# INFLUENCE OF TRANSIENT EFFECTS ON FITTING PARAMETERS TO SAG FOAM CORE-FLOODS

By

Marnix Koops

Thesis submitted in partial fulfillment
of the requirements for the degree of

**Bachelor of Science**

in Applied Earth Sciences
at the Delft University of Technology

| | |
|---|---|
| Date: | July 2014 |
| Supervisors: | Prof. dr. W.R. Rossen<br>dr. L. Kapetas |
| Student number: | 4142675 |

**ABSTRACT:** Foam is a promising means of increasing sweep efficiency in enhanced oil recovery processes where gas is injected into a reservoir. However, the effectiveness of foam is hard to predict due to its complex behavior. Therefore accurate modeling is important to effectively predict field performance. This requires correct foam parameters in reservoir simulation. These parameters can be fit to laboratory data where gas is injected into surfactant-saturated rock samples. Foam simulation models can be used to translate experimental laboratory data to the field scale. Unfortunately, the widely used local equilibrium model assumes instant local equilibrium everywhere, while the lab data shows dynamic behavior. Distortion in late-time simulation results of the model could occur as consequence of not having a good fit in the dynamic period, where the chosen foam modeling parameters are incorrect. The aim of this paper is to research how much influence this mismatch has on the fit of the model to the laboratory data, which is taken from literature. Testing will be performed with the use of a one-dimensional dynamic foam model. The findings in this work do not indicate clear evidence that fitting long-time behavior is distorted by disregarding a good fit in dynamic short-time behavior. This means that the researched modeling parameters could be correct in predicting local equilibrium behavior. Even when a good fit for short times where dynamics play a role is disregarded.

## 1. INTRODUCTION

In Enhanced Oil Recovery (EOR) processes gases, such as carbon dioxide or steam, are injected into oil reservoirs to improve oil recovery. The displacement efficiency of the gas is high in the zones where the gas contacts oil (Rossen 1996). However the gas contacts and sweeps only a small fraction of oil in the reservoir, primarily due to poor sweep efficiency. The main reasons for poor sweep efficiency are reservoir heterogeneity, viscous instability and gas gravity override (Lake 1989). Sweep efficiency in EOR processes can be greatly increased by utilizing foam as a mobility-control agent (Rossen 1996). Foam is defined as a dispersion of gas separated by thin liquid films called lamellae (Ashoori 2011a). Foam in porous media reduces the gas mobility by decreasing the gas relative permeability and increasing the gas effective viscosity (G.G. Bernard 1964). When gas gravity override is the main concern, the most effective way to sweep a reservoir is by injection of alternating large slugs of surfactant solution and gas (SAG injection), while maximizing the injection rate (D. Shan 2004).

Comprehension of foam behavior is hindered by its complexity, which in turn influences the capability of designing effective field treatments (J.M. Alvarez 2001). Laboratory experiments where gas is injected into a surfactant filled rock core are done to predict foam performance. The obtained results can be translated into field scale by the use of computer simulations. The primary foam simulation models are local-equilibrium (LE) models (Boeije 2013a, Ma et al. 2013) and population-balance models (Ashoori 2011a, Kam 2007, Falls 1988b, Kovscek 1993). Population-balance models define gas mobility based on the evolution of bubble size (referred to as texture) in the medium. Fine-textured foam with small bubbles and many lamellae is referred to as a strong-foam state (L. Kapetas 2013). On the other hand, LE models assume equal foam generation and destruction rates everywhere in the medium at all times (Ashoori 2011a). Foam is assumed to obtain a strong-foam state instantaneously, which means that gas mobility reduction is also reached instantly. Studies show that these simplifications are generally congruent with each other, since population-balance models generally predict rapid attainment of local equilibrium (Rossen 1996, Kam 2007, Chen 2010).

However, foam can be slow to reach steady-state (LE) on a laboratory scale. Processes that play a significant role on a laboratory scale may be unimportant on the field scale. Therefore, it is of importance to understand and fit the dynamics of foam core-floods, where LE does not apply, before the model is applied on the field scale (L. Kapetas 2013).

During gas injection in a SAG foam process represented by a model with LE behavior, one would expect to find the maximum pressure gradient value at the time of gas breakthrough in the core. This implies a peak in pressure difference before 1 pore volume of gas is injected (PVI). However, this result is not always the case in laboratory experiments. An example is found in a study by Ma et al. (2013). A comparison between transient experimental data and model simulation results was done. The simulation results were obtained using the STARS foam model (Boeije 2013a), which assumes steady-state (LE) throughout the entire core sample.

Laboratory data often display a peak in pressure difference across the core well after gas breakthrough occurs, as seen in *Figure 1*. This is in conflict with the expectation of peak pressure gradient at time of gas breakthrough before 1 PVI. A significant increase in foam generation can be observed in the core between 1 and 1.5 pore volumes of gas injection. These results imply that local equilibrium is not achieved within the first pore volume injected. Similarly, a delay in reaching steady-state in the core sample was observed in previous studies for a SAG core-flood (Friedmann 1991) and for co-injection of gas and surfactant solution (Persoff 1991).

A mismatch between the experimental laboratory data and calculated model data for short times is shown in *Figure 1.* This suggests that the foam parameters used in the model are not correct for the dynamic period in the beginning, where LE does not yet apply. The aim of this paper is to research what effects will arise in late time-behavior as consequence of the mismatch in short time behavior. A possible finding would be that the misfit for short times influences or distorts the late time behavior of the foam model. This hypothesis will be tested with the use of a one-dimensional foam simulation program. An explanation for the possible distortion could originate from the predicted water saturation values in the model. If the foam model is not correct in calculating the water saturations in the dynamic early times where it does not match the lab data and underestimates the amount of water pushed out of the rock sample, this could have an effect on the calculations of the water saturation at later times, where local equilibrium would apply.

The equations and parameters used in the simulation model are described in the method in section 2. After this the results and findings will be discussed and compared in section 3. Finally a conclusion is formulated based on the obtained results in section 4. Nomenclature is listed in section 5 and references can be found in section 6. Appendices are located in section 7 and the used Matlab code in section 8.

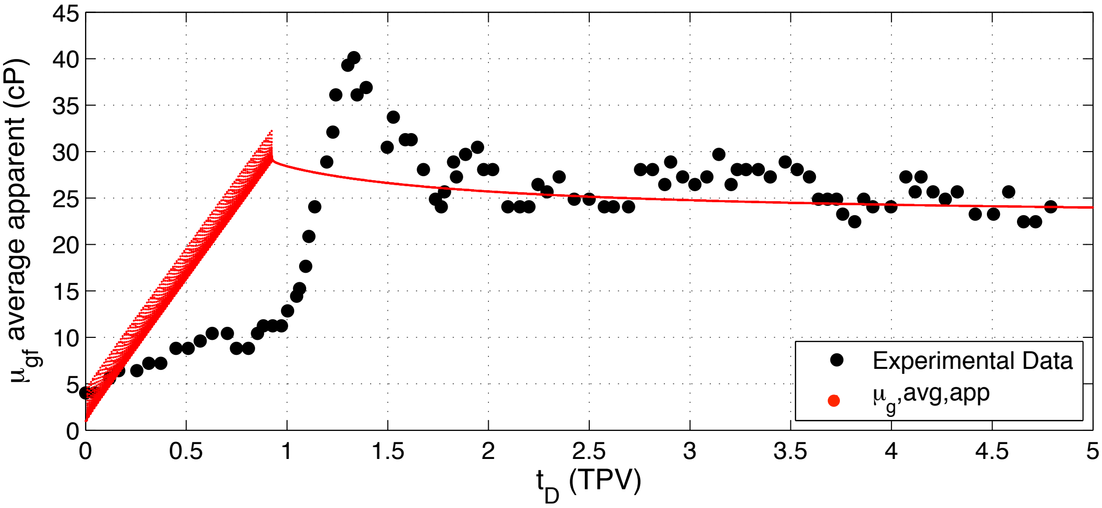

**Figure 1: Average apparent viscosity (Eq. 13) versus pore volume injected in a SAG core-flood by Ma et al. (2013) compared to values calculated by an LE simulation model fit to the late-time behavior.**

# 2. METHOD

## 2.1 EQUATIONS IN LE MODEL

The constructed model is based upon the transport equation in fluid dynamics. In case of one dimensional, two immiscible and incompressible phases the displacement in rectilinear flow through a porous medium is given by the Rapoport-Leas equation (Ashoori 2011a):

$$\varphi \frac{\partial S_w}{\partial t} + u_t \frac{\partial f_w}{\partial x} + \frac{\partial}{\partial x}\left(\left(\frac{\lambda_w \lambda_g}{\lambda_w + \lambda_g} \frac{dP_c}{dS_w}\right)\frac{\partial S_w}{\partial x}\right) = 0 \tag{1}$$

where $S_w$ is water saturation, $f_w$ is water fractional flow, $\varphi$ is porosity, $u_t$ is total superficial velocity, $x$ represents position, $t$ represents time, $P_c$ is capillary pressure and $\lambda_w$ and $\lambda_g$ are the mobilities of water and gas. Our model assumes that the capillary pressure gradient is negligible. This means Equation 1 can be simplified to the Buckley-Leverett equation:

