NUMERICAL SIMULATION OF RADIAL NON-NEWTONIAN FOAM FLOW IN A RESERVOIR

M. BOS [1]

Prof.dr. W.R. Rossen [2] & MSc. R.O. Salazar [3]

CONTENTS

ABSTRACT

Non-Newtonian foam flow in a reservoir can be modeled numerically by discretization of the corresponding analytical formulas. The injection of foam is compared to the injection of water by comparing the injection pressures, which is represented as a dimensionless pressure rise at the injection well. The model first applies the forward-difference method to compute the changes in water saturation over space and time as foam is injected. These changes in water saturation are related, via Darcy's Law, to changes in dimensionless pressure. The non-Newtonian foam behavior is implemented in the model by making the gas relative permeability a function of position in radial flow, based on the exponent defined for a power law fluid.

The validity of the model is assessed by a comparison with an analytical model using the method of characteristics to simulate Newtonian foam flow. This model was created by A.H. Al Ayesh. From this comparison it follows that the numerical model converges to a correct solution for sufficient fine discretizations. Finer discretizations do however introduce drawbacks, such as long computation times and high computer-memory requirements. Another drawback of the numerical model is the inevitable error that is introduced by a numerical artifact in the computation of the total relative mobility in each grid block at the front as foam advances. This error can only be reduced by even-finer discretizations.

The validity of the model for non-Newtonian foam flow simulations is not assessed directly in this thesis. But the model is expected to have similar or coarser grid-refinement criteria for shear-thinning foam flow, and finer or similar grid refinement criteria for shear-thickening foam flow.

[1] *Student Applied Earth Sciences, University of Technology Delft, The Netherlands*
[2] *Professor and Chair of Petroleum Engineering, University of Technology Delft, The Netherlands*
[3] *PhD Petroleum Engineering and Geosciences, University of Technology Delft, The Netherlands*

1 INTRODUCTION

Currently, about 9% of the world's oil production, and virtually all 'enhanced oil recovery', comes from injecting gases, primarily steam and carbon dioxide, into oil reservoirs.[1,2] In enhanced oil recovery, one can use steam or CO_2 injection to sweep a reservoir of virtually all its oil. However in practice this process often has a low efficiency due to the low viscosity of the injected gases and the geological differences in the subsurface. Since this so-called low 'sweep efficiency' not only causes oil to be left behind, but also creates operational problems, it is worthwhile investigating the usage of foam injection in order to increase this efficiency.[3]

Foam stability is defined by the capillary pressure; p_c. If the capillary pressure is too high, lamellae break and the foam collapses.[4] The value of capillary pressure at which this occurs is referred to as the 'limiting capillary pressure'; p_c^*.[5,6,7] As a result the state of foam in the reservoir is dependent on the value of p_c in the reservoir. In a reservoir the value of p_c is dependent on the water saturation, S_w. Because of this dependency, the flow rate can be represented as a function of water saturation instead of capillary pressure.

The value of water saturation at which foam abruptly weakens is referred to as 'fmdry' in this paper. This value is a constant for a Newtonian foam. But for non-Newtonian foams the value of 'fmdry' changes with superficial velocity due to the shear-thickening or -thinning behavior of the foam. This means that in steady, radial flow, as is discussed in this paper, 'fmdry' is a function of distance from the well.

2 SCOPE

This research aims to further investigate the propagation or flow of non-Newtonian foams in a reservoir. This includes the case of shear-thickening and that of shear-thinning foams. These cases are currently analytically analyzed by ongoing research.[8] In this paper it will be attempted to evaluate these cases numerically, and compare the findings with analytical results. Hence, it will be attempted to verify the approach, by achieving the same results, analytically and numerically.

3 METHODS

In order to evaluate the cases numerically, time and space are discretized. This is represented by using matrices with a certain number of rows and columns to represent a certain number of time- and space-steps. After setting this up the water saturation, S_w, is calculated at each position in time and space. When the values for S_w are known, the flow rates within the reservoir will be known. Then by using Darcy's Law a comparison will be made in pressure, showing how much the pressure at the well will increase as compared to a scenario without the use of foam. These steps will be elaborated in more detail in the next paragraphs.

DISCRETIZATION In setting up the model there has to be a balance between letting the amount of space- and time-steps approach infinity, and accounting for computational time. Besides this fact the model is created such that the discretization for

1 G. Moritis, Oil & Gas Journal, Issue 51 (1992).
2 P. Jacquard, Proc 6th European Symposium on Impr. Oil Recovery, Stavanger, Norway, May 21-23, 1991.
3 W.R. Rossen, Foams in Enhanced Oil Recovery, 1996.
4 A.J. Jimenez and C.J. Radke, in Oil-Fiel Chemistry: Enhanced Recovery and Production Stimulation (J.K. Borchardt and T.F. Yen, eds.) ACS Symposium Ser. 396. Am. Chem. Soc., Washington D.C., 1989.
5 Z.I. Khatib, G.I. Hirasaki, and A.H. Falls, SPE Reserv. Eng. 3: 919 (1988).
6 R.A. Ettinger and C.J. Radke, SPE Reserv. Eng. 7: 83 (1992).
7 P. Persoff, C.J. Radke, K. Pruess, S.M. Benson, and P.A. Witherspoon, SPE Reserv. Eng. 6:365 (1991).
8 Research done by S. ter Haar & C.G. Ponners, supervised by Prof.dr. W.R. Rossen.

time should be much smaller than the discretization for space. Hence, the model accounts for this dependence by generating the discretization for time automatically based on the discretization in space, based on a so called dependence factor. This dependence factor determines how much smaller the discretization in time should be. In doing so the model automatically remains numerically stable for each space discretization. The model works such that it starts with a linear point spacing between the well, r_w, and the outer radius of the reservoir, r_{max}. This spacing is converted to dimensionless distance, x_D, using Equation 1. This is done because analytical models also use dimensionless distance. Next the discretization in time is based on the grid for x_D.

It was decided that the grid for r should not have a logarithmic point scaling; this would decrease the smallest value of dx significantly. This means that a much finer discretization in time is required for numerical stability, resulting in much longer computation times and computer-memory requirements.

$$x_D \equiv \frac{r^2 - r_w^2}{r_{max}^2 - r_w^2} \tag{1}$$

FOAM PROPAGATION Having boundary conditions along with matrices that allow for numerical computations, the changes in S_w will be tracked through time and space. In order to do so, Equation 2, which is an analytical equation, needs to be evaluated in a numerical manner. Via discretization and rearrangement this equation then becomes Equation 3, which is used to solve the matrices.

$$\frac{\partial S_w}{\partial t} + \frac{\partial f_w}{\partial x} = 0 \tag{2}$$

$$S_{wi}(t + \Delta t) = S_{wi}(t) + \Delta t\left(\frac{f_w(S_{wi-1}(t)) - f_w(S_{wi}(t))}{\Delta x}\right) \tag{3}$$

