Butler et al. 2009, Ayala 2010, Gudmundsdottir 2012, Nandanwar & Anderson 2014, Franz 2016, Matsumoto 2018). Our experience has shown us that the key to successfully coupling these simulators is by having a robust wellbore simulator which can reliably find solutions under a wide range of thermodynamic conditions; if an exact solution can not be found it needs to make sensible simplifications or adjustments since it is impractical to hold a whole simulation and manually fix wellbore model

[1] e.g. https://www.gutenberg.org/files/1152/1152-h/1152-h.htm

issues. It has hence been our main motivation to develop the whole reservoir, wellbore and surface simulator as part of a single package in order to ensure that the different simulators can easily exchange data and are built on the same set of internally consistent tools.

3 THE GRAPHICAL USER INTERFACE (BRYNHILD)

In the Volsung reservoir modelling software Brynhild is the graphical user interface that enables users to develop reservoir models, run coupled reservoir-wellbore-surface network simulations and visualise the results. A key philosophy in Brynhild is that the conceptual model is developed independently of any particular grid. This massively simplifies the task of regridding a model or developing sector or process models of a particular sub-region of an existing model. Conceptual model features in the model are input as 3D shapes in Cartesian coordinates. An example of this is shown in Figure 1 where you can see the welltracks and conceptual model elements such as model regions and faults implemented without any reference to a particular grid structure.

The 3D shapes that define model regions and boundary conditions can be built from within the Brynhild GUI as points, lines, polygons and surfaces; alternatively they can be imported from GIS shapefiles or other 3D data formats such as exported LeapFrog or VTK files. This means users have the flexibility to develop conceptual models completely from within the Brynhild GUI or they can import a conceptual model developed in other GIS or earth modelling software.

In terms of grid generation the Brynhild GUI enables three types of grids: radial, rectilinear and Voronoi. Grids can be developed in local coordinate systems and then transformed into global coordinates to make it easier to combine with information from other coordinate systems such as well tracks and surface topography. The Voronoi grid generation is achieved by a triangle force relaxation method (Persson & Strang, 2004) followed by Delaunay triangulation and Voronoi cell identification. The Voronoi cells can be easily refined around welltracks and other points of interest.

Permeability is entered in the Brynhild GUI as tensor components in x, y, z directions along with options for rotation. When a particular grid is generated an appropriate permeability is calculated by projection of the tensor onto each connection vector between grid block centres. This means grids can be rotated but permeability anisotropy is maintained in the original direction, and anisotropy can be implemented in Voronoi grids. A full range of relative permeability and capillary pressure functions can be set from the GUI and these are associated with individual rock types. As such it is possible to have different relative permeabilities or capillary pressures within different lithological units.

Grid blocks in the model can be separated in fracture and matrix components and the matrix can be further refined into a flexible number of subdomains. This is equivalent to the MINC methodology used in TOUGH2 (Pruess, 1992). The number of MINC layers and the parameter values used to define the MINC formulation such as fracture volumes and fracture spacing are defined per lithological unit. In this way it is possible to have grid blocks in the reservoir with multiple MINC layers and grid blocks outside the reservoir with fewer MINC layers or even single porosity. Normally this capability would be used to more accurately model the heat transfer mechanisms in the reservoir where it matters without the additional computational cost of splitting “unimportant” grid blocks outside of the main reservoir into multiple subdomains.

Brynhild includes several features to make life easier as a modeller by simplifying the process of implementing hot plates, atmosphere blocks and other boundary conditions. Extra blocks can be added to the model where the user can manually specify the connection parameters.

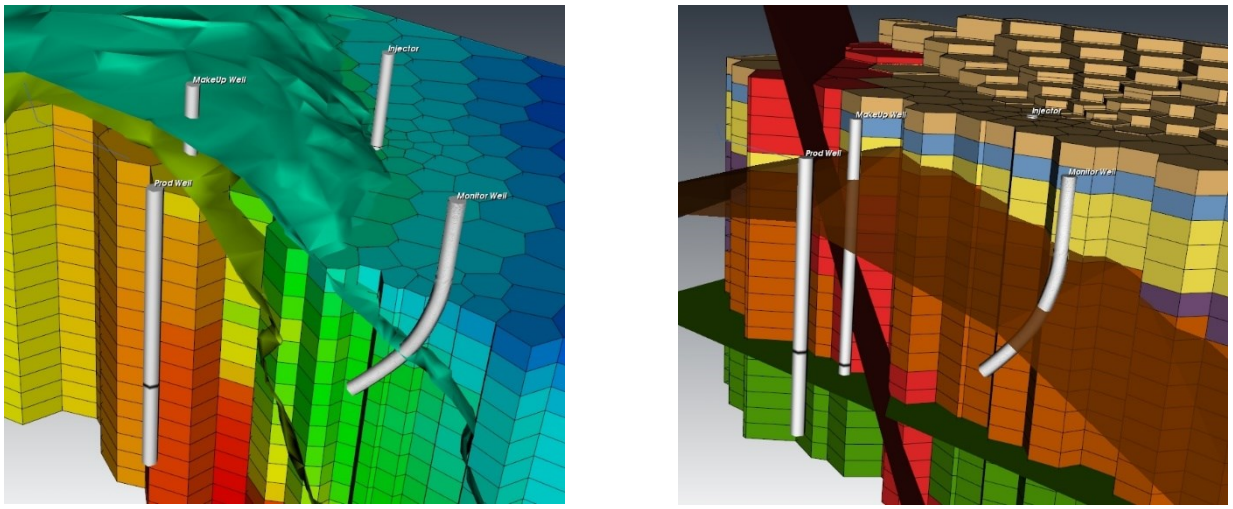


Figure 1: Visualisation showing cells coloured by temperature with isosurfaces (left). Conceptual model in Brynhild showing well-tracks and surfaces defining model regions and faults (right). One can see two conceptual layers (brown and green) which act as surfaces for filling the model with lithological units. The red slanted surface represents a fault, which overrides the lithological unit choice made by the layers.