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

where $u_w$ is water superficial velocity. For simplicity, it is assumed that surfactant is present in the entire aqueous phase in full-strength concentration. This can be achieved by a large pre-flush of surfactant. Therefore, no separate material balance is needed on the surfactant solution (Ashoori 2011a). Since we assume the phases are incompressible (Ma et al. 2013):

$$S_w + S_g = 1 \tag{3}$$

and

$$u_w + u_g = u_t \tag{4}$$

where $S_g$ is gas saturation, $u_g$ is gas superficial velocity and $u_t$ is total superficial velocity, which is constant. The model assumes that foam has no effect on water relative permeability and viscosity (Bernard 1965, Vassenden 2000, Huh 1989). Corey-type equations are used for relative permeability's in the absence of foam:

$$k_{rw}(Sw) = k_{rw}^0 \left(\frac{S_w - S_{wc}}{1 - S_{wc} - S_{gr}}\right)^{n_w} \tag{5}$$

$$k_{rg}(Sw) = k_{rg}^0 \left(\frac{1 - S_w - S_{gr}}{1 - S_{wc} - S_{gr}}\right)^{n_g} \tag{6}$$

where $k_{rw}$ is water relative permeability, $k_{rw}^0$ is water end-point relative permeability, $k_{rw}$ is gas relative permeability and $k_{rg}^0$ is gas end-point relative permeability in the absence of foam. $S_{wc}$ is connate water saturation, $S_{gr}$ is residual gas saturation and $n_w$ and $n_g$ are the exponents in the relative permeability curves.

$$k_{rw} = 0, \qquad if\ S_w \leq S_{wc} \tag{7}$$

$$k_{rg} = 0, \qquad if\ S_w \geq 1 - S_{gr} \tag{8}$$

To define gas relative permeability in the presence of foam a mobility reduction factor (MRF) is introduced:

$$MRF = \frac{1}{1 + fmmob \times F2} \tag{9}$$

where $fmmob$ is the maximum attainable mobility reduction (Boeije 2013b, Ma et al. 2013, Cheng 2000) and $F2$ is a water-saturation-dependent function shown in Equation 10.

$$F2 = 0.5 + \frac{arctan\big(epdry(S_w - fmdry)\big)}{\pi} \tag{10}$$

where $epdry$ is a parameter that controls the abruptness of foam collapse and $fmdry$ is a parameter controlling the limiting water saturation around which foam collapses (Boeije 2013b).

The MRF is then used to calculate the gas relative permeability in the presence of foam:

$$k_{rg}^{f} = k_{rg} \times MRF \tag{11}$$

If no foam is present $fmmob$ equals zero; thus MRF equals unity and $k_{rg}$ is unaffected by foam. If foam is present $fmmob$ has a large value; thus MRF is a small value and $k_{rg}$ is greatly reduced (Ashoori 2011a, Falls 1988a, Friedmann 1991, Kam 2002, Kovscek 1995a, Falls 1988b).

The foam apparent viscosity is the inverse of the total relative mobility, and can be calculated with the following equation (Ma et al. 2013):

$$\mu_{app}^{f}(S_w, MRF) = \frac{1}{\frac{k_{rw}(S_w)}{\mu_w} + \frac{k_{rg}^{f}(S_w, MRF)}{\mu_g}} \tag{12}$$

where $\mu_w$ is water viscosity and $\mu_g$ is gas viscosity. The foam apparent viscosity represents foam as a single-phase fluid (Boeije 2013b). Equation 12 results in apparent viscosity as a local property, which means at a given location (L. Kapetas 2013). Ma et al. (*Figure 1*) report an apparent viscosity averaged over the core as function of time (Ma et al. 2013), which can be calculated as follows:

$$\overline{\mu}_{app}^{f}(t) = \frac{k\ \Delta p(t)}{u_t\ \ L} = \frac{1}{L}\int_{0}^{L} \mu_{app}^{f}\big(S_w(x,t), MRF(x,t)\big)\, dx \tag{13}$$

## 2.2 CORRECTION FACTOR IN DYNAMIC MODEL

The mobility reduction function given in Equation 9 can be modified to represent non-equilibrium by introducing a correction factor $\mathrm{X}$. This correction factor is inserted in the mobility-reduction function (MRF) as follows:

$$MRF = \frac{1}{1 + fmmob \times F2 \times \mathrm{X}} \tag{14}$$

Addition of the correction factor in the MRF influences the calculation of water and gas saturations in the grid cells. Consequently the gas relative permeabilities and ultimately the average apparent viscosities (pressure gradient) of the foam are affected. The numerical method of obtaining $\mathrm{X}$ is explained in further detail in the results, in section 3.1 and 3.2.

The discretization that is used for the local equilibrium foam simulation model in Matlab using the above-mentioned equations can be found in *Appendix A*. The simulations with a 1-dimensional displacement are represented by fractional-flow theory using the widely adopted STARS foam model, which can be found in *Appendix B*. Parameter values that are used for calculations in the model are found in *Table 1*. These values are taken from a study by Ma et al. (2013). The Matlab code is found in section 8 in the back of this report.

**Table 1: Parameter values used in the model for this work**

| Parameter | Value |
|---|---|
| $\boldsymbol{\varphi}$ | 0.36 |
| $\boldsymbol{u_t}$ (ft/day) | 20 (7.06 × $10^{-5}$ m/s) |
| $\boldsymbol{k_{rw}^0}$ | 0.79 * |
| $\boldsymbol{k_{rg}^0}$ | 1.00 |
| $\boldsymbol{n_w}$ | 1.96 |
| $\boldsymbol{n_g}$ | 2.29 |
| $\boldsymbol{S_{wc}}$ | 0.07 |
| $\boldsymbol{S_{gr}}$ | 0.00 |
| $\boldsymbol{fmmob}$ | 4.72 × $10^4$ |
| $\boldsymbol{epdry}$ | 425 |
| $\boldsymbol{fmdry}$ | 0.101 |
| $\boldsymbol{\mu_w}$ (cP) | 1.00 (0.001 Pa·s) |
| $\boldsymbol{\mu_g}$ (cP) | 0.02 (0.0002 Pa·s) |
| $\boldsymbol{L}$ (ft) | 1 (0.305 m) |
| $\boldsymbol{grid\ cells}$ | 100 |
| $\boldsymbol{dt}$ (s) | 0.1 |

(* A Value of 0.79 for $k_{rw}^0$ in combination with 0 $S_{gr}$ results in $k_{rw}(Sw = 1) = 0.79$, this is equivalent to assuming an apparent viscosity of water of 1.28 cP at residual gas saturation (L. Kapetas 2013).

# 3. RESULTS

This section reports the results of 1-dimensional foam simulations using the dynamic foam model in Matlab. The problem is approached in two different ways. The first approach involves calculating the correction factor $\mathrm{X}$ directly from the laboratory. The second approach is to define $\mathrm{X}$ as a linearly increasing function from 0 until it reaches a value of 1 at some fixed time, which corresponds to local equilibrium. The modeled data will be compared to the lab data points. The experimental lab data points are interpolated and smoothed to remove outliers. This is done to have a suitable data set with corresponding dimensions to the model for calculations and for better comparison. The results and findings of these approaches will be examined and discussed.

## 3.1 MODEL WITH CORRECTION FACTOR BASED ON DATA

The model without any correction did not succeed in delaying the peak in average apparent viscosity. The model shows a peak at 0.92 PVI, at the time where gas breakthrough occurs. However, by introducing a correction factor $\mathrm{X}$ in the Mobility Reduction Function as seen in Equation 14, a better fit with the lab data can be modeled. At first the correction factor is based on the ratio between average apparent viscosities of the model, $\overline{\mu}^{f}_{app,model}(t)$, and the findings of the laboratory experiments of Ma et al. (2013). $\overline{\mu}^{f}_{app,lab}(t)$. This means $\mathrm{X}$ is the result of a combination of Equation 13 and interpolated lab data:

$$\mathrm{X(t)}_{new} = \mathrm{X(t)}_{old}\,\frac{\overline{\mu}^{f}_{app,lab}(t)}{\overline{\mu}^{f}_{app,model}(t)} \qquad (15)$$

For the first iteration $\overline{\mu}^{f}_{app,model}(t)$ is calculated for each grid block based on the saturations in the given grid block using Equations 5-14 to determine $\mathrm{X(t)}_{new}$. $\mathrm{X(t)}_{old}$ is not yet defined and is given a value of 1 at every time step. *Figure 2* displays a comparison between the results of SAG injection from laboratory experiments performed by Ma et al. and the model with and without correction. A delayed peak in foam average apparent viscosity is modeled at 1.34 PVI injected as found in the experiments of Ma et al. The uncorrected LE model using the parameters of Ma et al. does not show a good fit for the period from 0 to 2 PVI. The corrected model was more successful in fitting short time behavior. Both models show a good fit with the lab data after 2 PVI in the experiment where LE applies.

