This work is presented in Padjadjaran Earth Dialogue: International Symposium on Geophysical Issues (PEDISGI 2015), 8-10 June 2015, Jatinangor, Indonesia.

Geothermal well behaviour prediction after air compress stimulation using one-dimensional transient numerical modelling

W Yusman[1,4], **S Viridi**[2,4] and **S Rachmat**[3,4]
[1]Geothermal Engineering, Faculty of Mining and Petroleum Engineering
[2]Nuclear Physics and Biophysics, Faculty of Mathematics and Natural Science
[3]Petroleum Engineering, Faculty of Mining and Petroleum Engineering
[4]Institute Teknologi Bandung, Ganesha 10 st. Bandung 40132, Indonesia

E-mail: windrayusman@gmail.com

Abstract. The non-discharges geothermal wells have been a main problem in geothermal development process and well discharge stimulation is required to initiate a flow. Air compress stimulation is one of the methods to trigger a fluid flow from geothermal reservoir. The result of this process can be predicted by using by A_f/A_c method [3] but sometimes this method show uncertainty result in several geothermal wells and does not take into account the time required for the process. This paper presents a simulation model of non-discharges well under air compress stimulation to predict well behavior and time process required. The component of this model consists of geothermal well data during heating-up process such as pressure, temperature and mass flow in water column and main feed zone level (based on several geothermal wells). One-dimensional transient numerical model is run in the Single Fluid Volume Element (SFVE) simulator [4] and the input data is read in to model. Based on simulation result, the geothermal well behavior prediction after air compress stimulation will be valid under two specific circumstance such as single phase fluid density between 1 – 28 kg/m^3 and above 28.5 kg/m^3. The first condition shows that successful well discharge and the last condition represent failed well discharge after air compress stimulation (only for two wells data). Discharge time is about 80 s or less, which is too short compared to a couple hours as observed field data. This models need to improve by updating more geothermal well data and modified fluid phase condition inside the wellbore.

1. Introduction

Geothermal energy developments in Indonesia are constantly faced with several challenges such as exploration and engineering aspects. In engineering, the main goal of geothermal energy utilization is to have its well discharges but sometimes the geothermal wellbore do not show any fluid discharges. This problem is caused by the presence of water column in the wellbore which is standing above the major permeable zone [1] and to initiate flow, a well discharge stimulation is required. One of the effective method to stimulate non-flowing geothermal well is air compression method. Basically this method boost the energy in the water column by conductive heating at the certain depth.The liquid is depressed and it will increase the potential energy in the compressed gas [2]. Especially in air compression method, the prediction of "well behaviour" after stimulation process can be described by using A_f/A_c method [3]. This method is calculated the ratio between two areas in static pressure and temperature profile (defined as A_f and A_c) in order to find an association between successful and failure attempt of well stimulation trial. Sometimes the result of air compress stimulation prediction using A_f/A_c method represent different outcome than well discharge test trial [1]. The uncertainty result by using previous prediction might be minimized with numerical modelling which described the geothermal well condition during stimulation process. The one-dimensional numerical modelling method which used in this study modified by using Single Fluid Volume Element method (SFVE) in siphon system [4]. The modified model will be adapted in geothermal well condition during air compress stimulation process. The simulation result of this condition model will be expected to predict geothermal well behaviour after air compress stimulation (based on static pressure and temperature profile data).

2. Basic Theory

In this section all theory involved in geothermal well stimulation are listed and briefly explained, including geothermal wells principle, wells problem and wells stimulation. In this part also described about simulation using modified Single Fluid Volume Element method (SFVE).

2.1. Geothermal wells principle and problem

The geothermal wells represent the capability of the wells to produce heat from sub-surface. The main process of geothermal wells system are heat and mass transfer of geothermal fluid from reservoir into earth surface [5]. Especially in liquid-dominated geothermal system, the produce fluid can be flow consider following condition.

$$P_{reservoir} > P_{hydrostatic} + P_{air-inside} \quad (1)$$

Geothermal wells categorized into two type based on discharge capability such as artesian wells and non-artesian wells [1]. Artesian geothermal wells refer to naturally geothermal fluid flow through well casing into surface and fulfil Equation (1) condition. Conversely, non-artesian wells show the un-natural occurrence caused by the shallow water level and deep major permeable zone. This type of wells located in the liquid dominated geothermal systems which have distinction in elevation and consequently the water levels in each wells show a different depth [1].The presence of water column level inside the wells will increase the hydrostatic pressure and consider the Equation (1) there are no flow from reservoir into surface.