As can be seen in Equation 3, the value of S_w in a specific grid block at a specific time step is calculated based on the value of S_w at the previous time step in that same grid block and an additional value which is calculated based on the flow in and out of that grid block at the previous time step. These values for flow are subsequently dependent on the relative permeability of water, the relative permeability of gas, the viscosity of water, and the viscosity of gas. The equation for f_w is shown in Equation 4. It should be noted here that the representation of a shear-thickening or shear-thinning foam is done, not by letting the viscosity change, but by letting the relative permeability change. This is most clearly understood when looking at the equation for the relative permeability, which is shown in Equation 5. In this equation, x represents dimensionless distance from the injection well, and, as can be seen, the value of 'fmdry' changes with x. Each case of a shear-thickening or -thinning foam is assigned a different range of 'fmdry' values as a function of x, which, as explained earlier, represent the different limiting capillary pressures. The fmdry(x) functions are provided in tabular form by R.O. Salazar; the derivation of these functions is outside the scope of this study. It should however be noted that 'fmdry' changes with velocity, which means it changes according to $\frac{1}{r}$, which subsequently means it changes approximately according to $\frac{1}{\sqrt{x}}$. Thus the behavior of 'fmdry' is extremely nonlinear in terms of x. Note that in Equation 5 the last term is the same at the previous term, but evaluated at S_{wr}, which is the residual water saturation. This term is added, as suggested by Namdar Zanganeh et al. (2011), to

compensate for foam not collapsing completely at residual water saturation in the model.[9]

$$fw = \frac{\frac{Krw(S_w)}{\mu_w}}{\frac{Krg(S_w,x)}{\mu_g} + \frac{Krw(S_w)}{\mu_w}} \tag{4}$$

$$Krg = Krg^o(S_w)(1 + fmmob(\frac{\arctan(epdry(S_w - fmdry(x)))}{\pi} - \frac{\arctan(epdry(S_{wr} - fmdry(x)))}{\pi})) \tag{5}$$

PRESSURE CHANGES In addition to the propagation of foam, we want to simulate how the pressure changes in the reservoir as the foam prograde. The goal here is to find the pressure at the well based on a reference pressure at the outer location of the radius of interest. This value for pressure at the outer radius is known and fixed. The pressure at the well is then computed numerically by applying Darcy's Law, working from the outer radius inward, using the computed values for S_w as a function of position. Darcy's Law, along with it's numerical approximation after discretization and rearrangement, are given in Equation 6 and 7 respectively. The total relative mobility, λ_{rt}, which is used in these equations, is given in Equation 8.

$$\frac{Q}{2\pi rh} = -K\lambda_{rt}(\frac{dP}{dr}) \tag{6}$$

$$P_{n-1} = P_n + (\frac{Q}{4\pi hK})(\frac{1}{\lambda_{rt}(r_n, S_{wn})} + \frac{1}{\lambda_{rt}(r_{n-1}, S_{wn-1})})ln(\frac{r_n}{r_{n-1}}) \tag{7}$$

$$\lambda_{rt} = \frac{K_{rw}(S_w)}{\mu_w} + \frac{K_{rg}(S_w, r)}{\mu_g} \tag{8}$$

Once the changes in pressure over time and space are calculated, these changes can be translated to dimensionless pressure changes over time. This dimensionless value represents the value of pressure at the well as compared to the value of pressure at the well when injecting water at the same rate into a water-saturated reservoir. This latter scenario without foam is referred to as the 'waterflood' scenario. The dimensionless pressure can be calculated as shown in Equation 9; the pressure at the well minus the pressure at the outer radius, divided by the same values, but for the waterflood scenario.

$$P_D = \frac{(P_w - P_e)}{(P_w - P_e)_{Waterflood}} \tag{9}$$

BOUNDARY CONDITIONS As we are trying to simulate gas injection in a SAG foam process, in which gas and surfactant solution meet in the reservoir itself, there are certain boundary conditions that can be applied. First of all, in order to account for this injection process, it is assumed that f_w is 0 at the well at all times. The second assumption that was made governs the initial conditions of the reservoir. It is assumed that initially S_w is 1 everywhere in the reservoir. These two boundary conditions are shown in Equation 10.

$$\begin{aligned} f_w &= 0 \qquad \text{for: } [x = 0,\ \text{all } t] \\ S_w &= 1 \qquad \text{for: } [t = 0,\ \text{all } x] \end{aligned} \tag{10}$$

9 A.H. Al Ayesh, Optimal SAG Design in Heterogeneous Reservoirs - Effect of Permeability on Foam Diversion, pg. 30, 2016.

4 RESULTS

VALIDITY OF THE NUMERICAL MODEL First of all the validity of the model should be assessed. In order to do so the numerical model will be compared to the Matlab code that was written by Ahmed H. Al Ayesh[10]. This is an analytical script that, amongst other things, allows for calculating P_D for Newtonian foam flow in a system up to four layers, using the method of characteristics to solve Equation 2. For this comparison all these layers are set to have the same properties, such that it resembles a uniform subsurface, as is the case in the numerical model. Besides that the value of n in the numerical script is set to 1, such that these models both simulate Newtonian foam flow. These and other parameters were set to be the same for both models. An overview of these parameters can be found in Table 1. The results of the simulations done with Ahmed H. Al Ayesh's code can be found in Figures 1 and 3. The results of the simulations done with the numerical code can be found in Figures 2 and 4.

Table 1: Input Parameters

Parameter	Value	Units	Parameter	Value	Units
n	1.00	-	μ_w	0.001	mPa.s
Krw00	0.39	-	μ_g	0.00002	mPa.s
n_w	2.86	-	r_w	0.1	m
Krg00	0.59	-	r_{max}	100	m
n_g	0.70	-	fmmob	47700	-
S_{wr}	0.25	-	epdry	400	-
S_{gr}	0.20	-	$fmdry_{r_{max}}$	0.271	-

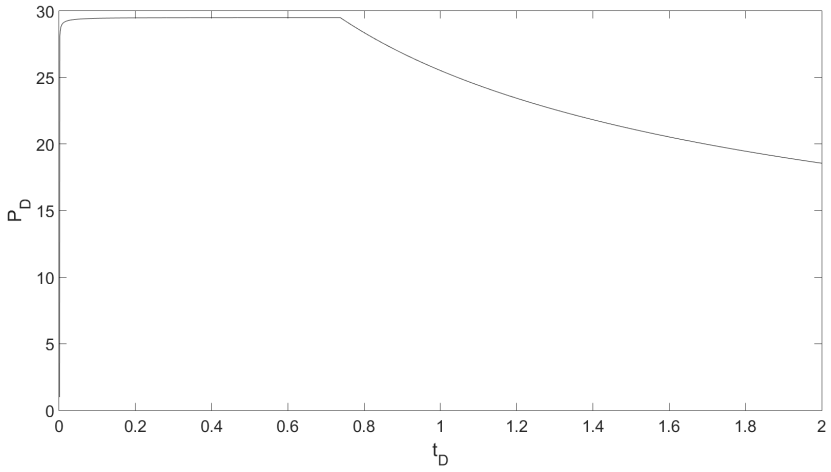

Figure 1: A.H. Al Ayesh's Model [t_D(max)=2.00]

10 A.H. Al Ayesh, Optimal SAG Design in Heterogeneous Reservoirs - Effect of Permeability on Foam Diversion, Appendix C: Matlab Code, 2016.

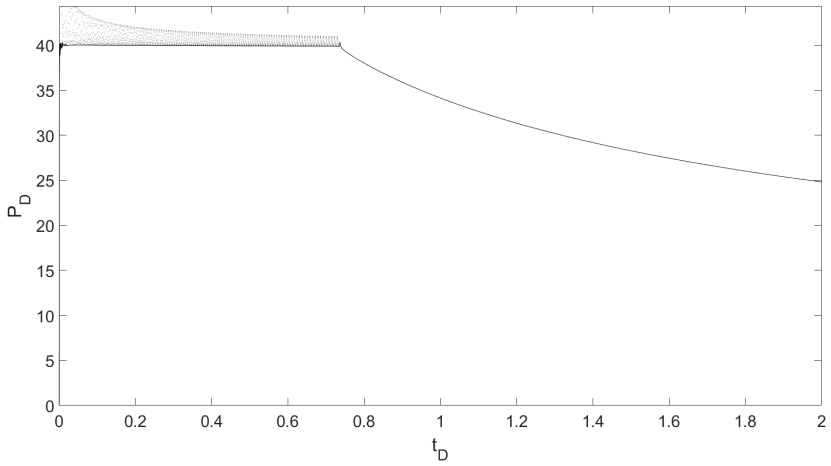

Figure 2: Numerical Model [t_D(max)=2.00, No. of grid points=200]