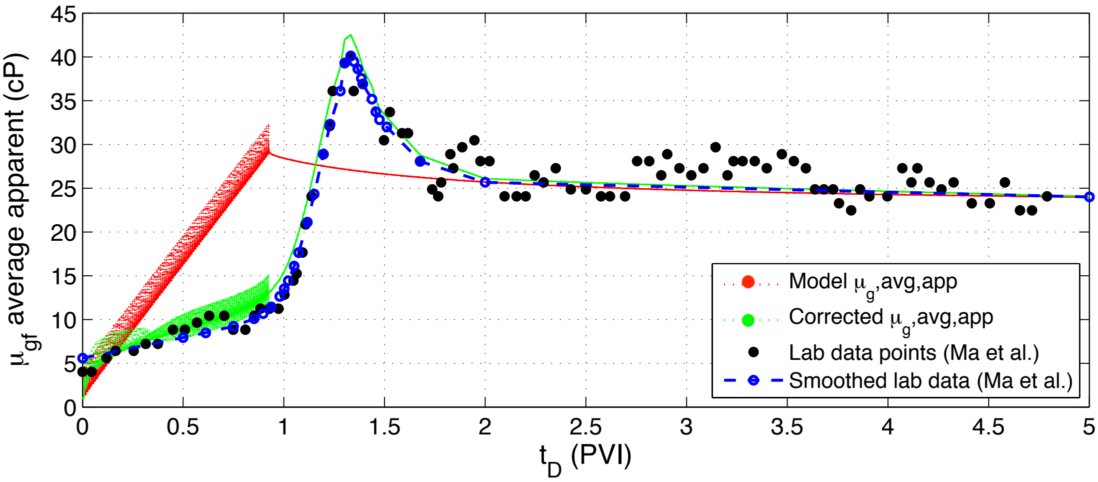

**Figure 2: Average apparent viscosity over pore volume injected from model with and without correction compared to lab data from Ma et al. (2013).**

Fluctuations in average apparent viscosity can be observed in in the model up till 0.92 total pore volume injected. *Figure 3* shows a magnified view of these oscillations. The fluctuations are a common phenomenon in foam models and not the result of numerical instability. The oscillations are a consequence of the foam responding in an abrupt, nonlinear way to changes in water saturation within one grid block as foam advances (Rossen 2013). Foam cannot form and collapse in a shock front of zero width, the minimum width is one grid block. Foam apparent viscosity rises as gas enters a grid-block while its mobility remains low. The following blocks must respond instantly with an equal rise in apparent viscosity. Pressure in the grid block drops as water is drained until gas enters the next grid block. Grid refinement leads to a larger amount of fluctuations, since every grid-block corresponds to 1 peak. However, overall impact on average apparent viscosity is smaller because average apparent viscosity is proportional to grid-block size. A more in-depth review of these numerical artifacts can be found in a study by Rossen et al. (Rossen 2013).

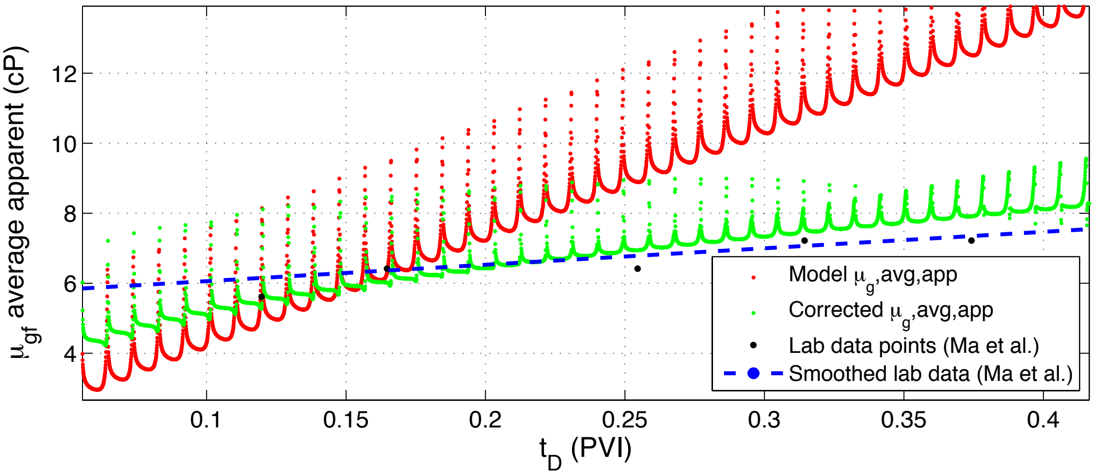

**Figure 3: Oscillations in LE model for foam viscosity.**

After remodeling the corrected foam average apparent viscosities by implementation of Equation 15, another iteration can be done to estimate X. Instead of using the $\overline{\mu}^{f}_{app}(t)$ calculated by the LE model the values of the corrected dynamic model are used. Now X can be re-estimated as:

$$\mathrm{X(t)}_{new} = \mathrm{X(t)}_{old}\,\frac{\overline{\mu}^{f}_{app,lab}(t)}{\overline{\mu}^{f}_{app,corrected\ model}(t)} \qquad (16)$$

This will again influence the calculated gas and water saturations in the grid cells for the time steps and thus the resulting $\overline{\mu}^{f}_{app,model}(t)$. This method to estimate $\mathrm{X(t)}_{new}$ can be repeated; $\mathrm{X(t)}_{new}$ becomes $\mathrm{X(t)}_{old}$ in the next repetition and $\overline{\mu}^{f}_{app,corrected\ model}(t)$ can be recalculated by using the new corrections in the MRF. Up to four corrections were done using the method described above. The model results are displayed in *Figure 4.*

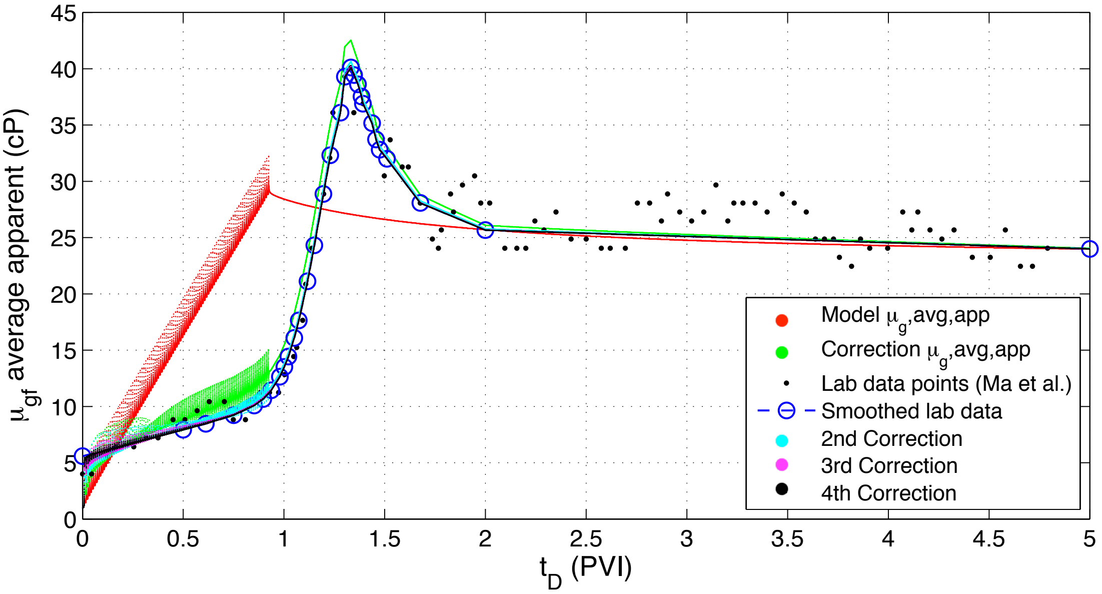

**Figure 4: Average apparent viscosity versus PVI modeled up to four corrections**

Every iteration of the correction factor $\mathrm{X(t)}$ in the MRF returns a closer match to the lab data. After going through the process four times the resulting model calculation is nearly identical to the smoothed experimental data. Especially around the peak pressure gradient between 1.3 and 1.4 PVI the model after fourth correction results in a significantly better fit than the model after first correction. This is displayed in *Figure 5*.