2.2 Geothermal wells stimulation and previous prediction method

The problem of non-artesian geothermal wells can be solved by using several stimulation method. The principal of those stimulation method is decrease hydrostatic pressure inside the wellbore by reducing a density or thickness of water column [1]. There are several methods of well stimulation for non-artesian type such as air (or gas) compression, gas lifting, and well-to-well two phase injection and all of these methods are usually used in liquid dominated geothermal systems. In particular, air compression method boost the energy in the water column by conductive heating at the certain depth to which the liquid is depressed [2]. Volume of column water is heated exceed the saturation curve for the zero wellhead pressure and the immediate opening of the master valve will cause flashing in the well and this condition will initiate a discharge mechanism [2]. All of the circumstance before air compress stimulation shown in Figure 1 and after air pressure compress shown in Figure 2.

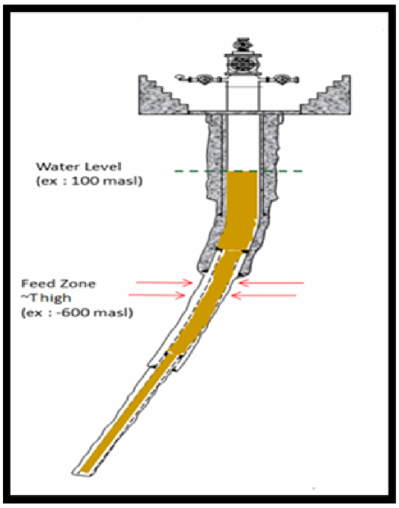

Figure 1. Initial condition of non-artesian geothermal wells

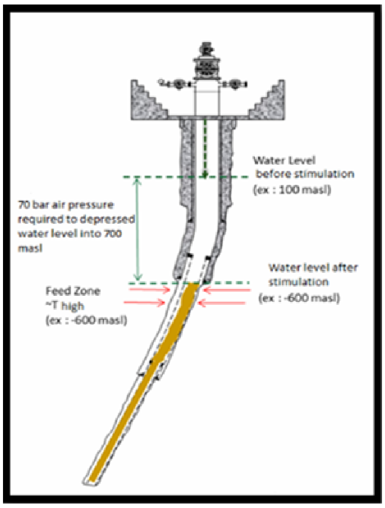

Figure 2. Final condition after depressed with air pressure compress

In air compression method, the success prediction of the well discharge had been developed by created a simple empirical formulation [3]. Basically, this formulation is calculated the ratio between two areas such as A_f and A_c. A_f refer to the flashing area that is limited by the saturation temperature of the water column with wellbore temperature. A_c is the condensation areas including near the wellhead that is limited by temperature curve in the well and the curve of 100^{o}C.The ratio between A_f and A_c of some targeted wells were taken together with water column level before stimulation to find an association between successful and failures attempt, yet there are several limitation of this formulation. The first condition is an initial limiting case; the upper well flowing temperature must be around 100^{o}C. The ratio of A_f and A_c less than 0.7 refer as failure attempt and the wells with A_f/A_c above 0.85 define as high probability to discharge. The ratio between both values (0.75 to 0.85) is uncertainty results [3]. However, based on field experience, there are several wells with high discharge probability using A_f/A_c prediction unfortunately imprecise estimate.

2.3 Single Fluid Volume Element (SFVE)

The uncertainty result by using A_f/A_c ratio can be minimized with develop a simple well model which represent the condition of non-artesian well before and after air compress stimulation method. The wellbore model will be simulated based on well data such as pressure, temperature and mass flow for each depth. One of the method to represent the well stimulation process are one-dimensional transient flow simulation and basically this method related with physical processes in geothermal wellbore. This research is conducted by simulation of transient fluid flow in self-siphons using Single Fluid Volume Element Method (SFVE). Basically, this study can be modified to predict flow occurrence in the geothermal systems.