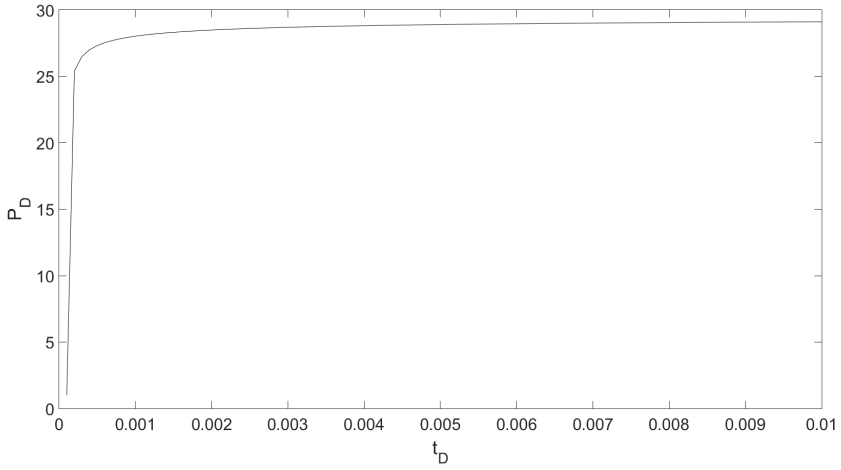

Figure 3: A.H. Al Ayesh's Model [t_D(max)=0.01]

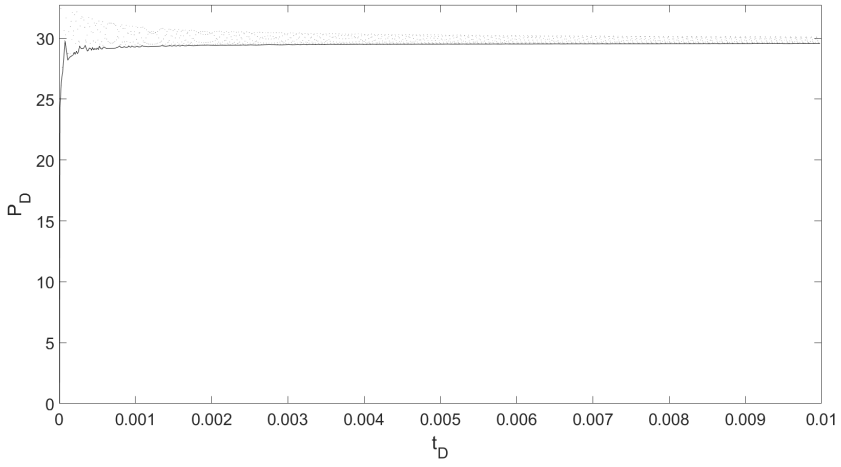

Figure 4: Numerical Model [t_D(max)=0.01, No. of grid points=2500]

In verifying the model it becomes clear Figure 2 strongly resembles Figure 1. The moment of foam breakthrough at r_{max} is exactly the same; t_D=0.72. However the

absolute values are a factor of a third higher. When going to larger discretizations such as in Figure 4, the solution clearly yields the same plateau level as in the solution of Ahmed H. Al Ayesh. Unfortunately the value of t_D(max) has to be decreased in order to compensate for the computation time and computer-memory requirements when using a finer grid. The impact of using a finer discretization on the memory requirements is shown in Figure 5. This figure shows that memory requirements increase exponentially with grid refinement.

When comparing Figures 3 and 4, it is clear that the two solutions approach the same value. The reason that the solutions do not agree very well at early times might have two reasons. At early times in the numerical model, the foam front occupies relatively few grid blocks. Thus the refined model still posses a relatively coarse grid at early times. Besides that the analytical model has a larger dt in these simulations, which also explains why it has no solution for t_D<0.0001. As time increases the solutions agree well. For discretizations coarser than 2500 grid points the model appears to overshoot the initial increase in P_D. When comparing these findings with a simulation that has a discretization of 3000 points, which is shown in Figure 6, it becomes clear that there is relatively little difference in the plateau level between the case for 2500 grid points or that of 3000 grid points. Hence it can be concluded that finer discretizations are indeed converging to the same solution, at least for grids finer than 2500 points, which is the same solution as the analytical solution provided in Figure 1. It can also be concluded that the model appears to overshoot the initial increase in P_D for discretizations coarser than 2500 grid points.

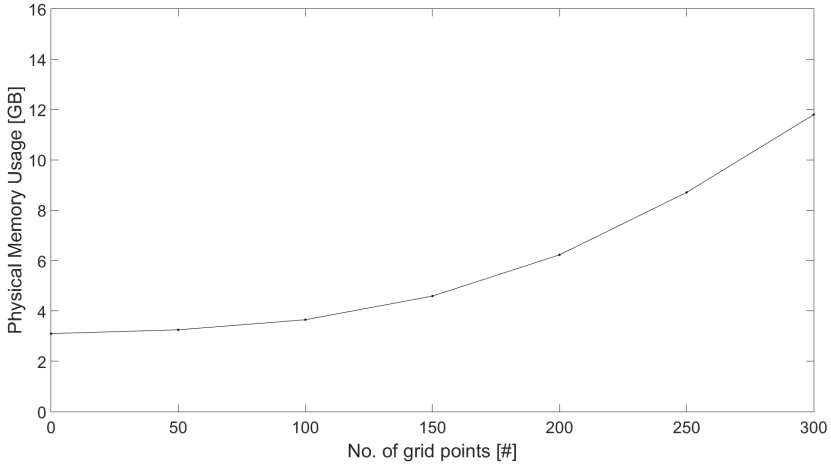

Figure 5: Memory Usage of the Model on a Computer with 16GB RAM Memory [t_D(max)=2.00, 'dep'=7]

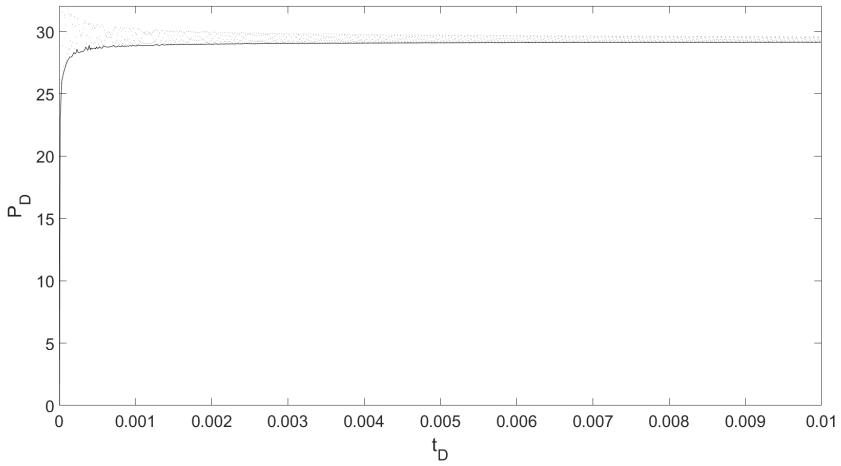

Figure 6: Numerical Model [t_D(max)=0.01, No. of grid points=3000]

In concluding that this specific case is stable and converges to the correct solution for discretizations finer than 2500 points it is relevant to ask ourselves, keeping the scope that was set in mind, how this criteria translates to non-Newtonian problems. The first thing that can be said about this, is the fact that the vector that is used to represent the changes in 'fmdry' stays mostly the same for non-Newtonian cases. That is, the vectors and the matrices remain of the same format and size; only the values change. Given this fact, it is expected that the criteria would not change too much when the exponent changes in the formula for power-law foams. That being said, another thing that changes with a changing 'fmdry' is the actual value of P_D. In the case of a shear-thinning foam this results in higher velocities near the well and thus smoother behavior there. In the case of a shear-thickening foam however, the viscosities near the well increase, resulting in lower mobilities, which then subsequently result in higher pressures. This means that the values for P_D will increase, and that the initial increase in P_D at very small t_D will be greater. Hence it is more likely that the model will 'overshoot' this initial increase for shear-thickening foams. This behavior was also observed in Figure 2. However in the case of shear-thickening foam simulations, it might even do so with finer discretizations.