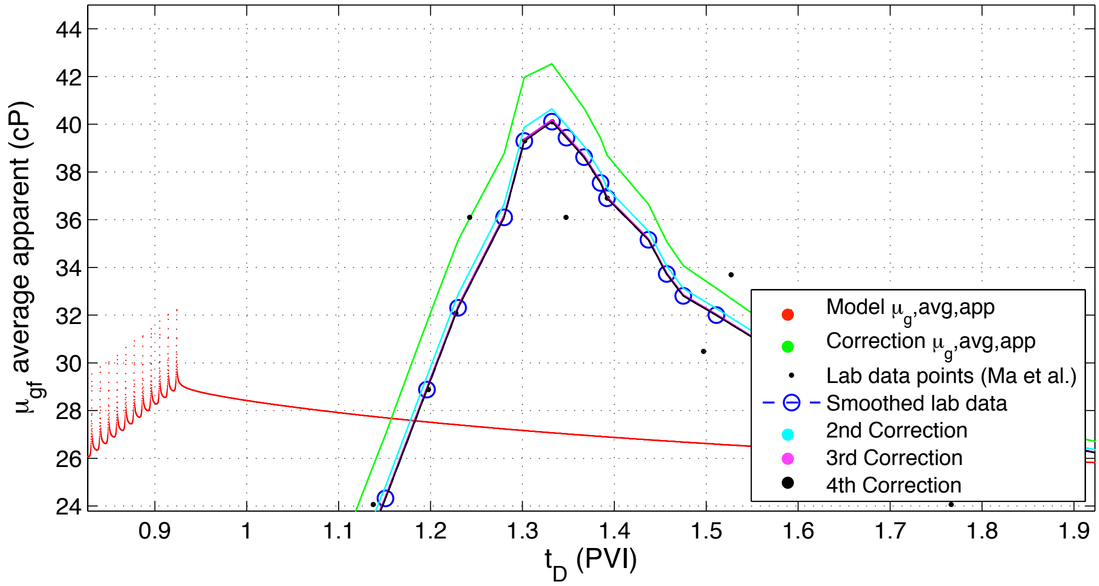

**Figure 5: Comparison in fit at peak pressure gradient of model with lab data up to four corrections**

The correction factors used in the corrected models plotted versus dimensionless time (PVI) are displayed in *Figure 6*.

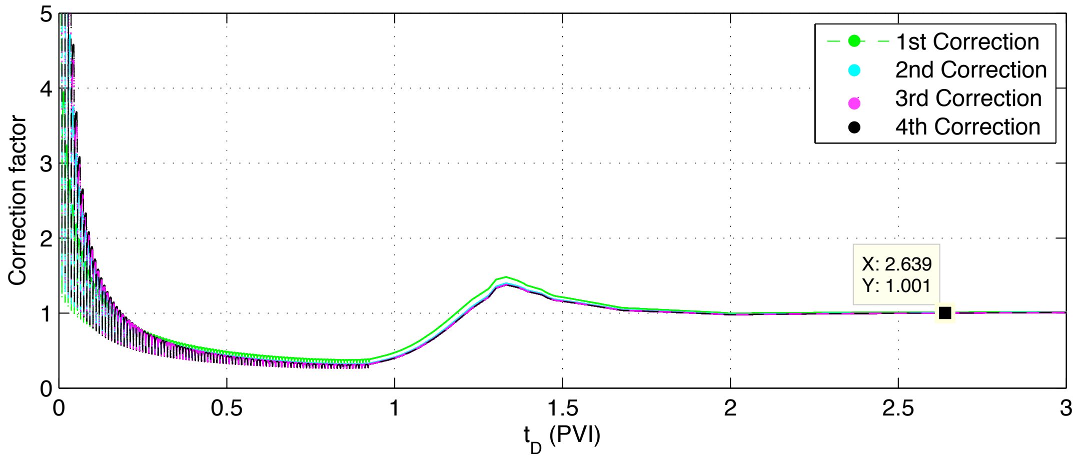

**Figure 6: Correction factors plotted versus dimensionless time (PVI) (on scaled axis).**

The study performed by Ma et al. claims to get an accurate fit by estimating parameters based on steady-state for late times, even when this parameters are not correct for short times, as seen in *Figure 2* or *4*. This claim is tested using our model by validating if our correction factor $\mathrm{X}$ attains a value of 1 at late times. If this is the case, this implies that the late-time data is not distorted by the failure to fit the early-time behavior.

An important outcome of the simulations is the finding that the correction factor $\mathrm{X}$ is equal or very close to a value of 1 at late times. This is seen in *Figure 6,* the data label displays a value of 1.001 for $\mathrm{X}$ at 2.64 PVI. This means that the parameters found by Ma et al. seem to be correct for late time behavior where LE applies. The reason being that a correction factor of equal to or close to 1 means that (almost) no correction is performed in the mobility reduction function of the models to match the experimental lab data at late times. The failure to match behavior at short times did not change the match at late times.

Contour plots of water saturation ($S_w$) and total relative mobility ($\lambda_{rt}$) as function of dimensionless time versus dimensionless position calculated with and without the correction factor support this statement. The contour plots of water saturation calculated with and without $\mathrm{X}$ in *Figure 7* do not show large differences at late times. This supports the claim that LE foam modeling parameters are adequate for modeling late-time behavior. The contour plots of water saturation of the corrected model show a small difference when compared to the uncorrected model. From 0 to 1.5 PVI a curvature in the contours can be seen. This displays the difference in models with and without correction for early times, since the corrected model provides a good fit in the dynamic period while the uncorrected model does not.

The same observations can be done based on the contour plots of total relative mobility shown in *Figure 8.* For late times there is no significant difference in contours. However, from around 0 to 1.5 PVI, deviation can be seen in the uncorrected and corrected total relative mobility's. This again suggests that the LE foam parameters used for the modeling of foam are indeed not correct for predicting short time behavior.

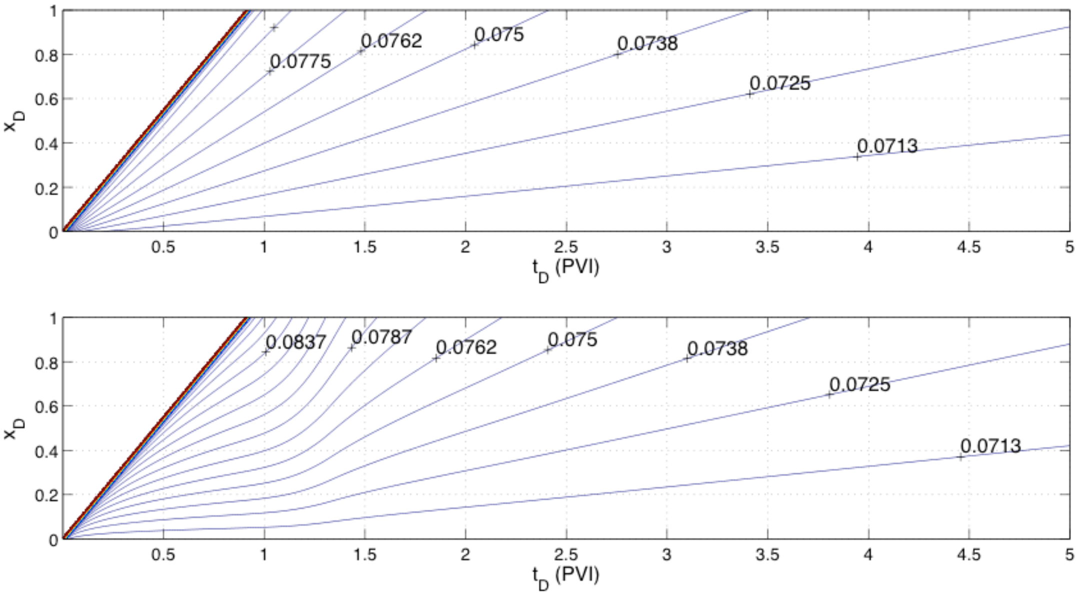

**Figure 7: Comparison of $S_w$ plotted as function of dimensionless time and dimensionless position. The top image is calculated without correction, the bottom image is calculated with correction.**

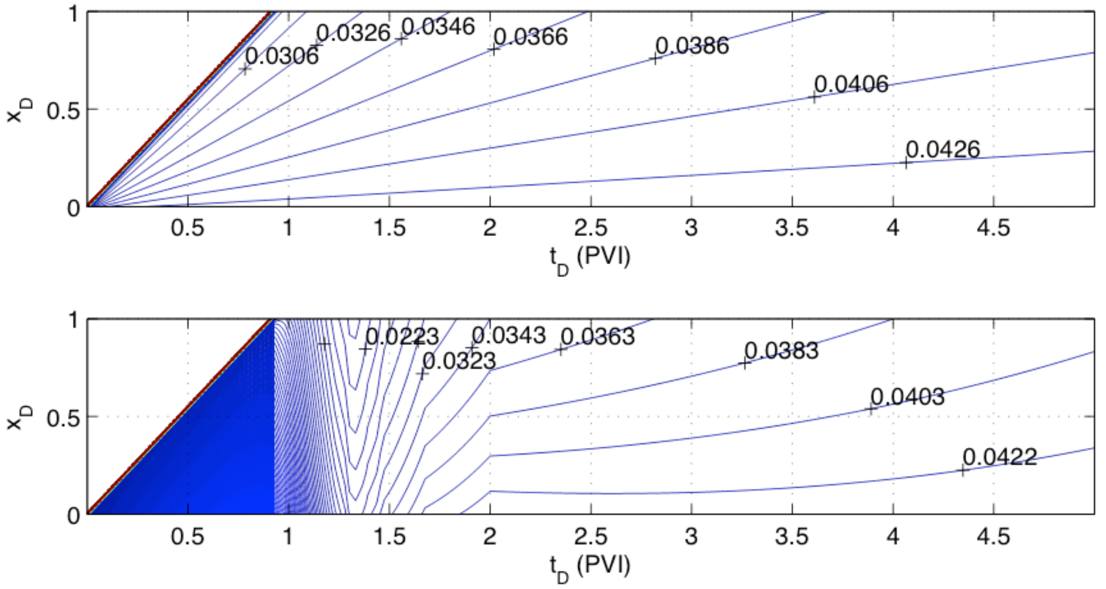

**Figure 8: Comparison of $\lambda_{rt}$ plotted as function of dimensionless time and dimensionless position. The top image is calculated without correction, the bottom image is calculated with correction.**

## 3.2 MODEL WITH CORRECTION FACTOR BASED ON LINEAR FUNCTION

The second approach is using a linear function for the correction factor in the mobility reduction factor as hypothetical experiment. This approach arises from the simplification of assuming that foam linearly strengthens with time. Instead of calculating the correction at every data point, $X$ is no longer based on the LE model or the lab data of Ma et al. (2013) here, as opposed to the previous approach discussed in section 3.1. The model is calculated with two different correction functions individually, to compare the results with each other, the LE foam model and to test their fit with the lab data. The following linear functions are implemented in the mobility reduction function.