Brynhild also serves as sophisticated viewer for model results. The model can be displayed in 3D coloured by temperature, pressure, saturation or any other thermodynamic property and tools are provided to cut and slice the model view. A slider bar enables model results to be displayed through time. Clicking on individual cells brings up full details of the individual cell properties including time series plots of that cell's thermodynamic condition over time and plots of heat and mass flow between neighbouring blocks. Isosurfaces and vectors can also be displayed. An additional feature in Brynhild is a module to calculate and display a change in microgravity over time. An example data visualization from Bryhild GUI is shown in Figure 1.

Last not least Brynhild enables running simulations on remotes, for example on leased cloud-based servers or shared high-performance workstations. Models can be sent to multiple remotes at the same time, for example for testing the influence of multiple parameters on simulation results; this feature can significantly reduce model calibration time.

4. THE RESERVOIR SIMULATOR (FAFNIR)

The Fafnir reservoir simulator simulates multi-phase, multi-component flow of mass and heat through porous and fractured media using the finite volume method. The mathematical approach and numerical methods used are very similar to the TOUGH2 methodology as described in Pruess (1999). The mass and energy balance equations solved for fluid and heat flow are described in Appendix A of Pruess (1999) and Section 4.2 of the *Volsung User Manual*[2] so do not need to be repeated here. Similarly the approach to space and time discretization is explained in Appendix B of Pruess (1999). Despite the similar approach Volsung has several key differences to TOUGH2 that enable significant performance improvements.

4.1. Performance

Methodological changes in Volsung allow for better overall performance versus TOUGH2. These changes include improved numerical accuracy in building the Jacobian matrix, backtracking the Newton step if it does not improve the intermediate solution, improvements in primary variable switching and removing of general bugs known to exist in TOUGH2. Work is in progress to use an artificial intelligence to predict better time stepping behaviour.

The main computational time required for the reservoir simulation can be split into two main tasks: calculation of thermodynamic states and solving a sparse linear system. Franz et al. (2019) describe the details on how the performance of these two computations can be improved. In short, the computation of thermodynamic states can be easily parallelized by employing multi-core CPUs or distributed memory systems.

Improving the performance for the linear solve operation is much harder. For small geothermal models, where the size of the linear system is smaller than the cache memory of a CPU, parallelized linear solvers like PETSc can improve the calculation time. However for large geothermal models the size of the linear system exceeds the cache memory; iterative solvers now need to cycle the linear system from RAM into the CPU once per iteration. This operation is bandwidth limited, with typical high-performing CPUs only achieving ~30-35GB/s bandwidth. The cutoff for this effect typically happens at a few thousand to ~20,000 elements, depending on the number of components in the model and the CPU used.

Two strategies can be used for overcoming the bandwidth limitation of the CPU. One solution is to employ computer clusters (distributed memory systems), as supported by the PETSc library and simulators based on it (e.g. Waiwera, TOUGH3). By splitting the linear solve operation to many CPUs one can add up their bandwidth, or when using a sufficient number of CPUs even relocate the system into the caches of the many-CPU system. The large drawback of this strategy is that access to a dedicated computer cluster is required, i.e. a performance improvement on a standard desktop computer is not possible for large models.

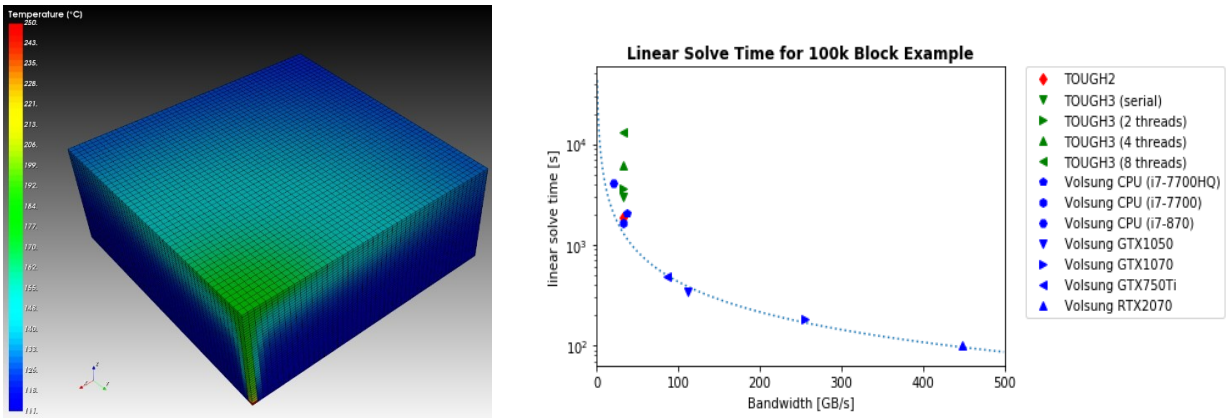


Figure 2: Model used in benchmark speed testing (left). The bottom block facing the viewer is held at high pressure and temperature; the far block in the top layer has a small sink attached. This view depicts the temperature after a 1 million year simulation time. The plot on the right shows the time spent for solving the linear system for this simulation using different simulators and architectures; the dotted line demonstrates the 1/bandwidth dependency for solving the linear system. Note the logarithmic y-axis.