SFVE method is a finite difference based method, which is used to solve equation of motion of a single fluid volume element. This element represents transient motion of the fluid along its channel. Details of illustration and derivation of this method for vertical, horizontal, inclined, and semi-circular pipes can be found in [4], while only required equations presented in this work. The element will have pressure force from fluid above and below it.

$$F_p = \rho g\left[\left(z + \Delta h\right) - z\right] A \Delta h, \qquad (2)$$

where A is well cross section, Δh thickness of the element, z position of the element, and ρ is fluid density, which is in general

$$\rho(T,P)=\frac{\rho_0}{[1+\beta(T-T_0)][1-(\rho-\rho_0)/E]} \tag{3}$$

With ρ_0= 1- 40 kg/m^3, E= 2.15 GPa, p_0= 1.023×10^5 Pa, β = 2×10^{-4} m^3/m^3·°C, T_0 = 0 °C. Temperature and pressure are also function of depth z, which are provided by temperature- and pressure-depth profile. There is also force due pressure drop related to position of the element relative to initial position of shallow water column z_0

$$F_d = \rho g[(z+z_0)] \tag{4}$$

Gravitation plays also a role to the element through

$$F_g = -\rho g A \Delta h \tag{5}$$

The last considered force is viscous drag between the element and its channel

$$F_v = -8\pi\eta\Delta h v \tag{6}$$

As in [6], which is modified from [7]. Fluid viscosity η is also function of temperature

$$\eta(T)=\eta_0 x 10^{247.8/(T-140)} \tag{7}$$

With η_0= 2414 cP. Other forms [8] of Equation (7) can also be used. Then using Newton second law of motion, acceleration of fluid element

$$a=\frac{1}{\rho A \Delta t}\left(F_p+F_d+F_g+F_v\right) \tag{8}$$

Integration of Equation (7) will produce velocity, and late integration will produce position. Numerical Method used is simple Euler method.

$$v(t+\Delta t)=v(t)+a(t)\Delta t \tag{9}$$

and

$$z(t+\Delta t)=z(t)+v(t)\Delta t \tag{10}$$

Iteration though Equations (2) - (10) from initial time to final time will give transient dynamics of fluid element with thickness Δh. Two termination conditions can be implemented, which $z > z_{\text{well head}}$ for dischargeable well and $v<\varepsilon$ for un-dischargeable well.

3. Geothermal wells properties

Data from two different wells are used in this work for testing the simulation, where "Wells-A" represents unsuccessful wells and "Wells-B" represents successful wells [1]. Figure 3 and Figure 4 show profiles of temperature-depth and pressure-depth for both well.

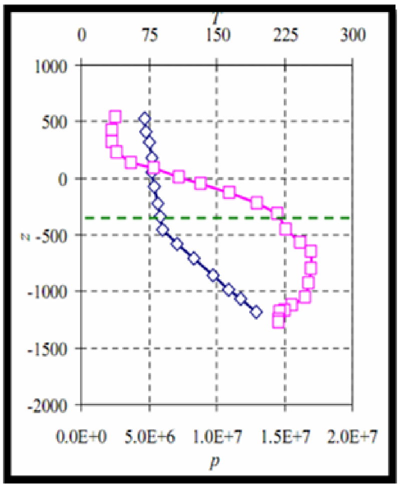

Figure 3. PT static profile "Wells-A" before stimulation (failed attempt)

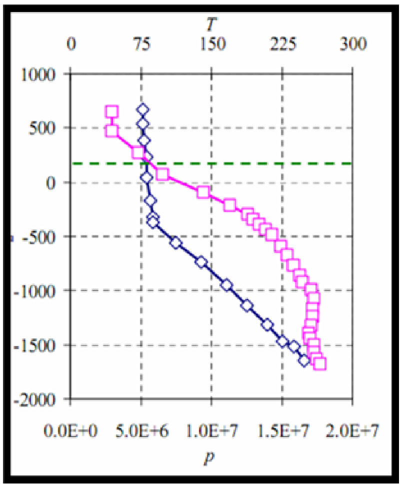

Figure 4. PT static profile "Wells-B" after stimulation (success attempt)

The "Wells-A" has shallow water position at -350 m, temperature about 40 °C, pressure of 1 MPa, and it fails to discharge, where "Wells-B" has shallow water position at about 200 m, temperature about 70 °C, pressure of 5.4 MPa, and it is successful to discharge.

