PROCEEDINGS, Thirty-Seventh Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, January 30 - February 1, 2012
SGP-TR-194

MODELING LIQUID DOMINATED TWO PHASE FLOW IN GEOTHERMAL RESERVOIRS IN VICINITY TO, AND INSIDE WELLS

Larus THORVALDSSON
Halldor PALSSON

University of Iceland
Hjardarhagi 2-6
107 Reykjavik, Iceland
lth31@hi.is

ABSTRACT

This paper describes a new modeling approach of two phase fluid flow in geothermal reservoirs. The flow is assumed to be liquid dominated with vapor traveling alongside the liquid if present, which is common for hydrothermal systems. The problem is governed by the Darcy-Forchheimer equation for fluid transport, along with a convective-diffusive energy equation. A special case arises inside wells, where the Forchheimer part of the flow equation becomes dominant and is quantified by using well known relations for fluid flow in pipes. The problem is formulated and solved by using a highly customizable set of C++ libraries and tools, collectively named OpenFOAM, along with the IAPWS-IF97 standard for the properties of steam and liquid water.

Preliminary results from the modeling work are presented in simple case studies, showing the basic abilities of the programming platform to solve large three dimensional problems for flow in porous media. It is concluded that the modeling framework is very promising, since it is under constant improvement by a large group of developers and incorporates cutting edge technologies in numerical analysis for mathematical modeling.

INTRODUCTION

Using numerical methods to solve non-linear partial differential equations (PDE) first became feasible in the late 1960's with the advent of digital computers. These methods were first applied to problems involving groundwater as well as oil and gas reservoirs, while the modelling of geothermal reservoirs lagged behind [1]. This was mostly due to the fact that the modelling of geothermal reservoirs is considerably more complicated where the equations are typically of the advection-diffusion type, describing conservation of mass, momentum and thermal energy. These equations are furthermore coupled with each other and are frequently non-linear, which adds considerably to the complexity of their solutions.

The earliest efforts to apply numerical models to geothermal reservoirs were made in the early 1970's, while the usefulness of numerical modeling did not begin to gain acceptance by the geothermal industry until after the 1980 Code Comparison Study [2]. Since that study was performed, the experiences gained in carrying out site-specific studies as well as generic reservoir modeling studies have led to a constant improvement in the capabilities of numerical reservoir models.

Over the last 20 years computer modeling of geothermal reservoirs using finite volume methods has become a standard practice. The most common approach is to apply the TOUGH2 code, developed by the Earth Sciences Division of Lawrence Berkeley National Laboratory in the 1980's. TOUGH2 is a general numerical simulation code for multi-dimensional coupled fluid and heat flows of multiphase multicomponent fluid mixtures in porous and fractured media [3]. Numerous case studies have been made using the TOUGH2 code, modelling geothermal reservoirs in countries such as Iceland [4 ,5 ,6], New Zealand [7 ,8 ,9], Japan [10], Russia [11], P.R. of China [12], Nicaragua [13], Ethiopia [14] and the Philippines [15].

In the current work the problem is formulated and solved by using a highly customizable set of C++ libraries and tools, collectively named OpenFOAM, along with the IAPWS-IF97 standard for the properties of steam and water. The object orientation and operator overloading of C++ has enabled the developers of OpenFOAM to build a framework for computational fluid dynamics that enables modelers to work at a very high level of abstraction [16]. This makes it possible to manipulate the set of partial differential equations that describe the problem and customize the solver itself for each class of cases that needs to be solved. This is the main motivation for using OpenFOAM, rather than currently existing models, such as TOUGH2.

Another reason is to enhance the use of state of the art methods of mathematical modeling of geothermal reservoirs. This includes applying the current leading numerical methods, such as algebraic multigrid [17], which creates a hierarchy of progressively coarser linear equation sets by restricting the fine matrix by agglomeration or filtering [18]. Since the solution of large linear systems is a key operation in solving PDE's, the improvement of such algorithms is vital in reducing simulation time and enabling the use of larger and more accurate models. To date, these methods are considerably faster than more classical iterative methods based on conjugate gradients or direct sparse solvers, frequently by a ratio of two to five.