[2] available from https://www.flowstatesolutions.co.nz/downloads or by contacting the authors

Volsung overcomes this problem by utilizing graphic processing units (GPUs) for solving large linear systems. GPUs have much higher memory bandwidths (500-1000GB/s) than CPUs and are available for desktop and laptop computers. GPUs have been successfully used for solving linear systems and are a fundamental part in today's neural network research and development.

We tested the performance of the Fafnir simulator versus TOUGH2 and TOUGH3. We chose a 100,000 block model (50x50x40 blocks) with block dimensions 20x20x10 metres, set to a cold initial state (300bar, 20ºC). One bottom corner block was set as a fixed state with high temperature and pressure (500bar, 250ºC); across the diagonal we produced 1kg/s fluid from a block in the top layer. The model used single porosity, 50mD permeability, specific heat of 1000J/kg, rock density 2650kg/m^3, heat conductivity of 2J/(K*m) and 1% porosity; Figure 2 shows a depiction of model. The simulation was run for 1 million years with no constraints on the time step. This model was chosen for its simplicity, i.e. it doesn't have phase changes. Also we wanted to demonstrate what's happening when the linear system for the model exceeds the available CPU cache; with a size of roughly 64MB this model will not fit into any common CPU caches (~25MB for Intel Xeon CPUs, or typically 8MB for Intel i7).

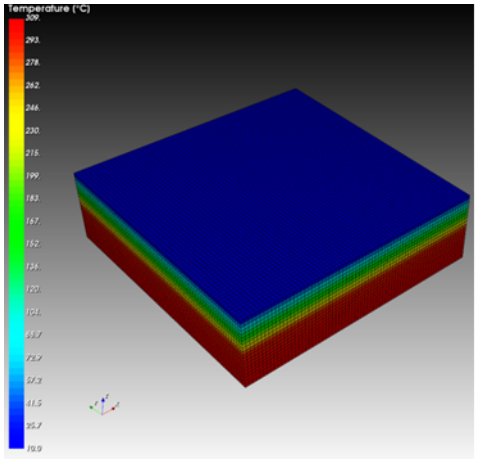
Figure 2 shows the linear solve times versus the bandwidth of the CPU or GPU used. Volsung solved the model in 93 steps; one can clearly see the 1/bandwidth dependency of the solve time in the chart; the best solve time of 99.5s was obtained using a consumer grade RTX2070. TOUGH2 took 162 steps and had a model run time of 1883s (since TOUGH2 does not provide output for the linear solve time we used the total run time as an estimate). TOUGH3 took 845 steps and spent between 2969s (serial) to 13,233s (8 threads) for solving linear systems.

The above results indicate multiple problems for comparing the numbers: Volsung clearly solved the system in less steps and hence required less time per linear solve in the first place, plus it had the advantage of increased bandwidth, leading to the best overall run times. TOUGH2 took more steps for the solution but was still competitive as compared to the CPU solver of Volsung since steps with smaller time steps tend to require less iterations in the iterative solver.

The behaviour of TOUGH3 is a bit puzzling; it looks like running it on more threads actually increases the run time - the number of 845 steps did not change between these runs. We assume that this behaviour is due to preconditioner degradation with increased number of threads, as one can expect when using block-style preconditioners. This means the iterative solver requires more iterations to achieve the same desired precision and hence increases the linear solve time.

Testing on the 100k Block Example showed that Volsung was more stable at larger timestep sizes and was able to reach simulation completion time in fewer timesteps. In a natural state run a user would get the benefit of increased stability at larger timestep sizes, but in a production run the timestep may be constrained due to the need to simulate particular well flow rates or transient processes in the reservoir. Accordingly, a series of production history tests were run with a restricted timestep size. The Stanford Problem 4 was used as a base for this, but instead of a column model it was converted into a rectilinear grid with varying degrees of spatial discretization. Three model grids were tested with dimensions 30x30x20 (18,000 gridblocks), 50x50x20 (50,000 gridblocks) and 70x70x20 (98,000 gridblocks). The 98,000 gridblock model is shown in Figure 3.

Because timestep size was restricted in the production simulations this test provides a comparison of how quickly the simulators calculate each timestep. Because a similar model was run with various levels of spatial discretization it is possible to see how this difference in run time scales with model size. In the 18,000 block model Volsung took 50s and TOUGH2 took 145s for a speedup factor of about 3. In the 98,000 block model Volsung took 179s and TOUGH2 took 1232s for a speedup factor of about 7. These are sensible speedups considering the parallelization of the thermodynamic routines which scale with processor count – a quad core i7 was used for this test – plus some speedup in the linear solve for the larger models which exceed the CPU cache. The runtime results for the models are summarised in the table shown in figure 3.