NON-NEWTONIAN FOAM SIMULATIONS A variety of simulations for non-Newtonian foam flow were done, though we have no analytical results yet to compare these with. These simulations can be found in Appendix A. The different values for n that are used for these simulations, were acquired by R.O. Salazar, and were used in the work of S. ter Haar and C.G. Ponners as well. All other parameters required for the simulations were kept the same as in Table 1.

5 DISCUSSION

NUMERICAL ARTIFACTS One of the largest drawbacks of using this model, next to the large computation time and computer-memory usage as were discussed earlier, are artifacts that are introduced by solving the problem numerically. These artifacts are discussed in a paper by W.R. Rossen[11]. In this paper it is explained how these artifacts are introduced in the simulations, by looking at the term for λ_{rt}, which was shown in Equation 8 of this paper. The relation between this term, the total relative mobility, λ_{rt}, and the water saturation, is shown in Figure 7. This Figure was originally presented in the paper by W.R. Rossen and illustrates what changes in mobility take place at the foam front. In an injection process there is a 'shock'

11 W.R. Rossen, Numerical Challenges in Foam Simulation: A Review, SPE, pg. 3-5, 2013.

going from the state ahead of the foam, with higher water saturations, to the foam bank, which has lower water saturations. In a finite-difference simulation however, each grid block at the foam front passes through all these saturations when foam prograde through the grid blocks. This results in an overestimation of mobility reduction at the foam front, because grid blocks go all the way down the curve that is shown in Figure 7.

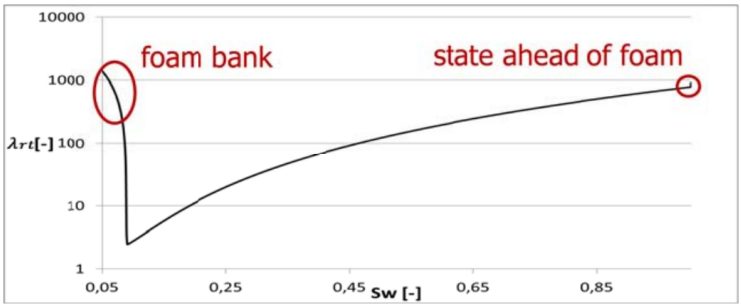

Figure 7: The total relative mobility λ_{rt} as a function of water saturation for the model in Fig.1. In a SAG Process, there is a shock from the initial state ahead of the shock past the water saturations with lowest mobility. In a finite-difference simulation, each grid block passes through all of the intermediate water saturations as the foam front enters the grid block. (Rossen, 2013)

When the decrease in mobility is overestimated, the pressure will be overestimated as well. The effect of grid blocks passing through all water saturations will cause different magnitudes of overestimation. Each time a grid block slides down the curve in Figure 7 there is a certain moment of maximum overestimation for that specific grid block. This can be observed in Figures 2, 4, and 6. In these figures there is a scatter of data points above the actual solution. These data points all have different magnitudes of overestimations; if' the dots would be connected a fluctuating function would become visible. When looking at these figures it becomes clear that the solution that is presented in this paper is simply the minima of these values. This was done in an attempt to minimize the error that is introduced by this numerical artifact. However it should be noted that, as becomes clear from Figure 7, a finite-difference solution can never be completely without these errors. The effect is especially great when the foam front is near the injection well, because the model still posses a coarse grid for these early times as was explained earlier. However, a finer grid spacing does indeed reduce the magnitude of this artifact. W.R. Rossen concluded in his paper: 'Refining the grid does not eliminate the artifact; each grid block must still pass through the minimum in mobility in Fig. 2 (Figure 7 here), but refining the grid does reduce its impact on overall injectivity.'. This conclusion is also in line with the differences between the simulations in Figures 2, 4, and 6.

VALIDITY FOR NON-NEWTONIAN FOAM As was discussed in the results section of this paper under the paragraph 'Validity of the Numerical Model', the numerical model is able to match the results that were acquired using the method of characteristics as implemented by A.H. Al Ayesh, at least after a brief period of injection. This validates the numerical model in simulating Newtonian foam flow, if used along with certain discretization criteria. There is reason to believe that this implies that the numerical model produces valid results for non-Newtonian foam flow as well, with more or less the same discretization criteria. It is however possible that shear-thickening foam requires a finer grid to achieve valid solutions. But such remarks are all speculation at this point, and no such comparisons have been done to validate these expectations. This means that the phrase "more or less the same discretization criteria" becomes quite a broad statement, and thus actual applicability of the model for non-Newtonian foam flow is unknown.

6 CONCLUSION

Numerical analysis, and more specifically forward-difference, does prove to be a valid method to simulate foam flow in a heterogeneous reservoir. The applicability of these methods is questionable however. Using numerical methods to analyze foam flow seems to yield a number of practical problems such as long computation times, and high computer memory requirements. In addition there is a certain error introduced in total mobility of the foam by discretizing the problem, that cannot be removed; it can only minimized by using a finer grid. It should be said that time at hand was short, and that the model could perhaps be optimized further, which might result in shorter computation times. It should also be noted that even though the model does allow for non-Newtonian foam flow, this has not been compared with trusted, analytical, results yet. Keeping these things in mind the real potential of the numerical analysis is perhaps still somewhat unknown, however there seems to be little advantage thus far in using a numerical approach instead of a analytical approach in simulating foam flow in a reservoir.

7 APPENDICES

Appendix A: Simulation Overview

NEWTONIAN FOAM [N=1.00] Simulations for Newtonian foam. Using all the parameters shown in Table 1 other than 'n'.

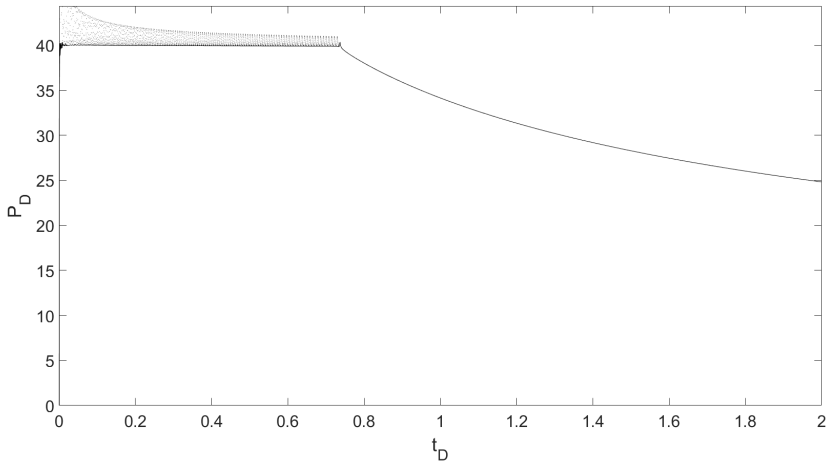

Figure 8: Newtonian Foam [n=1.00, t_D(max)=2.00, No. of grid points=200]

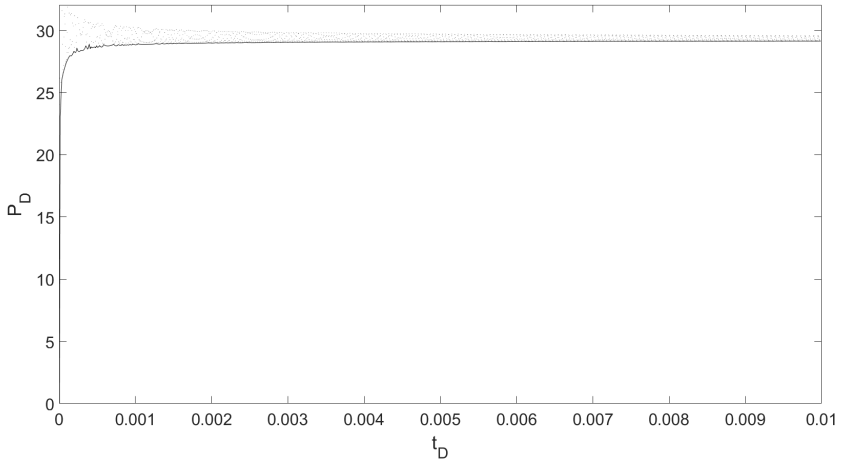

Figure 9: Newtonian Foam [n=1.00, t_D(max)=0.01, No. of grid points=3000]

SHEAR-THINNING FOAM [N=0.77] Simulations for shear-thinning foam. Using all the parameters shown in Table 1 other than 'n'.