Function 1: $X$ rises linearly from a value of 0 to 1 over a period of 0 to 1.34 PVI, when foam appeared to reach its maximum strength in the experiments of Ma et al. After 1.34 PVI the correction factor $X$ stays at a constant value of 1.

Function 2: $X$ rises linearly from a value of 0 to 1.3 over a period of 0 to 1.34 PVI. After 1.34 PVI the correction factor $X$ drops linearly back to a value of 1 over the period from 1.34 to 5 PVI.

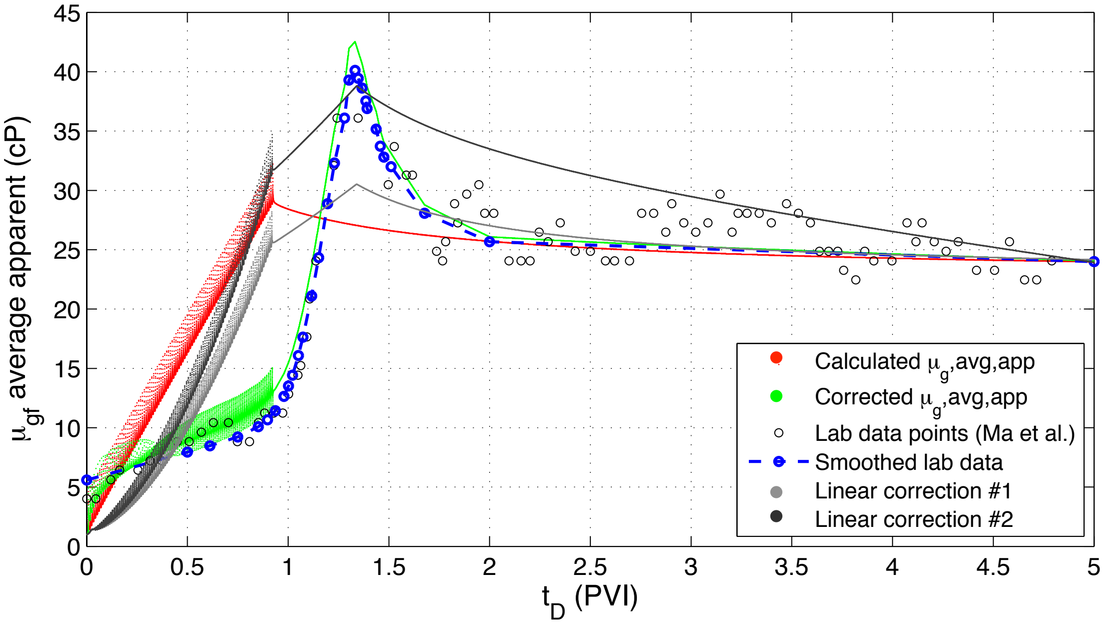

**Figure 9: Average apparent viscosity versus Pore Volume Injected from model with and without linear corrections compared to lab data from Ma et al. (2013).**

The results of the two functions compared to the data obtained in laboratory SAG experiments are shown in *Figure 9*. Both functions do not show a good fit to the lab data in the dynamic period, yet both models with the linear correction functions do a better job at simulating the lab data when compared to the uncorrected model. However, the fit of the linear corrected model is significantly worse than the fit obtained by the model when using the definition for $X(t)$ based on the data as described in section 3.1. This is as expected, since $X(t)$ was fit to the data in section 3.1.

By implementing function 1 into the MRF a delayed peak at the right time of total Pore Volume Injected is calculated; the uncorrected model could not achieve this. However, the obtained peak gives a too low value of average apparent viscosity when compared to the lab data. In function 2 $X(t)$ increases to a larger number than function 1 to compensate for the underestimation of the correction factor which results in a better match in peak pressure gradient at 1.34 PVI. The model with function 2 returns a better result by modeling the peak at the right time of PVI with a better matching value for average apparent viscosity as consequence of the larger correction factor in this region. However, a good match with the lab data is not achieved in the period from 2 PVI and on where local equilibrium would apply.

The model with linear correction function 1 supplies a good fit in the late time behavior when compared to the calculated model and lab data. The peak in pressure gradient happens at the same time as in the lab data, as opposed to the calculated model where the peak was at 0.92 PV where gas breakthrough took place in the laboratory SAG experiment. The second linear correction does not show a good fit at late times were LE would apply when compared to the calculated model and lab data. This mismatch is a consequence of the correction factor having a too large value to correctly model behavior here. As discussed in section 3.1, the correction factor at late times should be equal or very close to a value of 1 if the procedure of using late-time data to fit the LE model is adequate.

The same conclusions can be drawn from the contour plots of water saturation and total relative mobilities, shown respectively in *Figure 10* and *Figure 11*. Not much difference can be noticed when comparing the water saturation contours of all models, especially for late-time late behavior. This is not strange because all the models in this work match the lab data well for late times, with the exception of the model where linear correction function 2 was used in section 3.2.

A comparable shape or curvature can be observed in the middle and bottom contour plots of *Figure 11*, this can also be seen in the bottom contour plot of the total relative mobilities for the first approach in section 3.1 in *Figure 7.* This curvature is based around a PVI of 1.34 at the time of peak pressure gradient. The fact that function 2 does not achieve a good fit for late times can also be noticed by the significant differences in the contours of total relative mobility when compared with the contours of other models. As mentioned above, the mismatch for late times is due the correction factor having a value larger than 1 in this period, since it drops linearly from 1.3 to 1 over from 1.35 PVI to 5 PVI. To achieve a better fit, an option would be to let function 2 descend faster to a value of 1 at 1.34 PVI and on. The drop cannot be instant from 1.3 to 1, as this would result in a sudden gap between data points without connection.

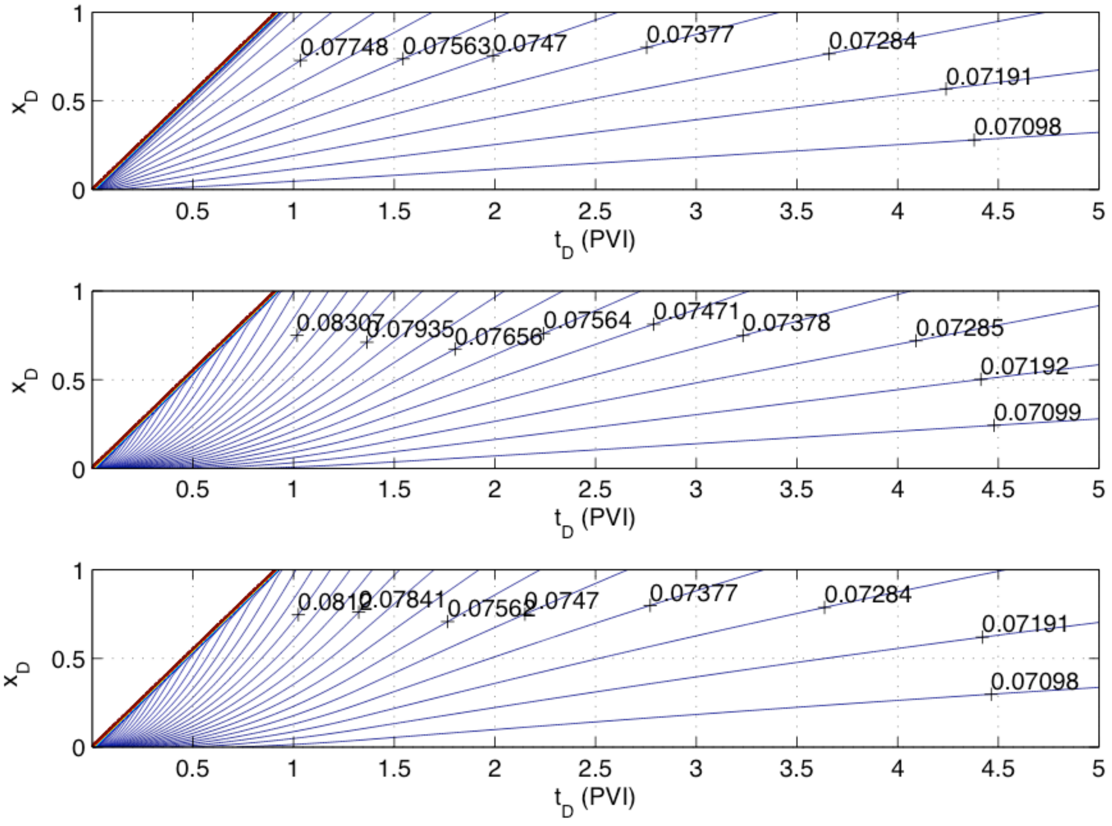

**Figure 10: Comparison of $S_w$ plotted as function of dimensionless time and dimensionless position. The top image is the LE model without correction, the middle plot with linear correction function 1 and the bottom plot with linear correction function 2.**