	Blocks	Volsung on GTX1070		TOUGH2		Speedup
		Timesteps	Runtime (s)	Timesteps	Runtime (s)	
Prod.Model	18,000	301	50	339	145	3
Prod.Model	50,000	301	99	300	452	5
Prod.Model	98,000	301	179	300	1232	7

Figure 3: The model used to simulate a production history model run with time step constraint. The left picture shows the model with 98,000 gridblocks resolution. The table shows run times and time steps required for running this model with different spatial discretization and the speedup achieved by Volsung.

4.2. Validation

TOUGH2 is the *de facto* industry standard for geothermal reservoir modelling and it has been widely used and tested. Accordingly it provides an appropriate simulation code to benchmark Volsung against. To facilitate the benchmark testing, functionality was built into Volsung so it would read a TOUGH2 input file and run the simulation using Volsung's reservoir simulator Fafnir.

Nine test problems were selected for accuracy testing and results from each model were compared between Volsung and TOUGH2. Six of these test cases were taken from the Stanford Geothermal Code Comparison Study (1980). Another test case was a version of the five-spot problem from the TOUGH2 manual modified to have 900 grid blocks vs the original 36 (Pruess, 1999). The other two cases were new with one representing a simple case with two blocks equilibrating while the transient reponse was monitored and the other case was a MINC model simulating a production and injection scenario. These test problems covered a large range of simulation conditions including transient responses to production, phase change, three dimensional reservoir flow, steam/liquid counter flow and fracture-matrix (MINC) heat and mass flow. In all cases identical models were run using TOUGH2 and Volsung and in all cases Volsung gave results in extremely close agreement with TOUGH2.

Since the tests are described in detail in the technical report we will not reproduce full details of each model; figure 4 shows some results from the Stanford #1 test problem which demonstrates the agreement between Volsung and TOUGH2. The full results are published in Franz et al. (2019) or are available online as part of the Volsung examples and in the Volsung benchmarking report.[3]

5. THE WELLBORE SIMULATOR (REGIN)

Volsung's wellbore simulator *Regin* features twice in the package: Once as fully integrated simulator between the reservoir and surface models, and once in a standalone GUI called *Gudrun* which can be used to model individual wells without the need to run a fully coupled simulation. Regin features both production and injection modes.

The wellbore simulator is based on solving the ordinary differential equations for mass and heat along a wellbore (e.g. Hasan & Kabir 2002). The integration is carried out using either a simple Euler or a more advanced Runge-Kutta (4th order) method. Fluid is exchanged at feedzones only; linear and quadratic (Forchheimer) draw equations are supported.

The flow path configuration offers the feature to have time-dependent segments present; this is of interest in running coupled simulation since it avoids having to create two or more separate well representations if the flow path has changed, for example by sleeving the well. It also offers the possibility to add known variations in flowpath diameter to the simulation, like scaling or cleanouts.

The wellbore simulator can be run using pure water, H_2O and NCG, or H_2O and salt equations of state; in the later case currently salt precipitation does not lead to wellbore scaling but this effect could be included in a later version. Another EOS with H_2O, salt and NCG is planned.

Pressure drop correlations available include homogeneous, Duns & Ros (1963), Hasan & Kabir (2010) and Hagedorn & Brown (1965). Further correlations similar to the ones used for GWELL/GWNACL/HOLA - Chisholm/Orkiszewski and Chisholm/Armand - have been implemented for comparison purposes with the GWELL (Aunzo et al. 1991) family of wellbore simulators. Lastly, the Orkiszewski (1967) pressure drop correlations are implemented but found unsatisfactory for practical use due to known numerical issues (discontinuities). The Gudrun GUI features import mechanisms for wellbore models from the GWELL and SimGWel (Marquez et al. 2015) wellbore simulators.

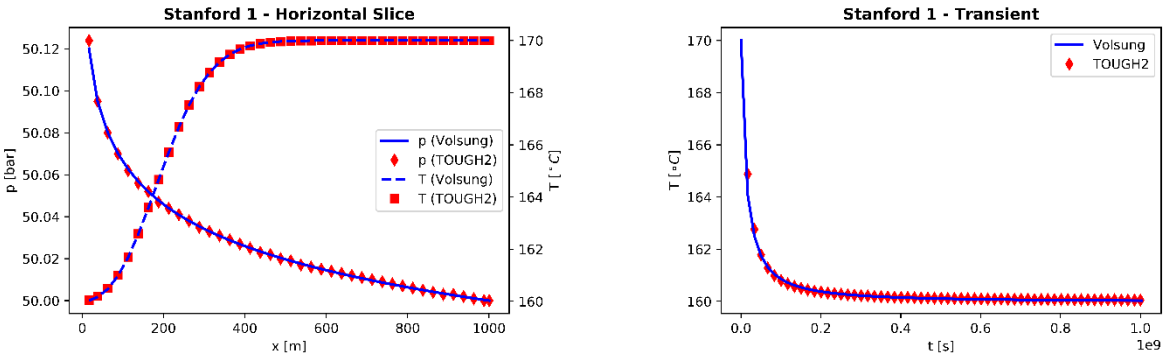


Figure 4: Comparison between TOUGH2 and Volsung results for the Stanford Test Problem #1. The left plot shows a comparison along a horizontal slice in the model, the right hand side plot depicts a temperature transient. The agreement for this model between the two simulators is excellent.