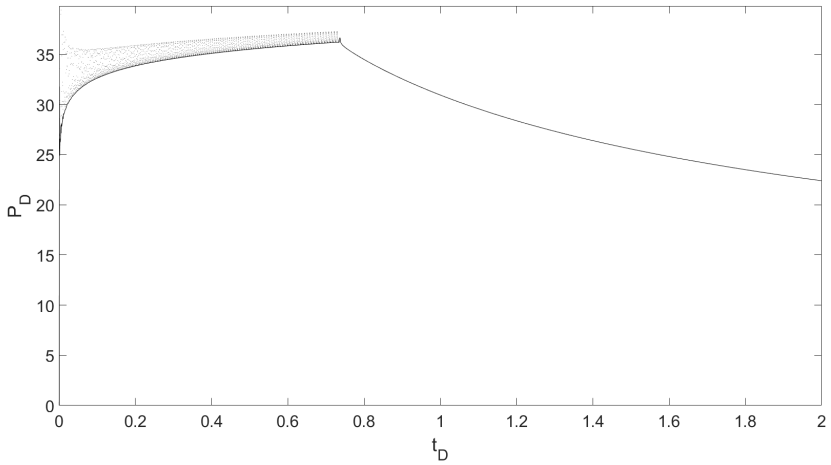

Figure 10: Shear-Thinning Foam [n=0.77, t_D(max)=2.00, No. of grid points=200]

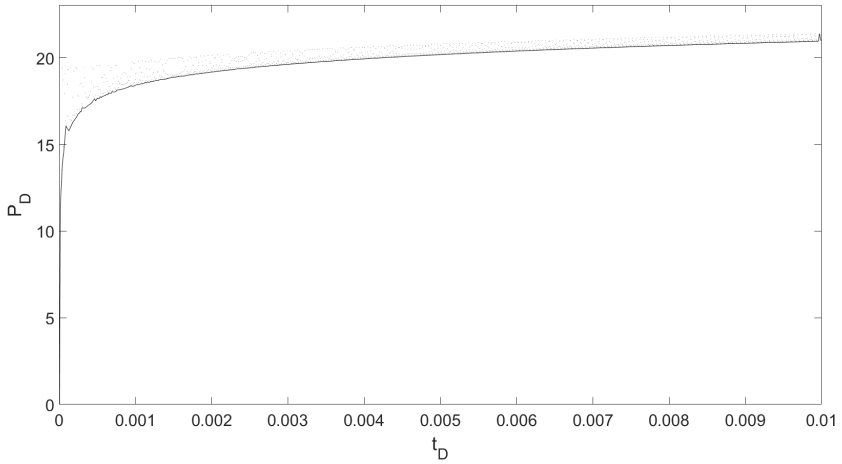

Figure 11: Shear-Thinning Foam [n=0.77, t_D(max)=0.01, No. of grid points=3000]

SHEAR-THICKENING FOAM [N=1.28; 1.30 ; 1.69; 1.70; 2.11] Simulations for shear-thickening foam. Using all the parameters shown in Table 1 other than 'n'.

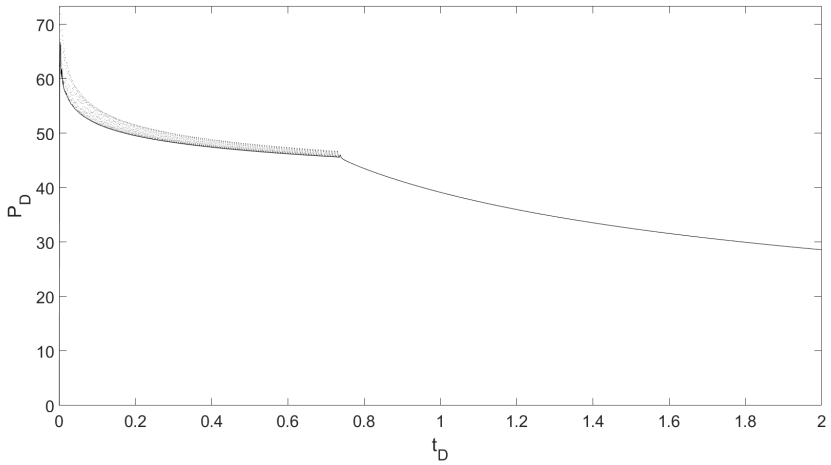

Figure 12: Shear-Thickening Foam [n=1.28, t_D(max)=2.00, No. of grid points=200]

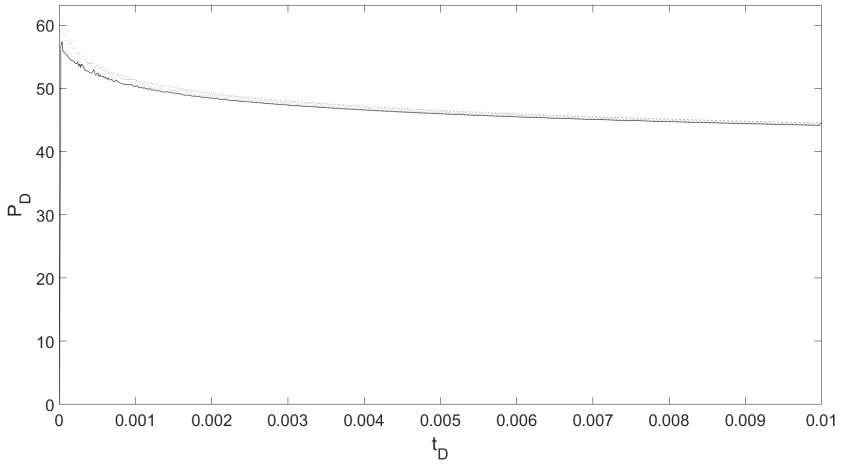

Figure 13: Shear-Thickening Foam [n=1.28, t_D(max)=0.01, No. of grid points=3000]

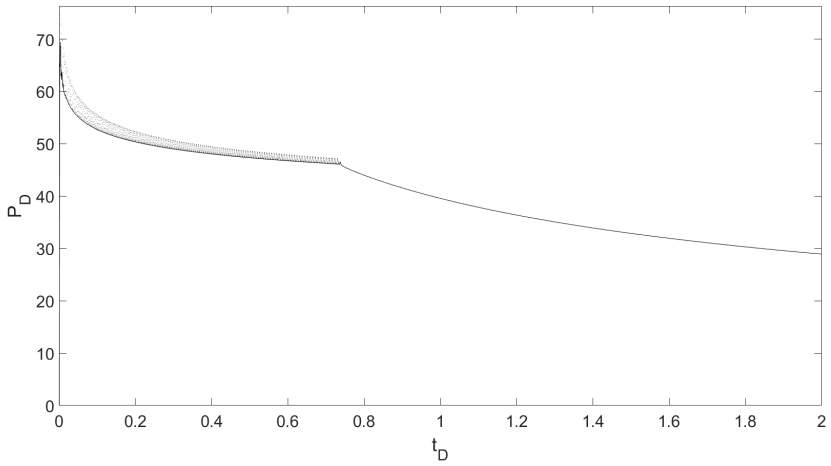

Figure 14: Shear-Thickening Foam [n=1.30, t_D(max)=2.00, No. of grid points=200]