Since the source code for OpenFOAM is open and freely available, new codes can easily be developed and linked into existing or new solvers for PDE's. In this paper the IAPWS-IF97 thermodynamic formulation, which has superseded the older and more computationally intensive IFC-67 formulation, has been implemented in C++ and connected directly into the implemented solver for the reservoir problem. This allows for considerable improvements in speed and accuracy, where the IAPWS-IF97 is more than five times faster than IFC-67, except in the supercritical region where it is approximately three times faster [19]. The standard implementation of TOUGH2 currently uses IFC-67, however there have been some recent efforts to modify the code to make use of IAPWS-IF97, especially in the supercritical region [20].

The paper is organized as follows. In the first section the basic set of equations for reservoir dynamics are presented, which form the basis for an customized implementation in the OpenFOAM framework. A part of this is the specific implementation of the IAPWS-IF97 standard. Then three illustrative case studies are presented as results. Finally conclusions are drawn and further work outlined.

METHODS AND MATERIALS

In this section the governing equations for two phase flow in porous media are presented in the form they are implemented in a numerical model. This involves the equations themselves, fluid properties, boundary conditions and then the programming implementation itself.

Continuity equation

In order to model two phase flow in porous media it must be assumed that mass and energy are conserved. The continuity equation describes mass conservation and is given such that

$$\frac{\partial}{\partial t}(\varphi\rho) + \nabla\cdot(\rho\vec{u}) = 0$$

where ϕ is porosity, ρ is density and $\vec{u}$ is superficial velocity. If the density is only a function of pressure *p* and enthalpy *h* the time derivative in the continuity equation can be expanded which gives

$$\varphi\frac{\partial\rho}{\partial p}\frac{\partial p}{\partial t} + \varphi\frac{\partial\rho}{\partial h}\frac{\partial h}{\partial t} + \rho\frac{\partial\varphi}{\partial t} + \nabla\cdot(\rho\vec{u}) = 0$$

Furthermore, if the inertial forces are negligible compared with viscous forces, as is the case in most hydrogeological systems, Darcy's law can be applied to the equation above

$$\vec{u} = -\frac{\kappa}{\mu}(\nabla p - \rho\vec{g})$$

which gives

$$\varphi\frac{\partial\rho}{\partial p}\frac{\partial p}{\partial t} + \varphi\frac{\partial\rho}{\partial h}\frac{\partial h}{\partial t} + \rho\frac{\partial\varphi}{\partial t} = \nabla\cdot\left(\frac{\rho\kappa}{\mu}(\nabla p - \rho\vec{g})\right)$$

The continuity equation is then a time dependent diffusion equation with source terms, which amongst other are functions of the energy state of the system, measured with enthalpy h. In many cases, it would be sufficient to assume that the time scales are large relative to the dynamics of the equation, which would be enforced by solving a steady state equation only, giving the pressure distribution as a result.

Energy equation

The energy equation includes both effects from the fluid and the solid, which are determined by the porosity ϕ. In the presence of temperature gradients, conduction takes place both in the fluid and the solid. If the combined thermal conductivity is denoted by Γ the energy equation then becomes

$$\frac{\partial}{\partial t}(\varphi\rho h + (1-\varphi)\rho_s c_s T) + \nabla\cdot(\rho\vec{u}h) = \nabla\cdot(\Gamma\nabla T)$$

where c_s is the specific heat of the solid and ρ_s its density. Since the temperature can be determined from known values of *p* and *h* the time derivative in the equation above can be written as

$$\begin{aligned}\frac{\partial}{\partial t}(\varphi\rho h + (1-\varphi)\rho_s c_s T) = {} & (\rho h - \rho_s c_s T)\frac{\partial\varphi}{\partial t} + (1-\varphi)\rho_s T\frac{\partial c_s}{\partial t} + (1-\varphi)c_s T\frac{\partial\rho_s}{\partial t} \\ & + \left(\varphi h\frac{\partial\rho}{\partial p} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial p}\right)\frac{\partial p}{\partial t} \\ & + \left(\varphi\rho + \varphi h\frac{\partial\rho}{\partial h} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial h}\right)\frac{\partial h}{\partial t}\end{aligned}$$

The spatial term for the temperature diffusion can also be expanded in terms of pressure *p* and enthalpy *h*, which gives

$$\nabla T = \frac{\partial T}{\partial p}\nabla p + \frac{\partial T}{\partial h}\nabla h$$