4. Result & discussion

Simulation parameters are $\Delta h = 10$ m and $\Delta t = 10^{-4}$ s, with range of ρ_0 (1 – 40 kg/m^3) in order to match observation data for flow occurrence at the two wells [1], where the result given in Figure 5. It is found that the cross over line between the flow and no flow states is located between values $\rho_0 = 28$ kg/m^3 and $\rho_0 = 28.5$ kg/m^3 for the well "Wells-B", while for the "Wells-A"all variation of ρ_0 give no flow. Observation result give flow occurrence of "Wells-B". This means that this method produce result that match observation only for values 1 kg/m^3 $< \rho_0 <$ 28 kg/m^3, which is the density rather for vapour (0.8 kg/m^3) than water (1000 kg/m^3). Single-phase is determined using Equation (3). If single-phase density can be defined simply through

$$\rho_0 = \alpha \rho_{water} + (1-\alpha)\rho_{vapor} \qquad (11)$$

or

$$\alpha = \frac{\rho_0 - \rho_{vapor}}{\rho_{vater} - \rho_{vapor}} \qquad (12)$$

Then value of α is between 2 x 10^{-4} and 2.72 x 10^{-2} for the previous range of ρ_0. It is very vapour-rich mixture of water-vapour. In this study Equation (12) is labelled as density fraction water. Time require for flow to occur as observed in well head is about 80 s simulation time, which is too short comparing to real required time about couple of hours.

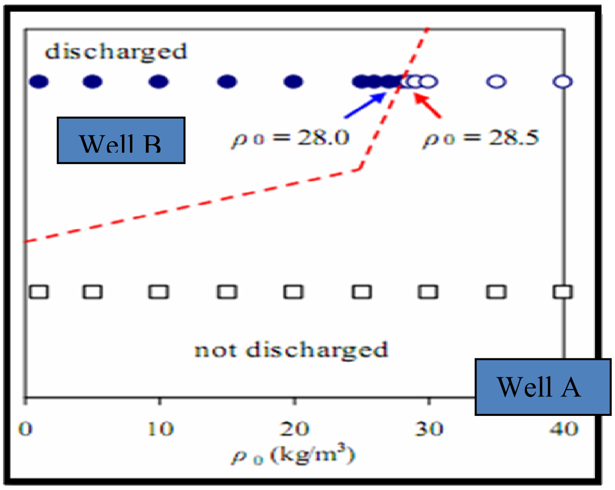

Figure 5. Flow occurrence as function of ρ for the two wells [1] using SFVE method.

This unrealistic result that density of the discharged fluid is too light and discharge time is too short shows that implementation of SFVE, which is considering single-phase fluid, does not work well. Advancement of this method by considering two-phase flow. The unrealistic result can also be addressed to static pressure-depth profile, which changes during performing the SFVE.

5. Conclusion

By and large, prediction of well behavior after air compress stimulation using 1-dimensional transient numerical modeling based on two geothermal wells will be valid with two specific circumstance. First condition is single phase fluid density between 1 – 28 kg/m^3 and the latter condition is above 28.5 kg/m^3. Based on simulation result, the first condition shows that successful well discharge and the last condition represent failed well discharge after air compress stimulation (only for two wells data). Single phase fluid between 1 - 28 kg/m^3, which could be too light for water-vapor mixture, where the density fraction is between 2×10^{-4} $kg/m^3$and 2.72 $\times10^{-2}$ kg/m^3 for water. Discharge time is about 80 s or less, which is too short compared to a couple hours as observed field data. This model need to improve by modified Single Fluid Volume Element (SFVE) method with two-phase flow assumption and collect more geothermal wells data with successful and failed discharge after stimulation.

Reference

[1] F. X. M. Sta. Ana 1985 *A Study on Stimulation by Air Compression on Some of the Philippine Geothermal Wells* Geothermal Institute. University of Auckland.

[2] Siega C, Saw V, Andrino Jr R and Canete G 2006 Well-toWell Two Phase Injection Using a 10in Diameter Line to Initiate Well Discharge in Mahagnadong Geothermal Field, Leyte, Philippines *Proc. of the 7th Asian Geothermal Symposium*.

[3] Stock 1983 *Condition Acquired for Successful Air Compression Stimulation* The Phillipines PNOC-EDC/KRTA Internal Report.

[4] Viridi S, Novitrian, Nurhayati, Latief F.D.E and Zen FP 2014 Development of Single Fluid Volume Element Method for Simulation of Transient Fluid Flow in Self-Siphons *AIP Conference Proceeding* **1615** 199-207

[5] Grant M,A and Bixley P.F 2011 *Geothermal Reservoir Engineering* Academic Press Boston Elsevier Vol.2 pp. 269-283

[6] S. Viridi, Suprijadi, S. N. Khotimah, Novitrian, and F. Masterika, "Self-Siphon Simulation using Molecular Dynamics Method", Recent Development in Computational Science 2, 9-16 (2011).