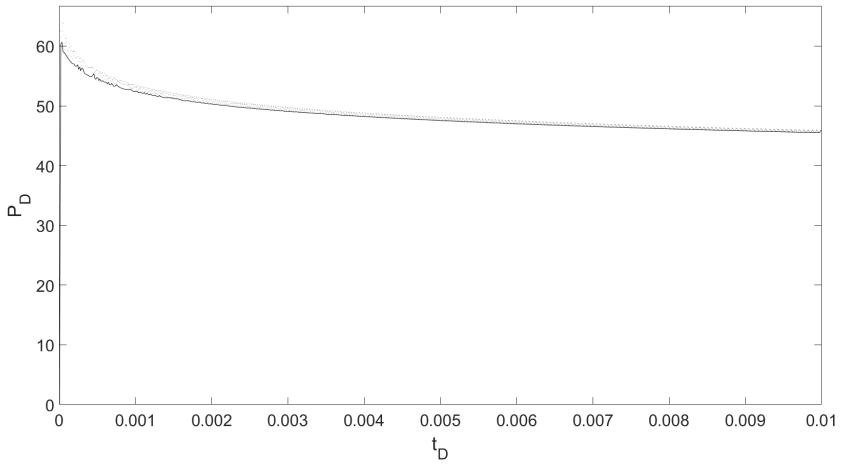

Figure 15: Shear-Thickening Foam [n=1.30, t_D(max)=0.01, No. of grid points=3000]

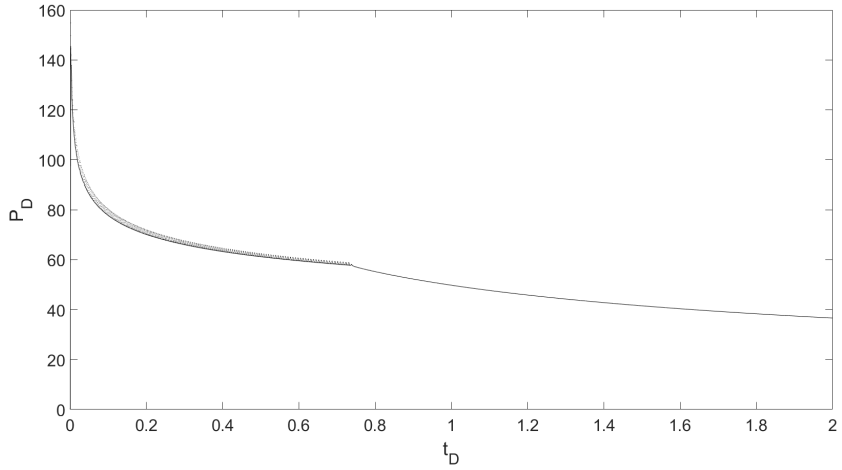

Figure 16: Shear-Thickening Foam [n=1.69, t_D(max)=2.00, No. of grid points=200]

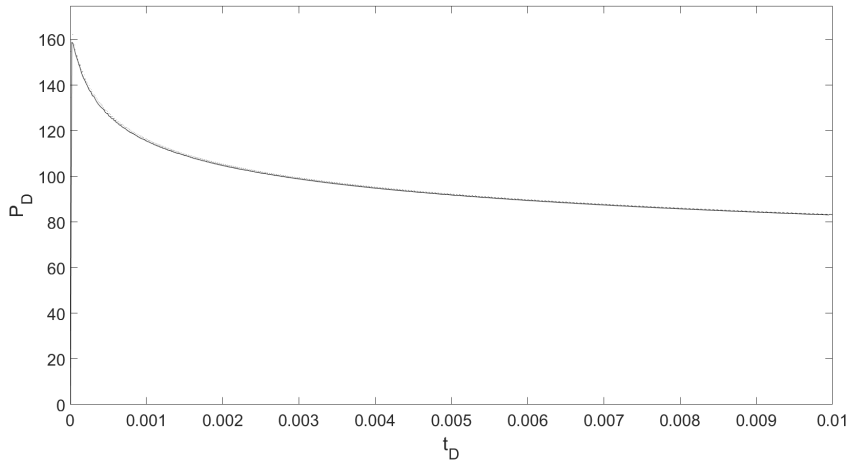

Figure 17: Shear-Thickening Foam [n=1.69, t_D(max)=0.01, No. of grid points=3000]

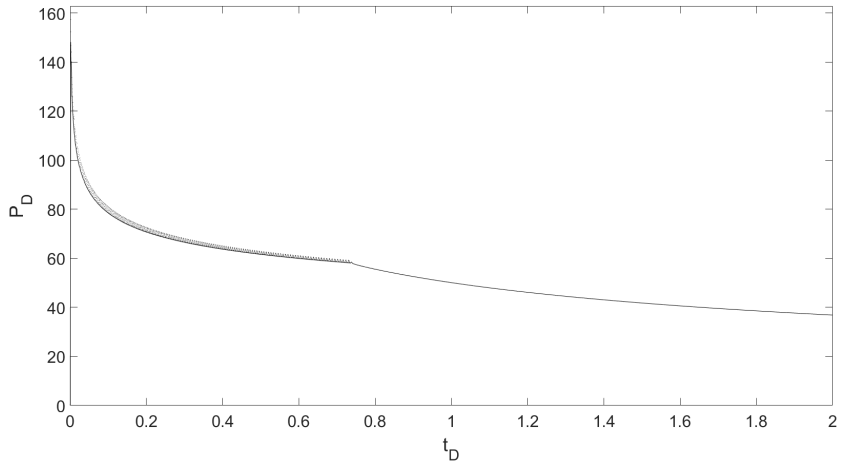

Figure 18: Shear-Thickening Foam [n=1.70, t_D(max)=2.00, No. of grid points=200]

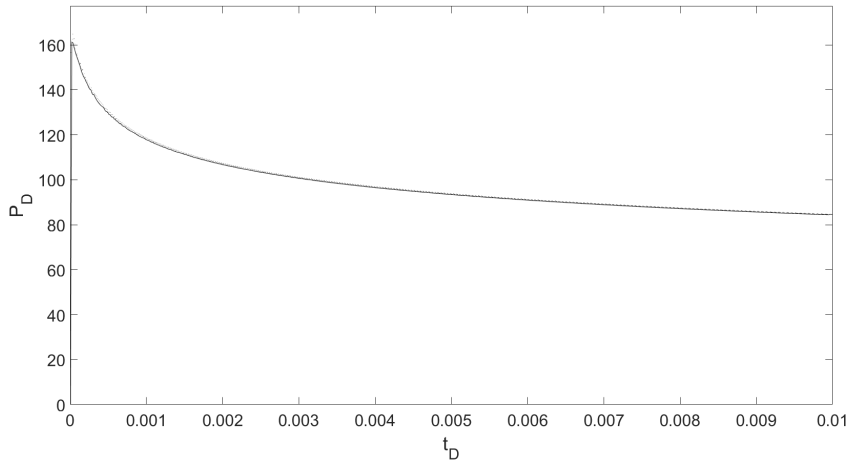

Figure 19: Shear-Thickening Foam [n=1.70, t_D(max)=0.01, No. of grid points=3000]