Combining the relevant terms in the energy equation yields

$$\left(\varphi h\frac{\partial \rho}{\partial p} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial p}\right)\frac{\partial p}{\partial t} + \left(\varphi\rho + \varphi h\frac{\partial \rho}{\partial h} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial h}\right)\frac{\partial h}{\partial t}$$
$$+ (\rho h - \rho_s c_s T)\frac{\partial \varphi}{\partial t} + (1-\varphi)\rho_s T\frac{\partial c_s}{\partial t} + (1-\varphi)c_s T\frac{\partial \rho_s}{\partial t} + \nabla\cdot(\rho\vec{u}h)$$
$$= \nabla\cdot\left(\Gamma\frac{\partial T}{\partial h}\nabla h + \Gamma\frac{\partial T}{\partial p}\nabla p\right)$$

This equation can be further simplified if it is assumed that the parameters ϕ, c_s and ρ_s do not vary in time and Darcy's law can be applied to the velocity giving

$$\left(\varphi h\frac{\partial \rho}{\partial p} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial p}\right)\frac{\partial p}{\partial t} + \left(\varphi\rho + \varphi h\frac{\partial \rho}{\partial h} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial h}\right)\frac{\partial h}{\partial t}$$
$$= \nabla\cdot\left(\frac{\rho\kappa h}{\mu}(\nabla p - \rho\vec{g})\right) + \nabla\cdot\left(\Gamma\frac{\partial T}{\partial h}\nabla h + \Gamma\frac{\partial T}{\partial p}\nabla p\right)$$

The equation above represents an unsteady advection-diffusion problem with respect to *h*, with source terms which are partially dependent on *p*.

System of equations

It was seen when applying the two laws in the preceding sections that they are mutually dependent, through *h* and *p*. Coupling the continuity equation and the energy equation together then finally yields a system of partial differential equations

$$\begin{bmatrix} \varphi\frac{\partial\rho}{\partial p} & \varphi\frac{\partial\rho}{\partial h} \\ \varphi h\frac{\partial\rho}{\partial p} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial p} & \varphi\rho + \varphi h\frac{\partial\rho}{\partial h} + (1-\varphi)\rho_s c_s\frac{\partial T}{\partial h} \end{bmatrix}\begin{bmatrix} \frac{\partial p}{\partial t} \\ \frac{\partial h}{\partial t} \end{bmatrix}$$
$$= \begin{bmatrix} \nabla\cdot\left(\frac{\rho\kappa}{\mu}(\nabla p - \rho\vec{g})\right) \\ \nabla\cdot\left(\frac{\rho\kappa h}{\mu}(\nabla p - \rho\vec{g})\right) + \nabla\cdot\left(\Gamma\frac{\partial T}{\partial h}\nabla h + \Gamma\frac{\partial T}{\partial p}\nabla p\right) \end{bmatrix}$$

which is rather complicated and should be set up and solved numerically. Note that it might be preferable to partition the spatial derivatives into advective terms, diffusive terms and source terms, especially when implementing the equations in a numerical code.

IAPWS-IF97 thermodynamic formulation

In most applications the IFC-67 thermodynamic formulation has now been superseded by the IAPWS-97 formulation. Its current revision consists of a set of equations which cover the following range of validity

$$273.15\,\mathrm{K} \le T \le 1073.15\,\mathrm{K} \quad \text{for} \quad p \le 100\,\mathrm{MPa}$$
$$1073.15\,\mathrm{K} \le T \le 2273.15\,\mathrm{K} \quad \text{for} \quad p \le 50\,\mathrm{MPa}$$

where the thermodynamic properties are considerably more accurate than in IFC-67. The algorithms are also more than five times faster than the ones in IFC-67, except for the supercritical region where it is approximately three times faster, see [19].

The C++ implementation of IAPWS-IF97 in this study was written from the specification given in [19], where the primary variables are defined as pressure and enthalpy. Given those two state variables the implementation returns the steam quality *x*, the density ρ, the temperature *T* and the partial derivatives of all those variables both with respect to *p* and *h*. Those values are then used in the system equations when applicable.