[3] www.flowstatesolutions.co.nz/downloads

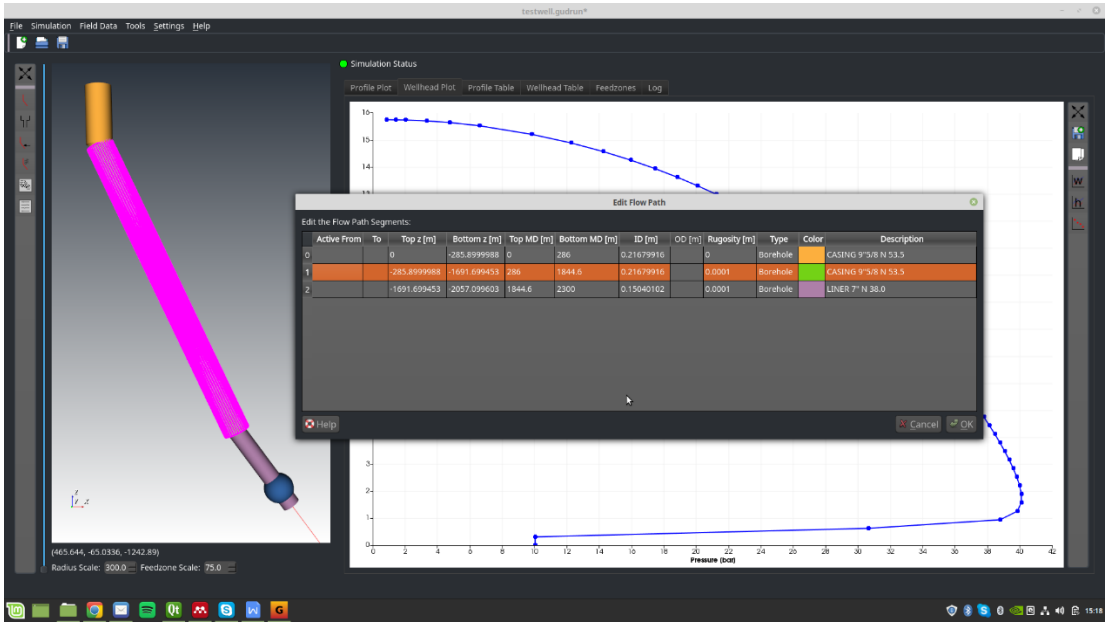


Figure 5: The Gudrun standalone GUI for the Regin wellbore simulator. The screen depicts a deviated well and its flow path configuration; a deliverability curve is shown in the background behind the dialog.

Several different simulation modes are available. The most common ones allow the modeller to simulate single wellbore conditions - for example when comparing to PTS surveys - or more complex modes for determining the wellhead characteristics (deliverability/injectivity curves) and searches based on them. Numerical stability in these searches was a paramount design criterion; the wellbore simulator aims to make sensible choices if it can't find a solution. For example weak feedzones below prolific feedzones can be automatically eliminated if their inclusion leads to an empty solution set; of course this behaviour is optional and appropriate warnings are given.

6. THE SURFACE NETWORK SIMULATOR (OTTER)

6.1. Principle of Operation

The surface network *Otter* in Volsung tracks pressure, mass flow and enthalpy from production wells through surface network features such as pipes, valves, separators, heat exchangers and power plants and then allocates the appropriate amount of flow to injection wells. Power plants can regulate the amount of produced fluid to meet their generation target; if the capacity of the producers is too small then they can automatically switch on predefined make-up wells.

Objects are placed into the surface network in an editor. Every object has input and output ports which can be connected to each other to form the network topology; each port provides a unique flow rate consisting of mass rate, dry energy/electrical rate, enthalpy, component mass fractions and pressure. Depending on its type an object calculates an output port response to its input port conditions. Some objects offer free parameters - for example mass rates for throttling a producing well - to the surface simulator. Others, like power plants, also provide an objective function to indicate a mismatch to a desired state.

The solution for the system is found iteratively. A basic iteration propagates a trial solution from upstream objects (e.g. producing wells) to downstream objects; each downstream object can then determine its current solution. Once all downstream objects have calculated their states the direction is turned around and pressure change suggestions are propagated upstream. If after such a basic iteration the pressures in the objects are not matched then the state is declared invalid. If a state is valid its objective function is calculated by summing up the weighted objective functions of the surface objects, e.g. the power plants.

During history matching runs only the basic iteration is performed; this is helpful to determine if parameters in the surface network are representative for the system and can be used in the scenario/forecasting mode.

In the scenario mode a simulated annealing type optimization algorithm is used to vary free parameters and optimize the objective function until its global optimum has been found. The algorithm for this process is a Temperature Parallel Simulated Annealing with Adaptive Neighborhood (TPSAAN) algorithm (Miki et al. 2002). The simulation is started by creating parallel clones of the surface network; each clone then works on a fixed simulated annealing temperature and tries to optimize the objective function. At intervals the clones exchange their solutions; better solutions move towards colder temperatures and vice versa. This results in a broad probing of the parameter space yet fast convergence on a good solution. Since the clones can be run in parallel this method performs very well.

The surface network model has several modes available to run production history and forecast simulations. During production history well flow is specified in flow rate tables for each well and typically the surface network does not take control of the simulation.