[7] Faber T E 1995 *Fluid Dynamic for Physicist* Cambridge: University Press pp 227-232

[8] Seeton C J 2006 *Viscosity –Temperature Correlation for Liquids* Tribology Letters pp 67-84

Appendix A: Wells data

```
# Simulation parameters
DT       1E-4
TSHOW    0.1
TEND     200
DH       1
EPS      1E-8
VERBOSE  0

# Physical properties
GRAVITY  9.81
VISCOSITY         1E-2

# Well information
W_NAME   Well-A
W_DATE   1984-01-00
W_STATUS Failed
W_DIAMETER        0.5
W_HEAD   550.0

# Shallow water column
SWC_POSITION      -350.0
SWC_TEMPERATURE   40.0
SWC_PRESSURE      1.0E6

# Feed zone
FZ_POSITION       -1450.0
FZ_THICKNESS      50.0
FZ_MASS_FLOW      20.0
FZ_TEMPERATURE    197.0
FZ_PRESSURE       11.4E6

# Air compress target
ACT_POSITION      -1100
ACT_TEMPERATURE   247
ACT_PRESSURE      1.18E7

# Interpolation data
DEPTH_PRESSURE_DATA        15
523.0    4.7E6
413.5    4.8E6
320.2    5.0E6
186.4    5.2E6
48.5     5.2E6
-77.2    5.4E6
-219.1   5.7E6
-332.7   5.8E6
-454.3   6.0E6
-576.0   7.1E6
-705.8   8.3E6
-859.9   9.7E6
-985.6   10.9E6
-1066.7  11.8E6
-1176.2  12.9E6

# Interpolation data
DEPTH_TEMPERATURE_DATA     22
541.0    38.2
425.5    34.8
315.6    34.8
227.6    39.9
139.6    55.5
84.6     80.6
7.7      109.2
-52.8    133.4
-135.3   164.6
-217.8   195.7
-311.3   218.2
```

```
-454.2   226.9
-569.7   243.3
-652.1   254.6
-795.1   254.6
-921.6   252.0
-1053.5  247.6
-1125.0  233.8
-1169.0  226.0
-1180.0  219.9
-1246.0  219.1
-1273.5  219.1


# Simulation parameters
DT       1E-4
TSHOW    0.1
TEND     200
DH       1
EPS      1E-8
VERBOSE  0

# Physical properties
GRAVITY  9.81
VISCOSITY        1E-2

# Well information
W_NAME   Well-B
W_DATE   1984-10-00
W_STATUS Success
W_DIAMETER       0.5
W_HEAD   682.0

# Shallow water column
SWC_POSITION      200.0
SWC_TEMPERATURE   70.0
SWC_PRESSURE      5.4E6

# Feed zone
FZ_POSITION       -1350.0
FZ_THICKNESS      100.0
FZ_MASS_FLOW      20.0
FZ_TEMPERATURE    254.0
FZ_PRESSURE       13.3E6

# Air compress target
ACT_POSITION      -400
ACT_TEMPERATURE   200
ACT_PRESSURE      6.0E6

# Interpolation data
DEPTH_PRESSURE_DATA        16
671.2    5.1E6
544.5    5.1E6
389.0    5.2E6
233.4    5.4E6
49.1     5.45E6
-169.8   5.7E6
-319.5   5.9E6
-365.6   5.9E6
-555.7   7.5E6
-740.0   9.3E6
-953.1   11.1E6
-1143.2  12.5E6
-1316.0  14.0E6
-1465.8  15.0E6
-1517.6  15.9E6
-1644.3  16.6E6

# Interpolation data
DEPTH_TEMPERATURE_DATA     27
646.4    44.1
```

```
464.4    44.8
268.0    73.5
71.6     99.2
-100.8   141.5
-210.9   170.1
-301.9   189.0
-345.0   194.3
-397.7   201.1
-440.8   207.9
-483.9   214.7
-598.9   224.5
-675.5   231.3
-766.5   237.3
-871.8   244.9
-929.3   247.9
-991.6   256.2
-1082.6  259.2
-1173.6  257.7
-1240.6  257.7
-1322.0  256.9
-1393.9  254.7
-1446.6  255.4
-1504.0  259.2
-1580.7  260.0
-1628.5  262.2
-1676.4  266.8
```

Appendix B: How to execute `gws1d` code

```
./gws1d Well-A.txt
./gws1d Well-B.txt
```