Implementation in OpenFOAM

Implementation of new models is in most cases relatively simple in OpenFOAM. Low level operations regarding individual computational cells or the solution of linear systems need not be addressed in most of the cases, and the programming framework is designed with customization in mind.
As an example of this, the basic lines of code required to represent the first equation in the matrix above can be written as

```
por * rho_p * fvm::ddt(p)
+ por * rho_h * fvc::ddt(h)
- fvm::laplacian((kappa / mu) * rho
, p) + fvc::div(kappa / mu *
(g & mesh.Sf()), rho * rho) == 0
```

with all the relevant parameters defined, e.g. the density rho and its partial derivatives (which have been calculated before) `rho_p` and `rho_h`. Of course there more coding is needed, such as for defining the variables as field function, but the developer has no need to become familiar with the inner workings of the numerics. A good example are the functions fvm::ddt and fvm::laplacian shown above, which will automatically result in a construction of a linear system for an implicit solution of an unsteady diffusion equation. Other functions such as fvc::div denote differential operators in an explicit manner, usually performed on fields that are not to be solved, e.g. the *h* field in the pressure equation.

Another short example can be examined here, which is the calculation of the flow field $\vec{u}$ after solving the pressure equation is implemented in OpenFOAM as

```
U = -kappa / mu * (fvc::grad(p)-
rho*g);
```

which calculates a vector field U using the pressure and appropriate differentiation operators. The energy equations which is a bit more cumbersome is implemented in a very similar manner, but is not shown here since the principles are the same as in the continuity equation.

One notable drawback of the OpenFOAM architecture is the fact that different equations are solved in a segregated manner, thus making it difficult to couple equations when performing a linear solution step between time iterations. In most cases though, this does not pose a problem, since typical flow problems are non-linar and require iterations between different solution procedures anyway. In the case of the pressure-enthalpy equation coupling presented here, the obvious approach is to solve the continuity equation and energy equation by an iterative process, and thus performing a fixed-point iteration between solutions. Currently, this approach works very well for single phase calculations.

Specific boundary conditions

This system of equations is solved by giving the boundary conditions for pressure and enthalpy or their derivatives. However, custom boundary conditions can be constructed from variables that are dependent either one of those variables. If for example, mass flux is required, Darcy's equation an be manipulated to give mass flux through unit area such that

$$\dot{\mathrm{m}} = \frac{\kappa\rho}{\mu}(\nabla p - \rho\vec{g})$$

This would make it possible to define constant mass flux, $\dot{m}$ along the boundary such that

$$\hat{\mathrm{n}} \cdot \nabla p = \frac{\mu}{\kappa\rho}\dot{m} + \rho(\vec{g} \cdot \hat{\mathrm{n}})$$

This can then be quite easily implemented in OpenFOAM in the following manner

```
gradient() = mu_ * massFlux_ /
kappa_ / rho + rho * (g_ & n);
```

RESULTS FROM CASE STUDIES

Since the study is a work in progress the case studies presented here are relatively simple and do not represent a fully fledged geothermal reservoir with accurate properties and dimensions. As the work progresses further details will be added to the implemented codes and will consequently be validated using currently available software such as TOUGH2 as well as measurements from reservoir sites and laboratories. The following section can mainly be viewed as testing of the application of the methodology presented in the paper.

Axisymmetric flow around a well

Subheadings within sections follow the same format as section titles, but with upper and lower case.
A case was set up in order to validate the model. An axisymmetric mesh with an angle of 5°5° was generated, where the number of cells in the radial direction was 400 and the number of cells in the horizontal direction was 400. The domain radius was chosen as 60m and with a depth of 60m into the ground. In order to simulate the lining of a well in the centre, impermeable walls were defined one cell-length away from the axis of rotation, and they were adjusted to reach down to 30 m.

Boundary conditions for the pressure were then defined as zero mass flux at the bottom of the domain, and a constant pressure of d was maintained on the top of the reservoir. At the outer edges of the reservoir, the gradient of the pressure in the outward direction was assumed to be zero. For the initial conditions the pressure was assumed to increase hydrostatically from 100 kPa at the top with a gradient 9584.37 Pa/m.

For the enthalpy the boundary conditions were defined such that a constant enthalpy of 400 kJ/kg was maintained at the bottom. Other boundaries where defined as having an enthalpy gradient of zero. The simulation time in the model was 10^7 seconds with a timestep of 10^4 seconds, which is rather large, but represents changes in a reservoir. The problem was broken down into 16 subdomains which were solved in a parallel manner on a computer cluster. The mesh can be seen on figure 1.