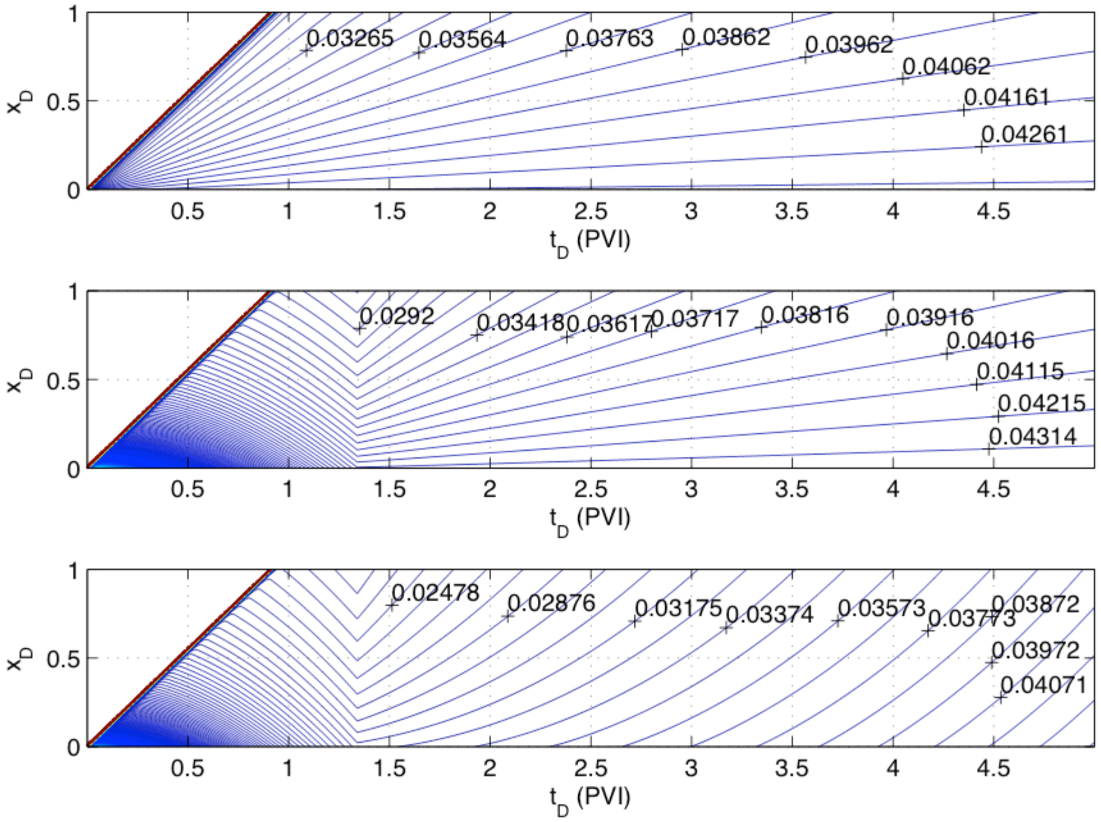

**Figure 11: Comparison of $\lambda_{rt}$ plotted as function of dimensionless time and dimensionless position. The top image is the LE model without correction, the middle plot with linear correction function 1 and the bottom plot with linear correction function 2.**

## 4. CONCLUSION

The hypothesis was that disregarding a good fit in the dynamic period would influence late-time behavior of the model. Our aim was to investigate if any distortion would occur in late-time behavior of the model where local equilibrium would apply, as a result of using the foam parameters obtained by Ma et al. (2013). Since these parameters do not fit the data obtained in the laboratory experiments at short times where dynamics are of importance.

It is assumed that the foam parameters used in the model are correct for fitting late times where LE applies. This assumption is supported by the results obtained; both the uncorrected and the corrected models show a good fit with the smoothed lab data from Ma et al. (2013) where the correction factor takes on a value of around 1. This means that no correction is needed on the model to predict the foam behavior that was observed in the laboratory SAG experiment.

The data suggests that the used parameters in the LE foam model simulations are correct in predicting steady-state late-time behavior. Several results support this, among which the finding that the correction factor in section 3.1 is equal to or nearly 1 later on in the experiment. The contour plots of water saturation and total relative mobility also correspond to the claim of correct foam parameters where LE applies. No significant anomalies can be spotted when comparing the contour plots of the corrected and uncorrected model. The exception to these findings is the model that uses linear function 2, which proved to be incapable of modeling late time behavior. This incapability is most evident when comparing the contours of total relative mobility to the contours produced by the other models.

The model results do not indicate clear evidence that fitting long time behavior is distorted by disregarding a good fit in short time behavior. No significant influence on the model could be detected as result of ignoring the dynamics in the beginning. This means that the modeling parameters obtained by Ma et al. (2013) could be correct for modeling LE behavior even though a good fit for short times is disregarded.

## 5. NOMENCLATURE

**Table 2: Nomenclature**

| Symbol | Description |
|---|---|
| $epdry$ | Parameter controlling abruptness of foam collapse |
| $fmmob$ | Reference mobility reduction factor |
| $fmdry$ | Critical water saturation at which foam collapses |
| $MRF$ | Mobility Reduction Factor |
| $\mathrm{X}$ | Correction factor |
| $k_{rg}$ | Relative permeability of gaseous phase |
| $k_{rg}^{0}$ | End-point relative permeability of gaseous phase |
| $k_{rg}^{f}$ | Gas relative permeability in presence of foam |
| $k_{rw}$ | Relative permeability of aqueous phase |
| $k_{rw}^{0}$ | End-point relative permeability of aqueous phase |
| $n_g$ | Exponent in $k_{rg}$ curve |
| $n_w$ | Exponent in $k_{rw}$ curve |
| $S_g$ | Gas saturation |
| $S_{gr}$ | Residual gas saturation |
| $S_w$ | Water saturation |
| $S_{wc}$ | Connate water saturation |
| $t$ | Time [s] |
| $u_g$ | Gas superficial velocity [ft/day] |
| $u_t$ | Total superficial velocity [ft/day] |
| $u_w$ | Water superficial velocity [ft/day] |
| $x$ | Displacement coordinate [ft] |
| $\varphi$ | Porosity |
| $L$ | Length of rock core sample [ft] |
| $\mu_g$ | Viscosity of gas [cP] |
| $\mu_w$ | viscosity of water [cP] |
| $x$ | Displacement coordinate |
| $\Delta p$ | Pressure difference across core [Pa] |
| $\mu_{app}^{f}$ | Apparent foam viscosity [cP] |
| $\overline{\mu}_{app}^{f}$ | Average apparent foam viscosity (averaged over core) [cP] |

**Superscripts**

| | |
|---|---|
| $f$ | Presence of foam |
| $o$ | Absence of foam |

## 6. REFERENCES

Ashoori, E., Marchesin, D. and Rossen, W. 2012. Multiple Foam States and Long-Distance Foam Propagation in Porous Media (in *SPE Journal* **17** (4): 1231-1245.

Ashoori, E., Marchesin, D., Rossen, W. R. 2011a. Roles of transient and local equilibrium foam behavior in porous media: Traveling wave (in *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **377** (1-3): 228-242.

Bernard, G. G., Holm, L. W., and Jacobs, W. L. 1965. Effect of Foam on Trapped Gas Saturation and on Permeability of Porous Media to Water (in *SPE Journal* **5**: 295-300.

Boeije, C., Rossen W.R. 2013a. Fitting Foam Simulation Model Parameters for SAG Foam Applications (in *SPE Enhanced Oil Recovery Conference, Kuala Lumpur, Malaysia*.

Boeije, C.S., and Rossen, W.R. 2013b. Fitting foam simulation model parameters to data (in *IOR 2013* (17th European Symposium for IOR, St Petersburg, Russia).

Chen, Q., Gerritsen, M. and Kovscek, A.R. 2010. Modeling Foam Displacement With the Local-Equilibrium Approximation: Theory and Experimental Verification (in *SPE Journal* **15** (1): 171-183.

Cheng, L., Reme, A.B., Shan, D., Coombe, D.A. and Rossen, W.R. 2000. Simulating Foam Processes at High and Low Foam Qualities (in *SPE/DOE Improved Oil Recovery Symposium*.

D. Shan, W.R. Rossen. 2004. Optimal Injection Stragegies for Foam IOR (in *SPE Journal* **9**: 132-150.

Falls, A.H. et al. 1988a. Development of a Mechanistic Foam Simulator: The Population Balance and Generation by Snap-Off (in *SPE Reservoir Engineering* **3** (3): 884-892.