Appendix C: gws1d.cpp

```
/*
     gws1d.cpp
     Geothermal well simulation 1-d using
     Single Fluid Volume Element (SFVE)
     Sparisoma Viridi | dudung@gmail.com

     Compile: g++ gws1d.cpp -o gws1d
     Execute: ./gws1d [pfile ofile]

     20150328
         Create this code with all depth dependent variables
         are inline in the code
         Function for depth-pressure and depth-temperature
         relations
     20150422
         Change program name from sfvegws.cpp to gws1d.cpp
         Move empirical functions into empiricalf.h
         Modify using rwparams.h, from now this program uses
         pameters file
     20150423
         Continu modification for the program to use parameters
         file
         Modify interpolation function double fpress(double,
         vector<double>) and double ftemp(double,
         vector<double>), add second arguments
         Remove fbpd(double x) for now
         Add command for compiling and executing this program
         Finish porting this program to fully controlled by
         parameters file
     20150425
         Verbose SWC and ACT, compared to intepolation data
         Remove (int) from tt in viewing t with Tshow
         Remove pressure drop to fit failed and success as
         observed in the field (PAL-1RD and PAL-9D),
         explanation is still unknown
*/
```

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <cmath>
#include <vector>

#include "rwparams.h"
#include "empiricalf.h"

using namespace std;

// Interpolation-extrapolation functions
double fpress(double, vector<double>);
double ftemp(double, vector<double>);

// Main program
int main(int argc, char *argv[]) {
     // Show how to use this program
     const char *pname = "gws1d";
     if(argc < 3) {
         cout << "Usage: " << pname << " pfile ofile" << endl;
         cout << "pfile\tfile containing ";
         cout << "simulation and physical parameters" << endl;
         cout << "ofile\toutput file for time series ";
         cout << "of water column";
         cout << endl;
         return 1;
     }
     const char *pfile = argv[1];
     const char *ofn = argv[2];

     // Read all information from parameters file
     bool VERBOSE = false;
     readparam(pfile, "VERBOSE", VERBOSE);
     double dt = 0.0;
     readparam(pfile, "DT", dt);
     double Tshow = 0.0;
     readparam(pfile, "TSHOW", Tshow);
     double tend = 0.0;
     readparam(pfile, "TEND", tend);
     double dh = 0.0;
     readparam(pfile, "DH", dh);
     double eps = 0.0;
     readparam(pfile, "EPS", eps);

     double gacc = 0.0;
     readparam(pfile, "GRAVITY", gacc);
     double viscosity = 0.0;
     readparam(pfile, "VISCOSITY", viscosity);

     char w_name[32];
     readparam(pfile, "W_NAME", w_name);
     char w_date[32];
     readparam(pfile, "W_DATE", w_date);
     char w_status[32];
     readparam(pfile, "W_STATUS", w_status);
     double w_diameter;
     readparam(pfile, "W_DIAMETER", w_diameter);
     double w_head;
     readparam(pfile, "W_HEAD", w_head);

     double swc_z;
     readparam(pfile, "SWC_POSITION", swc_z);
     double swc_t;
     readparam(pfile, "SWC_TEMPERATURE", swc_t);
     double swc_p;
     readparam(pfile, "SWC_PRESSURE", swc_p);

     double fz_z;
     readparam(pfile, "FZ_POSITION", fz_z);
     double fz_dz;
```

```
readparam(pfile, "FZ_THICKNESS", fz_dz);
double fz_mf;
readparam(pfile, "FZ_MASS_FLOW", fz_mf);
double fz_t;
readparam(pfile, "FZ_TEMPERATURE", fz_t);
double fz_p;
readparam(pfile, "FZ_PRESSURE", fz_p);

double act_z;
readparam(pfile, "ACT_POSITION", act_z);
double act_t;
readparam(pfile, "ACT_TEMPERATURE", act_t);
double act_p;
readparam(pfile, "ACT_PRESSURE", act_p);

vector<double> dpd;
readparam(pfile, "DEPTH_PRESSURE_DATA", dpd);

vector<double> dtd;
readparam(pfile, "DEPTH_TEMPERATURE_DATA", dtd);