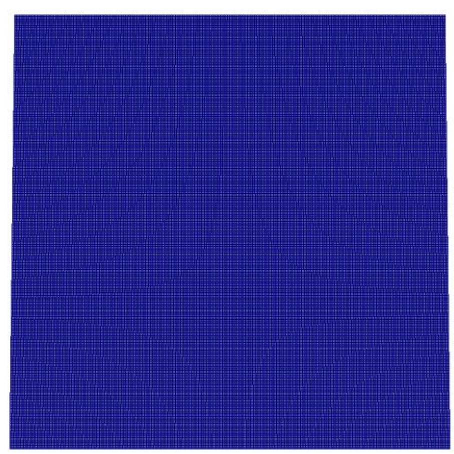

Figure 1: The axisymmetric mesh of the problem

The main results from the calculations are the enthalpy distribution, illustrated in figure 2 and the pressure distribution shown in figure 3.

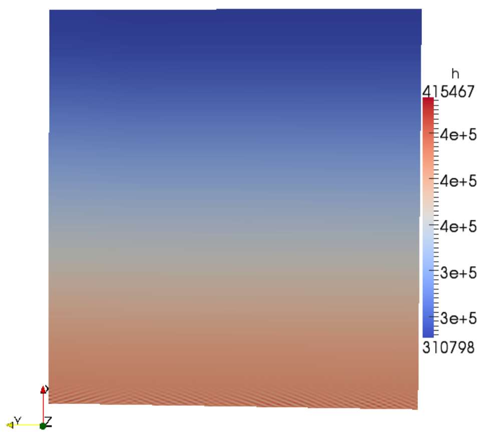

Figure 2: Enthalpy distribution

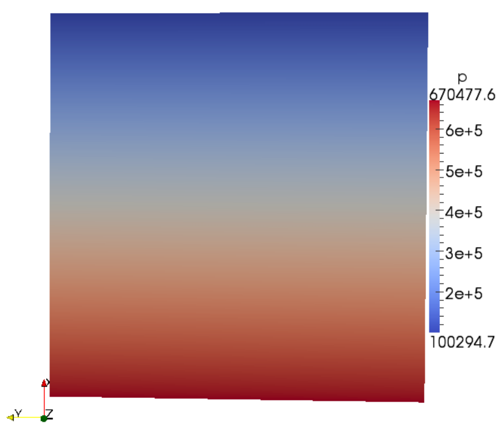

Figure 3: Pressure distribution

The enthalpy for various time steps as a function of coordinates in the vertical direction can be seen on figure 4 and the density on figure 5.

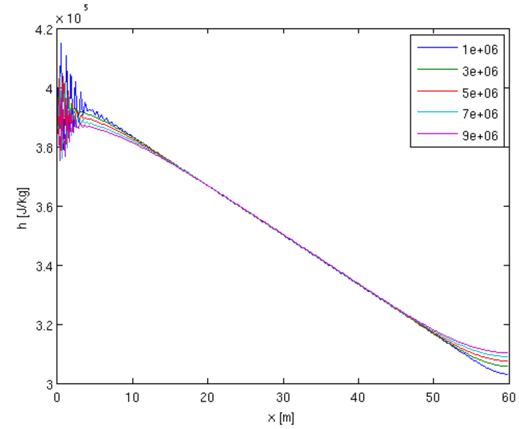

Figure 4: Enthalpy as a function of vertical coordinates

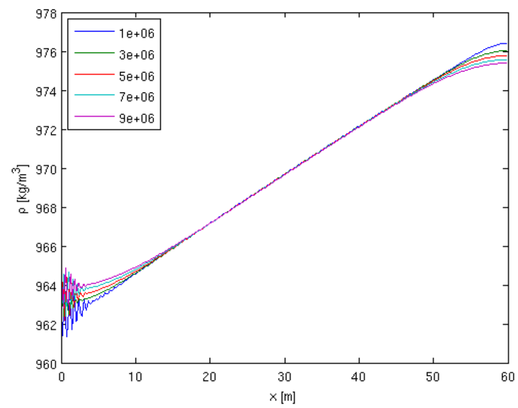

Figure 5: Density as a function of vertical coordinates.

<u>Three dimensional modeling around a well</u>

A model has also been constructed for a three dimensional flow around a well. The main benefits of that is the ability of modeling non-symmetrical problems, such as interaction between wells and directionally drilled wells. A cut through the mesh that was constructed for that problem can be seen on figure 6.