Falls, A.H., G.J. Hirasaki, T.W. Patzek et al. 1988b. Development of a Mechanistic Foam Simulator: The Population Balance and Generation by Snap-Off (in *SPE Reservoir Engineering* **3** (3): 884-892.

Friedmann, F., Chen, W.H. & Gauglitz, P.A. 1991. Experimental and Simulation Study of High-Temperature Foam Displacement in Porous Media (in *SPE Reservoir Engineering* **6** (1): 37-45.

G.G. Bernard, L. W. Holm. 1964. Effect of foam on permeability of porous media to gas (in *Society of Petroleum Engineers* **4** (3): 267-274.

Huh, D. G. and Handy, L. L. 1989. Comparison of Steady and Unsteady-State Flow of Gas and Foaming Solution in Porous Media (in *SPE Reservoir Engineering* **4**: 77-84.

J.M. Alvarez, H.J. Rivas, W.R. Rossen. 2001. A Unified Model for Steady-State Foam Behavior at High and Low Foam Qualities (in *SPE Journal* **6**: 325-333.

Kam, S.I. and Rossen, W.R. 2002. A Model for Foam Generation in Homogeneous Media (in *SPE Annual Technical Conference and Exhibition* (Copyright 2002, Society of Petroleum Engineers Inc., San Antonio, Texas).

Kam, Seung I., Quoc P. Nguyen, Qichong Li et al. 2007. Dynamic Simulations With an Improved Model for Foam Generation (in *SPE Journal* **12** (1): 35-48.

Kovscek, A.R., Patzek, T.W. and Radke, C.J. 1995a. A mechanistic population balance model for transient and steady-state foam flow in Boise sandstone (in *Chemical Engineering Science* **50** (23): 3783-3799.

Kovscek, A.R., T.W. Patzek, C.J. Radke. 1993. Simulation of Foam Transport in Porous Media (in *SPE Annual Technical Conference and Exhibition, Houston, Texas*.

L. Kapetas, A. van El, W.R. Rossen. 2013. Representing Slow Foam Dynamics in Laboratory Corefloods for Enhanced Oil Recovery (in *SPE 169059*.

Lake, L. W. 1989. Enhanced Oil Recovery (in *Prenctice Hall, Englewood Cliffs, New Jersey, USA*.

Ma, Kun, Jose L. Lopez-Salinas, Maura C. Puerto et al. 2013. Estimation of Parameters for the Simulation of Foam Flow through Porous Media. Part 1: The Dry-Out Effect (in *Energy & Fuels* **27** (5): 2363-2375.

Persoff, P., Radke, C.J., Pruess, K., Benson, S.M. and Witherspoon, P.A. 1991. A Laboratory Investigation of Foam Flow in Sandstone at Elevated Pressure (in *SPE Reservoir Engineering* **6**: 185-192.

Rossen, W.R. 1996. Foams in Enhanced Oil Recovery (in *Foams: Theory, Measurements and Applications, New York*.

Rossen, W.R. 2013. Numerical Challenges in Foam Simulation: A Review (in *Society of Petroleum Engineers* **16624432**.

Vassenden, F. and Holt, T. 2000. Experimental Foundation for Relative Permeability Modeling of Foam (in *SPE Reserv. Eval. Eng.* **3**: 179-185.

## 7. APPENDICES

### APPENDIX A – LOCAL EQUILIBRIUM MODEL

The porous medium is divided into $N$ grid cells. To solve equations (1) to (5) in 1D, an explicit upwind finite-difference scheme is used. By setting initial and boundary conditions a simulation model can be built for the porous medium over time.

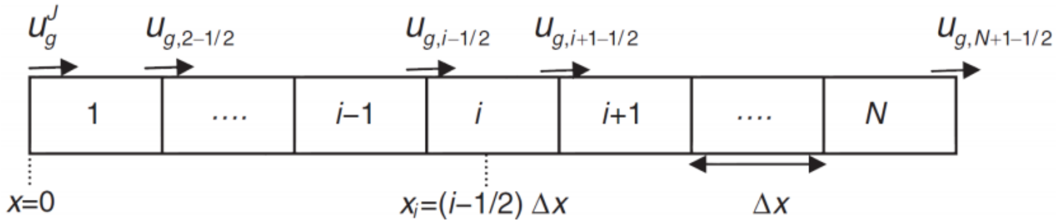

**Figure A1: Grid used in numerical scheme (from Ashoori et al. (Ashoori 2012)).**

$$\varphi\frac{S_{g,i}^{t+\Delta t}-S_{g,i}^{t}}{\Delta \mathrm{t}}+\frac{u_{g,i+\frac{1}{2}}^{t}-u_{g,i-\frac{1}{2}}^{t}}{\Delta \mathrm{x}}=0 \qquad \text{(A1)}$$

where $i$ represents the index of the $i^{th}$ cell. Equation (A1) can be rewritten to solve for gas saturation at the next timestep:

$$S_{g,i}^{t+\Delta t}=S_{g,i}^{t}+\frac{\Delta \mathrm{t}}{\varphi\Delta \mathrm{x}}\left(u_{g,i+\frac{1}{2}}^{t}-u_{g,i-\frac{1}{2}}^{t}\right) \qquad \text{(A2)}$$

In the model it is assumed that the first grid cell ($i = 1$) only gas is injected. This means that the gas superficial velocity is equal to the total superficial velocity at $x = 0$. We also assume that initially the entire porous medium is filled with surfactant solution and no gas is present at time $t = 0$. This gives the following boundary conditions for the model:

$$for\ x=0,\qquad u_g=u_t$$

$$for\ t=0,\qquad S_g=0$$

A constraint of this numerical scheme is reliance on the time step, which has to be small in order accurately model results. As a result many computations have to be executed to obtain model data. We used a fixed time step of 0.1. Which allowed a vector of $\mathrm{X(t)}$ values to be stored and used in the next iteration at exactly the same times.

# APPENDIX B – STARS FOAM MODEL

The foam model in STARS controls the gas mobility reduction with a function FM, which is built from various other functions $F_1$ to $F_6$. These functions are all constrained to a value less than or equal to 1, so that they can only reduce the gas mobility-reduction factor, which means an increase in gas mobility (Boeije 2013a):

$$FM = \frac{1}{1 + fmmob \times F1 \times F2 \times F3 \times F4 \times F5 \times F6} \tag{B1}$$

where $fmmob$ is the reference gas mobility factor for wet foams. This parameter represents the maximum attainable mobility reduction. The effects of surfactant concentration is modeled by $F_1$, the effects of water saturation by $F_2$, oil saturation by $F_3$, gas velocity by $F_4$, oil saturation by $F_5$ and critical capillary number by $F_6$. Only the function water saturation function is used in this work, which is given by:

$$F2 = 0.5 + \frac{arctan\big(epdry(S_w - fmdry)\big)}{\pi} \tag{B2}$$

where $fmdry$ is equal to the water saturation at which foam collapses $S_w^*$, if the transition between regimes is abrupt. The magnitude of $epdry$ controls the abruptness of the foam collapse. Small values of $epdry$ give a gradual transition between the two regimes, while larger values yield a sharper, but still continuous transition (Boeije 2013a).

# 8. MATLAB CODE

## MODEL WITHOUT FOAM

```
clear all; close all; clc

%%_________________________________________________________________________
%   Two-phase incompressible model to run simulations without foam
%   The model applies Buckley-Leverett flow. Gas
%   relative viscosities (or pressure drop) are calculated in the absence
%   of foam when injected into a fully water-saturated core sample.
%%_________________________________________________________________________

% Parameters _____________________________________________________________

    % Core sample parameters
    phi=0.36;                                  % Porosity
    L=1;                                       % size of the core used in the lab [ft]
    u_t=20;                                    % total velocity [ft/d]

    %   Corey exponents and rel.perm coefficients
    %   (from Fitting Foam Simulation Model Paramters for SAG Foam Applications, persoff et al.
(1991); SPE 165282)
    mu_w=1.0;                                   % Water viscosity [cP]
    Swc=0.07;                                   % residual water saturation
    k_rwo=1;                                    % end point water relative permeability
    nw=1.96;                                    % Corey exponent for water relative permeability

    mu_g=0.02;                                  % Gas viscosity [cP]
    k_rgo=1;                                    % end point gas relative permeability
    ng=2.29;                                    % Corey exponent for gas relative permeability
function
    Sgr=0.0;                                    % residual gas saturation

    %   Discretisation parameters
    dt=1/(24*60*60);                           % time step [s]
    tsteps=10000;                              % amount of time steps
    xgrid=100;                                   % amount of grid cells
    dx=L/xgrid;                                % size of grid cells

    %   Build matrices for calculations
    lambda_rt=zeros(tsteps,xgrid);             % total relative mobility
    krw=zeros(tsteps,xgrid);                   % water relative permeability
    krgnf=zeros(tsteps,xgrid);                 % gas relative permeability in the presence of foam
    fw=zeros(tsteps,xgrid);                    % water fractional flow
    Sw=zeros(tsteps,xgrid);                    % water saturation
    Sg=zeros(tsteps,xgrid);                    % gas saturation
    u_g=zeros(tsteps,xgrid);                   % gas superficial velocity [m s^-1]
    mu_gnf=zeros(tsteps,xgrid);                % gas apparent viscosity
    mu_gnfs=zeros(tsteps,1);                   % average gas apparent viscosity over the whole core
    tD=zeros(tsteps,1);                        % dimensionless time (PVI)

%  Initial & Boundary Conditions  ________________________________________

    Sw(:,:)=1;                                  % initially the core is fully water saturated
    Sg(:,:)=0;                                  % initially no gas has been injected in the core
    u_g(:,1)=u_t;                               % injection rate at the first boundary

%  Calculate Sg and Sw  __________________________________________________

    for t=1:tsteps-1;
        for i=1:xgrid;
```

```
         % Calculating rel perms, fractional flow, mobility and gas viscosity
[fw(t,i),u_g(t,i+1),mu_gnf(t,i),krw(t,i),krgnf(t,i),lambda_rt(t,i)]=fracflownf(Sw(t,i),Swc,Sgr,k_rwo,
k_rgo,nw,ng,mu_w,mu_g,u_t);

         % Calculating gas & water saturation at t+1
         Sg(t+1,i)=(Sg(t,i))+(u_g(t,i)-u_g(t,i+1))*dt/(dx*phi);
         Sw(t+1,i)=1-Sg(t+1,i);
        end
    end

%  Calculate avg apparent mu and PVI _____________________________________

 for t=1:tsteps;
     for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gnfs(t,1)=mean(mu_gnf(t,1:xgrid));
     end
 end

    plot(tD(1:tsteps-1,1),mu_gnfs(1:tsteps-1,1),'r.')
    grid on
    title('gas apparent viscosity')
    hold on
    axis('tight');
    xlabel('t_D [TPV]')
    ylabel('\mu_g_n_f [cP]')
    legend('Gas apparent viscosity', 'location','NorthEast')
```