// Verbose all read information if necessary
if(VERBOSE) {
    cout << "dt = " << dt << endl;
    cout << "Tshow = " << Tshow << endl;
    cout << "dh = " << dh << endl;
    cout << "eps = " << eps << endl;
    cout << "Gravity = " << gacc << endl;
    cout << "Viscosity = " << viscosity << endl;
    cout << "Well name = " << w_name << endl;
    cout << "Well date = " << w_date << endl;
    cout << "Well status = " << w_status << endl;
    cout << "Well diameter = " << w_diameter << endl;
    cout << "Well head position = " << w_head << endl;
    cout << "Shallow water column position = ";
    cout << swc_z << endl;
    cout << "Shallow water column temperature = ";
    cout << swc_t << endl;
    cout << "Shallow water column pressure = ";
    cout << swc_p << endl;
    cout << "Feed zone position = " << fz_z << endl;
    cout << "Feed zone thickness = " << fz_dz << endl;
    cout << "Feed zone mass flow = " << fz_mf << endl;
    cout << "Feed zone temperature = " << fz_t << endl;
    cout << "Feed zone pressure = " << fz_p << endl;
    cout << "Air compress target position = ";
    cout << act_z << endl;
    cout << "Air compress target temperature = ";
    cout << act_t << endl;
    cout << "Air compress target pressure = ";
    cout << act_p << endl;
    cout << "Depth pressure data = ";
    display(dpd, 2);
    cout << "Depth temperature data = ";
    display(dtd, 2);
}

// Set up SFVE method
double w_area = 0.25 * M_PI * w_diameter * w_diameter;
double swc_volume = w_area * dh;
double rz = swc_z;
double vz = 0.0;

// Feed zone properties
double fz_rho = frho(fz_t, fz_p);
double fz_vf = fz_mf / fz_rho;
double fz_v = fz_vf / w_area;

// Verbose SWC and ACT data
cout << "SWC (intp) = " << swc_z << "\t";
cout << swc_t << " (";
```

```
cout << ftemp(swc_z, dtd) << ")\t";
cout << swc_p << " (";
cout << fpress(swc_z, dpd) << ")" <<  endl;

cout << "ACT (intp) = " << act_z << "\t";
cout << act_t << " (";
cout << ftemp(act_z, dtd) << ")\t";
cout << act_p << " (";
cout << fpress(act_z, dpd) << ")" <<  endl;

// Simulation time
double t = 0.0;

// Velocity change in each iteration
double dv = 1.0;
double vold = 0;

// Open file for output
ofstream fout;
fout.open(ofn);
fout << "# t(s)\trz(m)\tvz(m/s)" << endl;

// Variable for Tshow
int Nshow = Tshow / dt;
int ishow = 0;
bool SHOW = true;
while(dv > eps && rz < w_head
    && t < tend + Tshow) {
    // Output simulation result every Tshow period
    if(SHOW) {
            SHOW = false;

            // Adjust output time according to Tshow
            double tt = (t / Tshow);
            tt = tt * Tshow;

            fout << tt << "\t";
            fout << rz << "\t";
            fout << vz << endl;
    }

    // Pressure force
    double ptop = fpress(rz, dpd);
    double pbot = fpress(rz - dh, dpd);

    double swc_T = ftemp(rz, dtd);
    double swc_rho = frho(swc_T, 0.5 * (ptop + pbot));
    double pdrop = swc_rho * gacc * (rz - swc_z);

    double pall = (pbot - ptop) + pdrop;
    double Fp = pall * w_area;

    // Gravitation force
    double swc_m = swc_volume * swc_rho;
    double Fg = -gacc * swc_m;

    // Viscous force
    double swc_eta = feta(swc_T);
    double Ff = -8 * M_PI * dh * swc_eta * vz;

    // Total force
    double SF = Fp + Fg + Ff;

    // Finite difference integration
    double az = SF / swc_m;
    vz = vz  + az * dt;
    vz = (vz > fz_v) ? fz_v : vz;
    rz = rz + vz * dt;

    // Overcome problem if vz exceeds fz_v
    dv = (vz < fz_v) ? abs(vz - vold) : fz_v;
```

```
        vold = vz;

        // Maintain showing time
        if(ishow >= Nshow) {
                ishow = 0;
                SHOW = true;
        } else {
                ishow++;
        }
        t += dt;

        if(vz < 0) break;
    }
    fout << t << "\t";
    fout << rz << "\t";
    fout << vz << endl;
    fout.close();