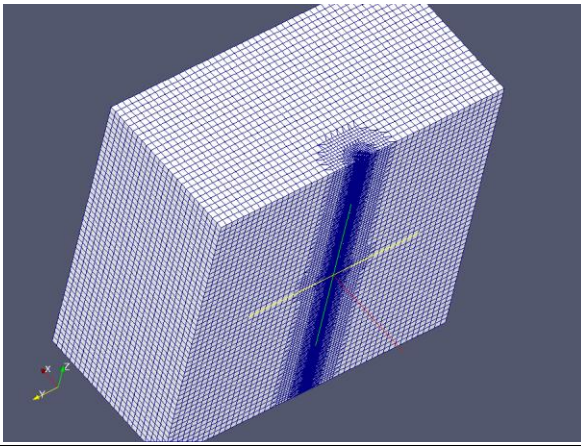

Figure 6: The mesh for the three dimensional modeling around a well.

If this problem is solved in a similar manner as the previous case, the results give a solution for the velocity field which can be seen in figure 7. This assumes a well that is under utilization, where the lining goes down along the well for the first 2/3rd of the length and inflow is allowed for the deepest part of the well.

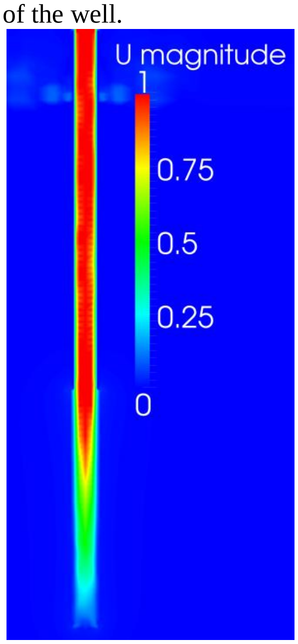

Figure 7: The velocity field for a well under utilization.

<u>Natural convection in a large ideal reservoir</u>

For final testing purposes a three dimensional reservoir has been modeled with a simplified version of the developed solver, namely assuming constant fluid properties and focusing on temperature instead of enthalpy. The physical behavior of these calculations can be characterized by the dimensionless Rayleigh number, defined for permeable media as

$$\mathrm{Ra} = \frac{\rho^2 c_p g \beta (T_1 - T_0) \kappa L}{\mu k}$$

where β is the thermal expansion coefficient and L is the reservoir height.

Two domains were specified in this case, both indicating a planar domain with a thickness that represents the height of the reservoir. In the first test, a rectangular domain was used with Ra=100, as shown in figure 8, which also shows the pressure distribution after a relatively long simulation period. The figure shows the high pressure areas underneath the reservoir, which form irregular patterns because of the fluid movement from the hot bottom to the cold top.

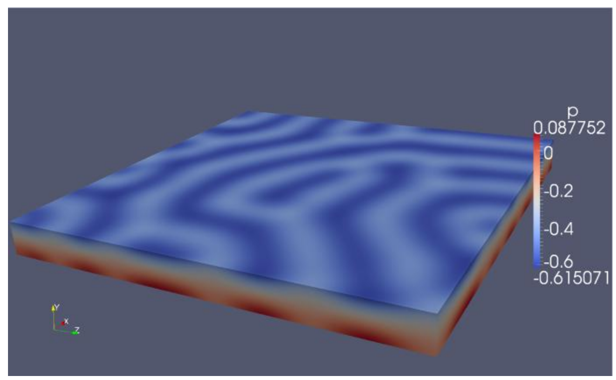

Figure 8: Pressure distribution in a rectangular domain.

The temperature distribution can also be observed by cutting through the center of the reservoir and plotting the temperature distribution at that given depth. Note that the bottom temperature is specified as the dimensionless value 1and the top is set to a temperature value of 0. Figure 9 shows the distribution, clearly indicating the irregular behaviour of upwards flowing regions (hot, red) and downwards flowing regions (cold, blue).