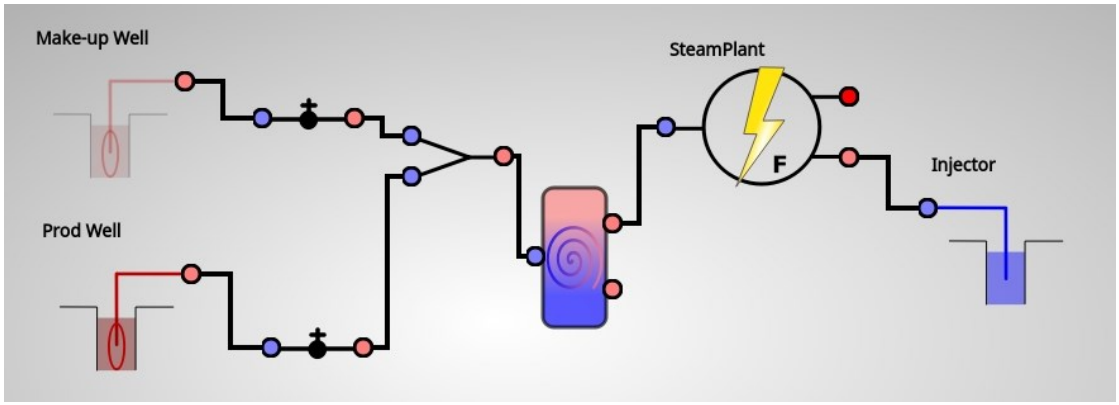


Figure 6: Example surface network in Volsung. Objects are placed on a canvas and can be repositioned with the mouse. Output ports (red) can be connected to input ports (blue) to setup the model's network topology. Each object has its own internal set of rules and equations how to determine its output flow rate from the input flow rates.

During forecasts wells can have a specified flow or they can fluctuate according to demand from the plant. An ordered list of make-up wells can also be specified. If fuel availability from current wells is not sufficient to meet a plant generation target then the next make-up well on the list is switched on. An example surface network from Volsung is shown in Figure 6. The example model shows a production well connected through a valve and a separator to a steam plant and the condensate from the plant is injected. A predefined makeup well is present which can be switched online by the steam plant once the need arises. The power generated and the flow rates required for this simple demonstration are shown in figure 7.

6.2. Available Objects

Due to its principle of encapsulating the objects in a surface network the simulator is easily extendable, i.e. new objects with different internal rules or equations for calculating the output port flow rates can be added with a minimal amount of code. Objects implemented so far are surface sources, unions, valves, splitters, injectors, heat exchangers, separators and a power plant model. Further, Otter includes several options for modelling wells including fixed mass fraction wells, wellbore coupled wells and pumped wells:

6.2.1. Fixed mass fraction wells

For fixed mass fraction wells mass is simply extracted from the gridblock coincident with the wells feedzone(s); each feedzone is allocated a fixed fraction of the total flow at wellhead. Importantly the well still requires a well track and feedzone elevations; this ensures that the specification of the well is independent of any particular grid.

6.2.2 Wellbore coupled wells

Wellbore coupled wells make use of the Regin wellbore model as described above. Thermodynamic conditions of the gridblock at model feedzones are passed to Regin to model the flow up the well and pass wellhead fuel availability information to the surface network simulator. When the appropriate flow from the well has been determined by the surface network simulator fluid is extracted from the well's feedzones consistent with feedzone contribution as determined by the wellbore model.

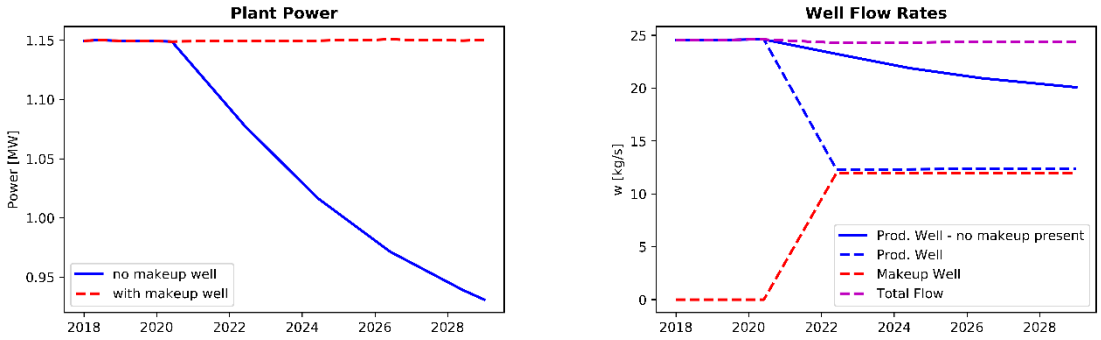


Figure 7: Simulation results for the surface network example: power generated by the plant (left) and fluid produced by the wells (left). Without a makeup well the production well can not supply the plant and power generation tails off (solid line). If the plant is allowed to bring the makeup well online then it can maintain its generation target; the production fluid required is distributed over the two available wells (dashed lines).