    // Status from observation and simulation
    if(rz >= w_head)  {
        cout << "Status (sim) = " << w_status;
        cout << " (Success)" << endl;
    } else {
        cout << "Status (sim) = " << w_status;
        cout << " (Failed)" << endl;
    }
    cout << "Position (Well Head) = " << rz;
    cout << " (" << w_head << ")" << endl;

    return 0;
}

// Depth-pressure relation
double fpress(double x, vector<double> xy) {
    int N = xy.size() / 2;
    double xx[N];
    double yy[N];
    for(int i = 0; i < N; i++) {
        xx[i] = xy[2 * i];
        yy[i] = xy[2 * i + 1];
    }
    int j = -1;
    if(x >= xx[0]) {
        j = 0;
    } else if(x <= xx[N-1]) {
        j = N-2;
    } else {
        for(int i = 0; i < N-1; i++) {
                if((xx[i] >= x) && (x >= xx[i+1])) {
                        j = i;
                }
        }
    }
    double xmax = xx[j];
    double xmin = xx[j+1];
    double ymax = yy[j];
    double ymin = yy[j+1];
    double m = (ymax - ymin) / (xmax - xmin);
    double y = m * (x - xmin) + ymin;
    return y;
}

// Depth-temperature relation
double ftemp(double x, vector<double> xy) {
    int N = xy.size() / 2;
    double xx[N];
    double yy[N];
    for(int i = 0; i < N; i++) {
        xx[i] = xy[2 * i];
        yy[i] = xy[2 * i + 1];
    }
```

```
    int j = -1;
    if(x >= xx[0]) {
        j = 0;
    } else if(x <= xx[N-1]) {
        j = N-2;
    } else {
        for(int i = 0; i < N-1; i++) {
                if((xx[i] >= x) && (x >= xx[i+1])) {
                        j = i;
                }
        }
    }
    double xmax = xx[j];
    double xmin = xx[j+1];
    double ymax = yy[j];
    double ymin = yy[j+1];
    double m = (ymax - ymin) / (xmax - xmin);
    double y = m * (x - xmin) + ymin;
    return y;
}
```

Appendix D: Output

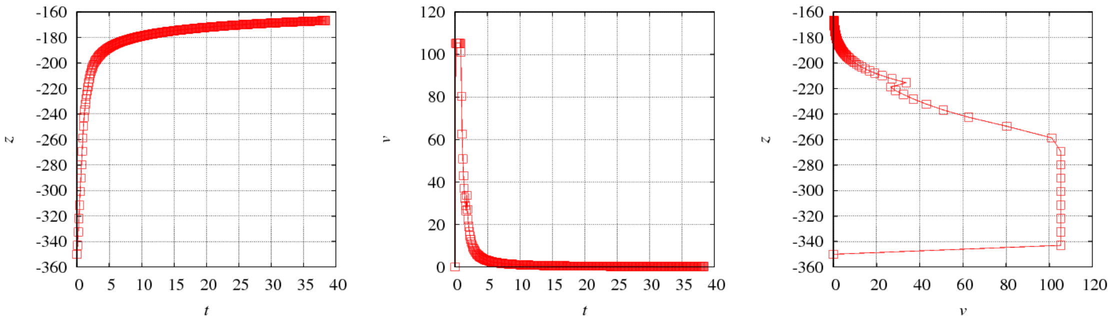

Figure D1. Fluid element position *z* (right) and velocity *v* (center) as function of time *t* and *z* against *v* for Well-A (not discharged).

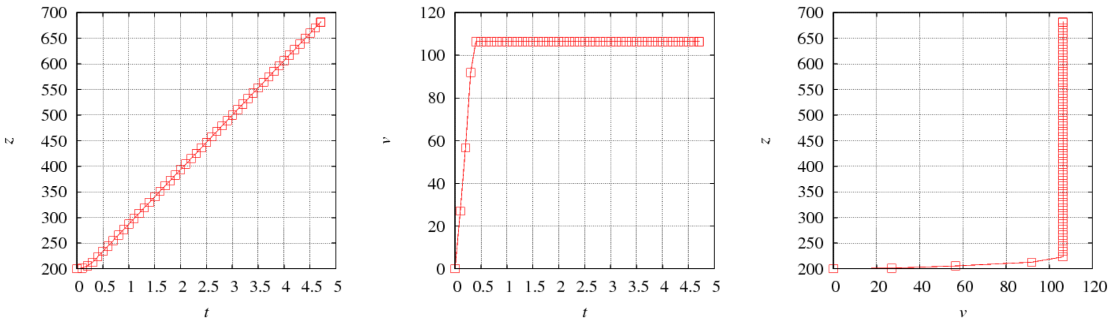

Figure D2. Fluid element position *z* (right) and velocity *v* (center) as function of time *t* and *z* against *v* for Well-B (discharged).

--

Previous version of this work is: Implementation of Single Fluid Volume Element Method in Predicting Flow Occurrence of One-Dimensional Geothermal Well, viXra:1504.0201v1 | 2015-04-26.