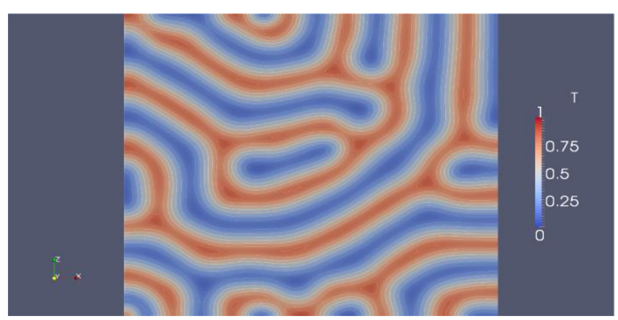

Figure 9: Temperature distribution halfway from the reservoir top to the bottom

The relation between flow and temperature can be illustrated further by looking at a vertical cross section of the domain, showing the flow direction and the corresponding hot plumes going from the bottom to the top, see figure 10.

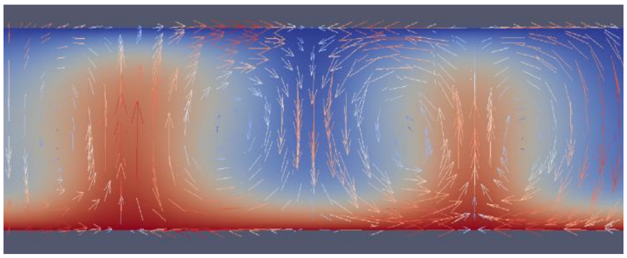

Figure 10: Temperature distribution and flow

The final illustration shows results from a calculation on a disk shaped domain, with Ra=500. Now the flow behaviour has become chaotic, much more irregular Rayleigh-Benard convection cells are observed in figure11. Note that this figure shows the temperature at the same vertical location as in figure 9

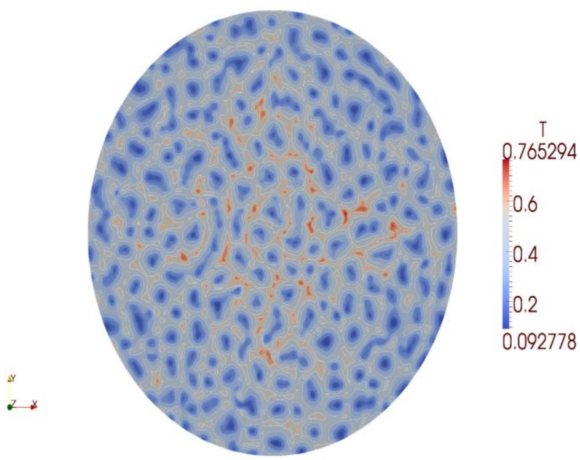

Figure 11: Temperature distribution with a high Rayleigh number

DISCUSSION

This paper illustrates the applicability of the OpenFOAM platform to take on current problems in geothermal reservoir modeling as well as multiphase flow in porous media in general. Because of the structure of the OpenFOAM libraries, both the partial differential equations which describe the problem, as well as the IAPWS-IF97 standard for the behaviour of water of steam, can be implemented in a consistent manner with minimal work.

However this work is still in progress, so there are many factors still unaccounted for. This includes relative permeabilities of steam and liquid water expressed as a function of various flow parameters, thus assuming different flow velocities for the two phases. Also, some important parameters such as fluid viscosity have not yet been implemented from standard, such as the IAPWS formulation.

Currently the main focus of the research is to include phase changes in the model and account for a phase mixture within some regions of the reservoir. The main challenge in this work is to ensure a stable solution despite the discontinuities in physical properties that arise as a result of phase changes. This has still not been resolved adequately and some instabilities are seen in two phase solutions, hence no results are shown here for such computations.

Despite those current issues, it can be proposed that the OpenFOAM platform is very promising for geothermal reservoir modeling. However, such further research and modeling work will always require comparison work, especially with well known and mature reservoir models.

On a whole, this approach in modeling geothermal reservoirs has several advantages over present methods. Since the libraries are highly customizable, the wellbore-reservoir interaction can be modeled in a flexible way and adjusted to represent known data from measurements. Furthermore, by using the IAPWS-IF97 standard the fluid properties are defined accurately for a wide range of temperatures and pressures, notably including the ranges close the critical point, where present models typically have shown some behavioral problems.

ACKNOWLEDGEMENTS

The authors would like to thank Landsvirkjun Power Company, the Geothermal Research Group (GEORG) and the University of Iceland for the financial support of this project. Their help is greatly appreciated.