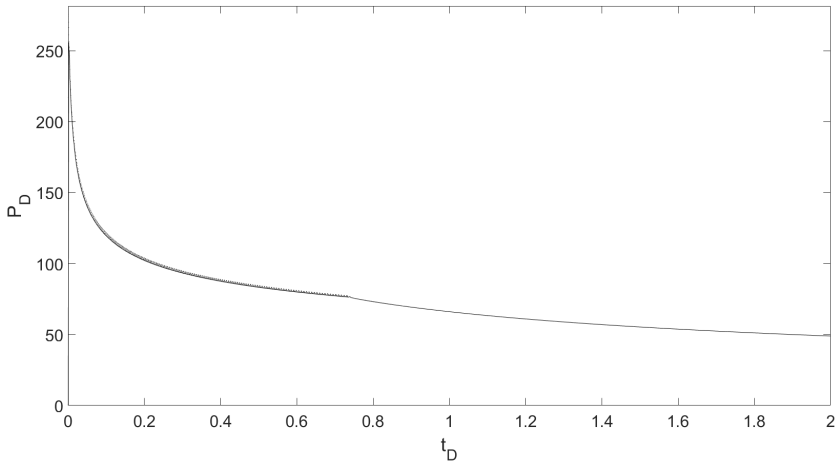

Figure 20: Shear-Thickening Foam [n=2.11, t_D(max)=2.00, No. of grid points=200]

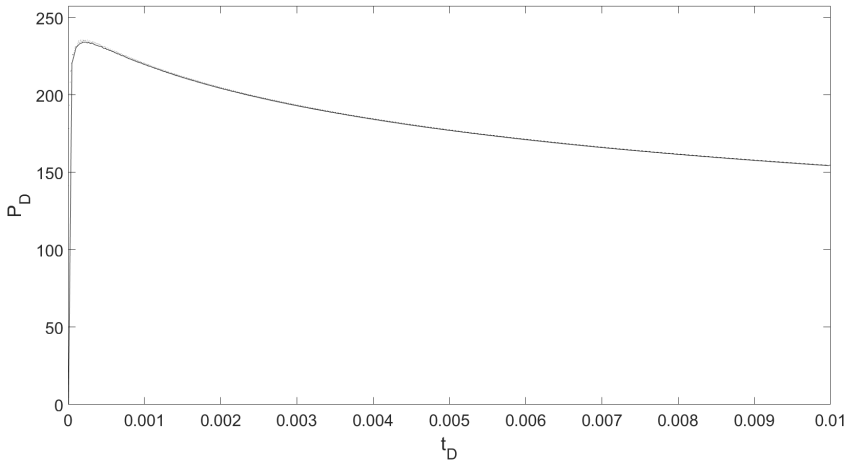

Figure 21: Shear-Thickening Foam [n=2.11, t_D(max)=0.01, No. of grid points=3000]

FOAM COMPARISON [N=0.77; N=1.00; N=1.69] Comparing Newtonian, shear-thinning, and shear-thickening foam for the parameters given in Table 1.

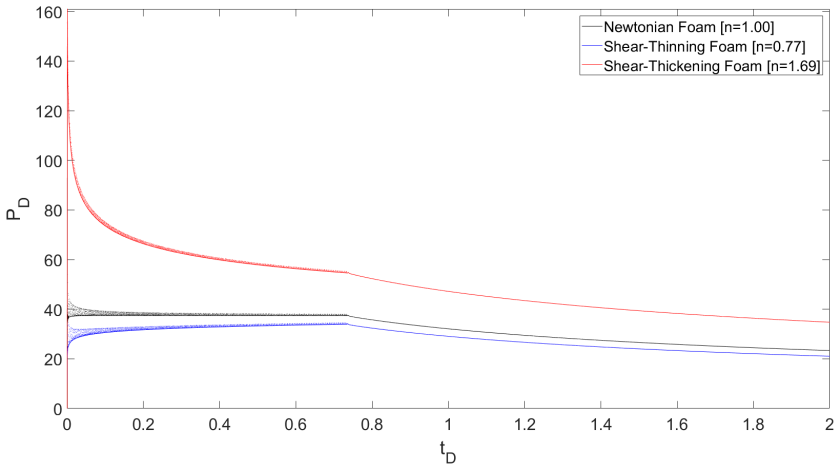

Figure 22: Foam Comparison [t_D(max)=2.00, No. of grid points=300]

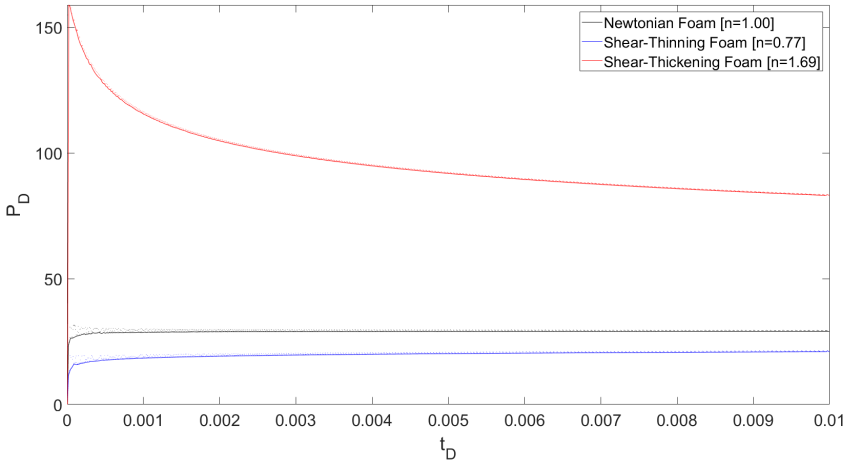

Figure 23: Foam Comparison [t_D(max)=0.01, No. of grid points=3000]