6.2.3. Pumped wells

Including pumped wells in Volsung enables systems to be analysed were the enthalpy and pressure in the reservoir is insufficient to support flow to surface without pump support. Pressure drawdown from a reservoir pressure P_{res} to a wellbore pressure P_{well} is described by:

$$w = \frac{\rho}{\mu} \frac{2\pi(kh)}{ln\left(\frac{r_{res}}{r_{well}} + s\right)} (P_{res} - P_{well})$$

In a numerical simulation the block pressure P_{res} is known but the unknown well pressure P_{well} is what is needed to determine whether or not a well will flow. Variables ρ, μ, kh and r_{well} are known in a simulation but it is not clear what the appropriate r_{res} for a given grid block given the spatial discretization of a numerical model. To establish the appropriate r_{res} for given gridblock size a comparison was made between the analytic expression and the block radius from a simple numerical radial model. In both the analytic model and the radial numerical model reservoir properties and thermodynamic conditions were identical. Production was simulated from a well of radius 0.1m with numerical radial grid spacings of 10m, 100m and 500m. Comparing analytic results with numerical model results it became clear that the appropriate radius to use was 0.67 of the grid block radius in a radial model; this was consistent with each grid spacing tested.

By using an appropriate r_{res} then P_{well} can be obtained from above equation and this can be used to determine whether or not a well will flow. Each pumped well is specified with a desired flow rate, specified wellhead pressure, pump maximum flow rate, pump differential pressure and pump efficiency. Using the wellbore parameters (welltrack, radius and roughness) and fluid properties the gravitational and frictional pressure drop from wellhead to feedzone is calculated. If P_{well} plus the pumps differential pressure is greater than the desired pressure at the wellhead then the well is capable of producing at the desired rate.

7. TRANSFERRING THE TIWI RESERVOIR MODEL

As part of the validation process for Volsung we transferred the existing reservoir model of the Tiwi geothermal field into the software. The Tiwi field is operated by Philippine Geothermal Production Company, Inc. (PGPC) and started commercial production in 1979. Tiwi had some two-phase regions in its natural state and developed extensive single phase steam zones during the production phase. At the time of writing we have successfully translated the natural state model into Volsung; work on the production history and scenario phase is currently ongoing.

A complicating factor in the translation to Volsung was that the existing model used a modified TOUGH2 version. The key differences were that the PGPC model provided used permeabilities and porosities as 3D fields instead of discrete rock types and that it contained modified connection areas for blocking/enhancing faults.

We solved the issue of 3D field variations by first writing a customized version of PyTOUGH[4] and then used it to analyze the TOUGH2 model and create the required number of "lithological units" to replicate the property variations. We then translated the grid into Volsung and used a file-based block association model where each grid block is associated with one of the defined lithological units; this approach works with other systems as well, for example with LeapFrog models.

The modified connection areas were also analyzed and translated into barrier representations; Volsung uses these barriers to modify permeability and heat conductivity of connections between individual blocks, which can increase model resolution to contain sub-grid features.

Once the model had been redefined in a Volsung compatible format, the necessary boundary conditions (heat and mass inflows and outflows) were also defined and we ran the natural state of the Tiwi model twice; first using initial conditions provided by PGPC, which were close to steady state, and then using a cold liquid state as the starting condition. We then compared pressure, temperature and gas saturation between the TOUGH2 and Volsung steady states by plotting the block-by-block differences in histograms; these are shown in figure 8. Pressure differences were all smaller than ~0.12bar and temperature differences were generally smaller than 0.7ºC. Given the fact that Volsung uses IAPWS-97IF steam tables versus IC67 used by TOUGH2 the two simulators show excellent agreement.

Gas saturation differences were all smaller than 0.005 with the exception of a single block which showed a difference of 0.05. Closer inspection showed that this block was located next to a region with ~0.05 gas saturation and had remained single phase. This behaviour is well understood in the context of a non-unique solution due to the relative permeability function used - a Grant curve with 0.05 residual saturation, i.e. the system is free to deviate by this amount.

Finally we compared the results from the cold start with the above results and only found maximum relative numerical deviations in the order ~10^{-7}, which is close to the single precision truncation error of ~10^{-8} for the file format in which the data is stored, i.e. the results can be seen to be nearly identical.

[4] https://github.com/acroucher/PyTOUGH

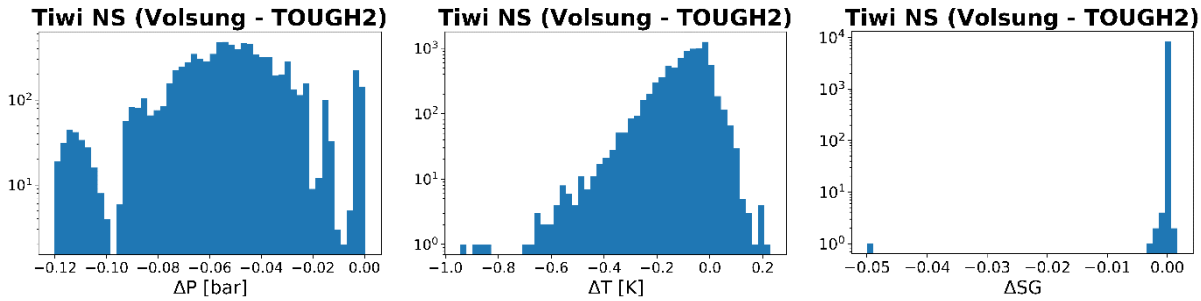


Figure 8: Histograms showing the block-by-block difference between Volsung and TOUGH2 results for the Tiwi natural state run. From left to right: Pressure, temperature and gas saturation differences. The -0.05 gas saturation difference in a single block is due to non-uniqueness of the solution for the Grant relative permeability curve.