REFERENCES

[1] Michael J. O’Sullivan. State of the art of geothermal reservoir simulation. *Geothermics*, 30(4):395–429, August 2001.
[2] Stanford Geothermal Program. Proceedings of the Special Panel on Geothermal Model Intercomparison Study. In *Report SGP-TR-42*, Stanford, CA, 1980.
[3] Karsten Pruess. TOUGH2 — A General Purpose Numerical Simulator for Multiphase Fluid and Heat Flow. Technical Report May, Lawrence Berkeley Laboratory, Berkeley, California, 1991.
[4] Gunnar Gunnarsson, Andri Arnaldsson, and Anna Lilja Oddsdóttir. Model Simulations of the Hengill Area, Southwestern Iceland. *Transport in Porous Media*, August 2010.

[5] Grimur Bjornsson, Arnar Hjartarson, Gudmundur S Bodvarsson, and Benedikt Steingrimsson. Development of a 3-D Geothermal Reservoir Model for the Greater Hengill Volcano in SW-Iceland. In *Proceedings, TOUGH Symposium 2003*, pages 1–11, 2003.
[6] Gudni Axelsson, Grimur Bjornsson, and National Energy Authority. Detailed Three-Dimensional Modeling of the Botn Hydrothermal System in N-Iceland. In *Proceedings, Eighteenth Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California, 1993.
[7] Michael J. O'Sullivan, Angus Yeh, and Warren I. Mannington. A history of numerical modelling of the Wairakei geothermal field. *Geothermics*, 38(1):155–168, March 2009.
[8] S Zarrouk, Michael J. O'Sullivan, A Croucher, and Warren I. Mannington. Numerical modelling of production from the Poihipi dry steam zone: Wairakei geothermal system, New Zealand. *Geothermics*, 36(4):289–303, August 2007.
[9] Warren I. Mannington, Michael J. O'Sullivan, and D Bullivant. Computer modelling of the Wairakei-Tauhara geothermal system, New Zealand. *Geothermics*, 33(4):401–419, 2004.
[10] Yurie Kumamoto, Ryuichi Itoi, T Tanaka, and Yukio Hazama. Modeling And Numerical Analysis Of The Two-Phase Geothermal Reservoir At Ogiri, Kyushu, Japan. In *Proceedings, Thirty-Fourth Workshop on Geothermal Reservoir Engineering*, 2009.
[11] A. V. Kiryukhin, N. P. Asaulova, S Finsterle, T. V. Rychkova, and N.V. Obora. Modeling the Pauzhetsky Geothermal Field, Kamchatka, Russia, Using ITOUGH2. In *Proceedings, TOUGH Symposium 2006*, pages 1–8, Berkeley, California, 2006.
[12] Jialing Zhu and Haiyan Lei. Analysis of the Doublet-Well Distance Effects in a Porous Medium Geothermal Reservoir in Tianjin, China. In *Proceedings, TOUGH Symposium 2009*, 2009.
[13] E Porras, T Tanaka, H Fujii, and Ryuichi Itoi. Numerical modeling of the Momotombo geothermal system, Nicaragua. *Geothermics*, 36(4):304–329, August 2007.
[14] A Battistelli, Amdeberhan Yiheyis, Claudio Calore, Corrado Ferragina, and Wale Abatneh. Reservoir engineering assessment of Dubti geothermal field, Northern Tendaho Rift, Ethiopia. *Geothermics*, 31(3):381–406, June 2002.
[15] E B Emoricha, J B Omagbon, and R C M Malate. Three Dimensional Numerical Modeling of Mindanao Geothermal Production Field, Philippines. In *Proceedings, Thirty-Fifth Workshop on Geothermal Reservoir Engineering*, number October 1995, 2010.
[16] H G Weller, G Tabor, Hrvoje Jasak, and C Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6), 1998.
[17] V Henson. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155–177, April 2002.
[18] Hrvoje Jasak, Aleksandar Jemcov, and Zeljko Tukovic. OpenFOAM : A C ++ Library for Complex Physics Simulations □. In *International Workshop on Coupled Methods in Numerical Dynamics*, volume m, pages 1–20, 2007.
[19] The International Association for the Properties of Water and Steam. Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. Technical Report August 2007, Lucerne, Switzerland, 2007.
[20] A Croucher and Michael J. O'Sullivan. Application of the computer code TOUGH2 to the simulation of supercritical conditions in geothermal systems. *Geothermics*, 37(6):622–634, December 2008.