Appendix B: Matlab Code

The code for the numerical model, written in Matlab. Note that only the sections 3 and 4 govern the actual model. Sections 1 and 2 process input parameters, and sections 5, 6, and 7 visualize the resulting data.

```
%-------------------------------------------------------------------------%
% Numerical Simulation of Radial Non-Newtonian Foam Flow in a Reservoir   %
% Student: M. Bos                                                         %
% Supervisors: Prof.dr. W.R. Rossen & MSc. R.O. Salazar                   %
% June 2017                                                               %
%-------------------------------------------------------------------------%

clear all
close all
clc

%% [1] INPUT
muw=0.001;          % Fluid property
mug=0.00002;        % Fluid property
Krw00=0.39;         % Corey parameter
nw=2.86;            % Corey parameter
Krg00=0.59;         % Corey parameter
ng=0.70;            % Corey parameter
Swr=0.25;           % Residual saturation
Sgr=0.20;           % Residual saturation
fmmob=47700;        % Foam model parameter
epdry=400;          % Foam model parameter
fmdryrmax=0.271;    % Foam model parameter
rw=0.1;             % Welbore radius
rmax=100;           % Reservoir radius

C=1000;             % Constant: Q/2pihK
Pi=0;               % Reference pressure at rmax

n=1.00;             % Power law fluid exponent.

rn=200;             % No. of grid points
tmax=2.00;          % Maximum dimensionless time

dep=10;         % [~7+]    Factor of dependance: dt=(min(dx))/dep
Pstep=50;       % [25-75]  Ratio between integrated rows for P and total rows
visstep=2000;   % [2000]   Ratio between visualized rows and total rows

%% [2] PREALLOCATION AND SETTING UP DEPENDENT PARAMETERS
fprintf('Preallocation, please wait...\n')
r=linspace(rw,rmax,rn);
xn=rn;
x=(r.^2-rw^2)/(rmax^2-rw^2);
dx=zeros(1,xn-1);
for i=1:xn-1;
   dx(i)=abs(x(i+1)-x(i));
end
dt=min(dx)/dep;
tn=str2double(num2str(round((tmax/dt),0)+1));
t=linspace(0,tmax,tn);
tP=zeros(1,round(tn/Pstep,0));
for j=1:tn/Pstep;
    tP(j)=t(j*Pstep);
end
```

```
P=zeros(round(tn/Pstep,0),xn);
P(:,xn) = Pi;
Pf=zeros(1,xn);
Pf(1,xn) = Pi;
dlP=zeros(1,round(tn/Pstep,0));
Sw = zeros(tn,xn);
Sw(1,:) = 1;
Sw(2,:) = 1;
fw = zeros(tn,xn);
Krw=zeros(tn,xn);
Krg0=zeros(tn,xn);
Krg=zeros(tn,xn);
C4=(fmdryrmax-Swr)/(rmax)^((n-1)/nw);
fmdry=zeros(1,numel(r));
for i=1:numel(r)
    fmdry(i)=Swr+C4*r(i)^((n-1)/nw);
end
clearvars i
timestr=sprintf('n = %.2f, 0 < t < %.2f',n,tmax);
timestr2=sprintf('%.2f',tmax);
titlestr=sprintf('n = %.2f',n);

%% [3] SOLVING NUMERICALLY FOR S_w & INTEGRATING THE RESULT FOR P
for j=1:tn; %S_w, for-loop over time.
    for i=2:xn; %S_w, for-loop over space.
            Krw(j,i)=real(Krw00*((Sw(j,i)-Swr)/(1-Swr-Sgr)).^nw);
            Krw(j,1)=Krw(j,2);

            Krg0(j,i)=real(Krg00*((1-Sw(j,i)-Sgr)/(1-Swr-Sgr)).^ng);
            Krg(j,i)=Krg0(j,i)*(1+fmmob*((0.5+(atan(epdry*(Sw(j,i)-fmdry(i)))/pi
                ))-(0.5+(atan(epdry*(Swr-fmdry(i)))/pi)))).^-1;
            Krg(j,1)=Krg(j,2);

            fw(j,i)=(Krw(j,i)/muw)/((Krg(j,i)/mug)+(Krw(j,i)/muw));

            Sw(j+1,i)=dt*((fw(j,i-1)-fw(j,i))/dx(1,i-1)) + Sw(j,i);
            Sw(j+1,1)=Sw(j+1,2);
    end
    fprintf('%.1f%% - Solving numerically for Sw - rn=%.f, t=%.2f \n',(j/tn)
        *100,xn,tmax)
end

for i=2:xn; % P_waterflood, for-loop over space.
    Pf(1,xn-i+1)=Pf(1,xn-i+2)+transpose(C*0.001*log(x(xn-i+2)/x(xn-i+1)));
    Pf(1,1)=Pf(1,2);
    fprintf('%.1f%% - Integrating for Pw - rn=%.f, t=%.2f \n',(i/xn)*100,xn,tmax
        )
end

for j=1:tn/Pstep; % P, for-loop over time.
    for i=2:xn; % P, for-loop over space.
        P(j,xn-i+1)=P(j,xn-i+2)+transpose(C*(1/(Krw(j*Pstep,xn-i+2)/muw+Krg(j*
            Pstep,xn-i+2)/mug)+1/(Krw(j*Pstep,xn-i+1)/muw+Krg(j*Pstep,xn-i+1)/
            mug))*0.5*log(x(xn-i+2)/x(xn-i+1)));
        P(j,1)=P(j,2);
        dlP(j)=(P(j,1)-P(j,xn))/(Pf(1,1)-Pf(1,xn)); % P_D
    end
    fprintf('%.1f%% - Integrating for P - rn=%.f, t=%.2f \n',(j*Pstep/tn)*100,xn
        ,tmax)
end
clearvars i j
```

```
110
111 %% [4] CORRECTING FOR NUMERICAL ARTIFACTS AT FOAM FRONT BY LOCAL MINIMA
112 fprintf('Correcting numerical artifacts at foam front by local minima, please
        wait...\n')
113 idlP=dlP*-1;
114 [mindlP,mindlPlocs]=findpeaks(idlP);
115 peaktP=tP(mindlPlocs);
116 mindlP=mindlP*-1;
117 fpeaktP(1)=tP(1);
118 fpeaktP(2:numel(peaktP)+1)=peaktP;
119 fmindlP(1)=peaktP(1);
120 fmindlP(2:numel(mindlP)+1)=mindlP;
121
122 %% [5] VISUALISATION OF CHANGES IN S_w OVER TIME AND SPACE
123 fprintf('Visualizing resulting data, please wait...\n')
124 f=figure(1);
125 axes('Parent',f,'position',[0.13 0.39 0.77 0.54]);
126 h=plot(x,Sw(1,:),'k','Marker','.','MarkerEdgeColor','k','MarkerSize',5);
127 axis([0 1 0 1])
128 ylabel('S_w')
129 xlabel('x_D')
130 title(timestr)
131 c=uicontrol('Parent',f,'Style','slider','Position',[81,54,419,23],'Min',1,'Max',
        tn,'Value',1);
132 cgcolor = f.Color;
133 cl1 = uicontrol('Parent',f,'Style','text','Position',[50,54,23,23],'String','0',
        'BackgroundColor',cgcolor);
134 cl2 = uicontrol('Parent',f,'Style','text','Position',[500,54,23,23],'String',
        timestr2,'BackgroundColor',cgcolor);
135 cl3 = uicontrol('Parent',f,'Style','text','Position',[240,25,100,23],'String','t
         = 0.00','BackgroundColor',cgcolor);
136 c.Callback = @(es,ed) {set(cl3,'String',sprintf('t = %.2f',t(round(es.Value))));
137                        set(h,'Ydata',Sw(round(es.Value),:))};
138
139 %% [6] VISUALISATION OF CHANGES IN P OVER TIME AND SPACE
140 e=figure(2);
141 axes('Parent',e,'position',[0.13 0.39 0.77 0.54]);
142 n=plot(x,P(1,:),'k','Marker','.','MarkerEdgeColor','k','MarkerSize',5);
143 hold on
144 plot(x,Pf,'k','Marker','x','MarkerEdgeColor','k','MarkerSize',5)
145 axis([0 1 Pi max(max(P(mindlPlocs(:))))])
146 ylabel('P')
147 xlabel('x_D')
148 title(timestr)
149 legend('P','P_W_a_t_e_r_f_l_o_o_d')
150 s=uicontrol('Parent',e,'Style','slider','Position',[81,54,419,23],'Min',1,'Max',
        round(tn/Pstep),'Value',1);
151 egcolor = e.Color;
152 el1 = uicontrol('Parent',e,'Style','text','Position',[50,54,23,23],'String','0',
        'BackgroundColor',egcolor);
153 el2 = uicontrol('Parent',e,'Style','text','Position',[500,54,23,23],'String',
        timestr2,'BackgroundColor',egcolor);
154 el3 = uicontrol('Parent',e,'Style','text','Position',[240,25,100,23],'String','t
         = 0.00','BackgroundColor',egcolor);
155 s.Callback = @(es,ed) {set(el3,'String',sprintf('t = %.2f',t(round(es.Value)*
        Pstep)));
156                        set(n,'Ydata',P(round(es.Value),:))};
157
158 %% [7] VISUALISATION OF CHANGES IN P_D OVER TIME
159 figure(3)
160 plot(tP(max(mindlPlocs):end),dlP(max(mindlPlocs):end),'k','LineWidth',0.75)
```

```
hold on
plot(tP(1:max(mindlPlocs)-1),dlP(1:max(mindlPlocs)-1),'LineStyle','none','Marker
    ','o','MarkerSize',0.8,'MarkerEdgeColor','none','MarkerFaceColor','k')
hold on
plot(fpeaktP,fmindlP,'k','LineWidth',0.75)
axis([0 max(t) 0 max(max(fmindlP))+0.1*max(max(fmindlP))])
set(gca,'FontSize',24)
xlabel('t_D','FontSize',28);
ylabel('P_D','FontSize',28);
```

Soli Deo Gloria