8. SUMMARY

Features and functionality of the Volsung reservoir simulation package have been described and the underlying motivation for developing the software has been discussed. Benchmark testing of the reservoir simulation code shows that Volsung is consistent with TOUGH2 across a broad range of numerical model test cases. Performance testing shows Volsung to be significantly faster than TOUGH2 for a wide set of problems. Some aspects of the improved performance are attributed to improved methodologies and some are attributed to calculating the thermodynamic state and solving the linear system in parallel. Testing showed that the relative speed-up was proportional to 1/bandwidth of the CPU or GPU used in the simulation when the size of the linear system was too large to fit into the cache memory of the CPU. Most full field models would be too large to fit into a typical desktop computer cache, justifying the use of GPUs due to their advantages of high memory bandwidth at low cost relative to CPUs.

Validating Volsung with benchmark tests highlighted the need for some new standards with which to compare various codes. The Stanford code comparison study (DOE, 1980) was seminal work, but by the time this paper is published it will be 40 years old. This is an eternity in current day computing so we feel it is now time to add additional benchmark tests to standardize code validation and provide a level playing field for speed comparison testing. The same can be said about wellbore model benchmarks which are even scarcer to find.

We would like to thank PGPC for allowing us to publish the results from the Tiwi model translation study. We hope to continue collaboration with PGPC and other interested parties and are looking forward to move more geothermal reservoir models into the Volsung framework.

REFERENCES

Aunzo, Z. P., Bjornsson, G., & Bodvarsson, G. S. (1991). Wellbore Models GWELL, GWNACL, and HOLA User's Guide. LBL-31428

Ayala, M. A. R. (2010). Coupled geothermal reservoir-wellbore simulation with a case study for the Namafjall fiel, N-Iceland, Thesis for Magister Scientiarum degree in Mechanical Engineering, University of Iceland

Butler, S. J., & Enedy, S. L. (2009). Numerical Reservoir-Wellbore-Pipeline Simulation Model of The Geysers Geothermal Field, California, (April), 25–29. https://doi.org/10.2118/121385-ms

Bhat, A., Swenson, D., & Gosavi, S. (2005). Coupling the HOLA Wellbore Simulator with TOUGH2. Proceedings, 30th Workshop on Geothermal Reservoir Engineering, Stanford University

Department of Energy. (1980): Proceedings Special Panel on Geothermal Model Intercomparison Study. Technical Report SGP-TR-42, Stanford

Duns, H., & Ros, N. C. J. (1963): Vertical flow of gas and liquid mixtures in wells. 6th World Petroleum Congress, 451–465. World Petroleum Congress

Franz, P. (2016). MataTauira: A Growing Software Package for Numerical Geothermal Reservoir Simulations. Proceedings 38th New Zealand Geothermal Workshop

Franz, P., Clearwater, J., Burnell, J. (2019): Introducing the Volsung Geothermal Simulator: Benchmarking and Performance, Proceedings 41st New Zealand Geothermal Workshop

Gudmundsdottir, H., Jonsson, M. T., & Palsson, H. (2012). Coupling Wellbore Simulator with Reservoir Simulator. Proceedings of the 37th Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford

Hadgu, T, Zimmerman, R., Bodvarsson G (1995): Coupled reservoir-wellbore simulation of geothermal reservoir behavior. Geothermics

Hasan, A. R., & Kabir, C. S. (2010). Modeling two-phase fluid and heat flows in geothermal wells. Journal of Petroleum Science and Engineering, 71(1–2), 77–86

Hagedorn, A. R., & Brown, K. E. (1965): Experimental Study of Pressure Gradients Occurring During Continuous Two-Phase Flow in Small-Diameter Vertical Conduits. Journal of Petroleum Technology, 17(4)

Marquez, S. L., Sazon, T. A. S., & Omagbon, J. B. (2015). SIMGWEL: EDC's New Geothermal Wellbore Modeling Software. World Geothermal Congress 2015

Matsumoto, M. (2018). Connecting Wellbore and Reservoir Simulation Models Seamlessly Using a Highly Refined Grid. Proceedings, 43rd Workshop on Geothermal Reservoir Engineering Stanford University, Stanford, (1), 1–8

Miki, M., Hiroyasu, T., Kasai, M., Ono, K., & Jitta, T. (2002). Temperature parallel simulated annealing with adaptive neighborhood for continuous optimization problem. Second International Workshop on Intelligent Systems Design and Application, 149–154

Nandanwar, M. S., & Anderson, B. J. (2014). Coupled Reservoir, Wellbore and Surface Plant Simulations for Enhanced Geothermal Systems. Proceedings of 39th Stanford Geothermal Workshop, (1), 1–10

Orkiszewski, J. (1967). Predicting Two-Phase Pressure Drops in Vertical Pipe. Journal of Petroleum Technology, 19(6), 829-838

Persson, P.-O., & Strang, G. (2004): A Simple Mesh Generator in MATLAB. SIAM Review, 46(2), 329–345

Pruess, K. (1992): Brief Guide to the MINC-Method for Modelling Flow and Transport in Fractured Media, Lawrence Berkeley Laboratory

Pruess, K. (1999): Tough2 User Guide, Lawrence Berkeley National Laboratory