## FRACFLOWNF FUNCTION

```
function [fw,ug,mu_gnf,krw,krgnf,lambda_rt] =
fracflownf(Sw,Swc,Sg_r,k_rwo,k_rgo,nw,ng,mu_water,mu_gas,u_t)


% Calculating krw and krg(relative water permeability) using Brooks-Corey correlation
if Sw>Swc;
       krw=k_rwo*((Sw-Swc)/(1-Swc-Sg_r))^nw;
   else
       krw=0;
end

if Sw<1-Sg_r;
        krgnf=k_rgo*((1-Sw-Sg_r)/(1-Swc-Sg_r))^ng;
     else
        krgnf=0;
end

%Calculating the mobilities and total mobility
    lambda_gnf=krgnf/(mu_gas);
    lambda_w=krw/mu_water;
    lambda_rt=lambda_w+lambda_gnf;

%Calculating fractional flow
    fw=1/(1+(1/lambda_w)*(lambda_gnf));

%calculating water and gas superficial velocity
    uw=fw*u_t;
    ug=u_t-uw;

%calculating gas apparent viscosity [cP]
    mu_gnf=1/((lambda_w)+(lambda_gnf));

end
```

## FRACFLOW FUNCTION

```
function [fw,ug,mu_gf,krw,krgnf,krgf,lambda_w,lambda_gf,lambda_rt] =
fracflowf(Sw,Swc,Sg_r,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_water,mu_gas,u_t)

% Calculate krw and krg (Brooks-Corey correlation)
if Sw>Swc;
       krw=k_rwo*((Sw-Swc)/(1-Swc-Sg_r))^nw;
   else
       krw=0;
end

if Sw<1-Sg_r;
        krgnf=k_rgo*((1-Sw-Sg_r)/(1-Swc-Sg_r))^ng;
     else
        krgnf=0;
end

% Calculate F2 (Water saturation dependent function)
    F2=0.5+(atan(epdry*(Sw-fmdry)))/pi;

% Calculate FM (Mobility reduction factor)
    FM=1/(1+fmmob*F2);

% Calculate gas rel perm with foam
    krgf=krgnf*FM;

% Calculate the mobilities and total mobility
    lambda_gf=krgf/(mu_gas);
    lambda_w=krw/mu_water;
    lambda_rt=lambda_w+lambda_gf;

% Calculate fractional flow
    fw=1/(1+(1/lambda_w)*(lambda_gf));

% Calculate water and gas superficial velocity
    uw=fw*u_t;
    ug=u_t-uw;

% Calculate gas apparent viscosity [cP]
    mu_gf=1/((lambda_w)+(lambda_gf));

end
```

## MODEL WITH MULTIPLE CORRECTIONS

```
clear all; close all; clc

%%________________________________________________________________________
%   Two-phase incompressible model to run simulations on Dyanamic SAG Foam Floods.
%   The model applies Buckley-Leverett flow for foam simulation. Gas
%   relative viscosities (or pressure drop) are calculated in the presence of foam
%   when injected into a fully water-saturated core sample.
%   The determined pressure drop and lab data is used to calculate a
%   correction factor to fit the late peak in the lab data of Ma et al.
%%________________________________________________________________________
```

**Parameters**

```
    %   Core sample parameters
    phi=0.36;                                   % Porosity
    L=1;                                        % size of the core used in the lab [ft]
    u_t=20;                                     % total velocity [ft/d]

    %   Corey exponents and rel.perm coefficients
    %   (from Fitting Foam Simulation Model Paramters for SAG Foam Applications, Persoff et al.
(1991); SPE 165282)
    mu_w=1.0;                                   % Water viscosity [cP]
    Swc=0.07;                                   % residual water saturation
    k_rwo=1;                                    % end point water relative permeability
    nw=1.96;                                    % Corey exponent for water relative permeability

    mu_g=0.02;                                  % Gas viscosity [cP]
    k_rgo=1;                                    % end point gas relative permeability
    ng=2.29;                                    % Corey exponent for gas relative permeability
function
    Sgr=0.0;                                    % residual gas saturation

    %   Discretisation parameters
    dt=1/(10*(24*60*60));                       % time step [s]
    tsteps=77760;                               % amount of time steps to reach certain PV, has to
correspond with PV below (3.5 PV=54432, 2 PV=31104)
    xgrid=100;                                   % amount of grid cells
    dx=L/xgrid;                                 % size of grid cells

    %   Foam parameters
    fmmob=4.72e4;                               % reference gas mobility reduction factor for wet
foams
    fmdry=0.101;                                % water saturation in vicinity of which foam
collapses
    epdry=425;                                  % abruptness of foam dry-out effect

    %   Build matrices for calculations of Sw and Sg
    lambda_rt=zeros(tsteps-1,xgrid);              % total relative mobility
    krw=zeros(tsteps,xgrid);                    % water relative permeability
    krgnf=zeros(tsteps,xgrid);                  % gas relative permeability without foam
    krgf=zeros(tsteps,xgrid);                   % gas relative permeability in the presence of foam
    fw=zeros(tsteps,xgrid);                     % water fractional flow
    Sw=zeros(tsteps,xgrid);                     % water saturation
    Sg=zeros(tsteps,xgrid);                     % gas saturation
    u_g=zeros(tsteps,xgrid);                    % gas superficial velocity [m s^-1]
    mu_gf=zeros(tsteps,xgrid);                  % gas apparent viscosity with foam
    mu_gfs=zeros(tsteps,1);                     % average gas apparent viscosity over the whole core
    tD=zeros(tsteps,1);                         % dimensionless time (PVI)

    %   Build matrices for calculations of Swx and Sgx (with correction)
    lambda_rtx=zeros(tsteps-1,xgrid);             % total relative mobility with correction
    krwx=zeros(tsteps,xgrid);                   % water relative permeability with correction
    krgfx=zeros(tsteps,xgrid);                  % gas relative permeability in the presence of foam
with correction
    fwx=zeros(tsteps,xgrid);                    % water fractional flow with correction
    Swx=zeros(tsteps,xgrid);                    % water saturation with correction
    Sgx=zeros(tsteps,xgrid);                    % gas saturation with correction
    u_gx=zeros(tsteps,xgrid);                   % gas superficial velocity with correction [m s^-1]
    mu_gfx=zeros(tsteps,xgrid);                 % gas apparent viscosity with foam with correction
    mu_gfsx=zeros(tsteps,1);                    % average gas apparent viscosity  with correction
over the whole core

    %   2nd correction
    lambda_rtx2=zeros(tsteps-1,xgrid);             % total relative mobility with correction
    krwx2=zeros(tsteps,xgrid);                   % water relative permeability with correction
    krgfx2=zeros(tsteps,xgrid);                  % gas relative permeability in the presence of foam
with correction
    fwx2=zeros(tsteps,xgrid);                    % water fractional flow with correction
    Swx2=zeros(tsteps,xgrid);                    % water saturation with correction
    Sgx2=zeros(tsteps,xgrid);                    % gas saturation with correction
    u_gx2=zeros(tsteps,xgrid);                   % gas superficial velocity with correction [m s^-1]
    mu_gfx2=zeros(tsteps,xgrid);                 % gas apparent viscosity with foam with correction
    mu_gfsx2=zeros(tsteps,1);                    % average gas apparent viscosity  with correction
over the whole core

    %   3th correction
    lambda_rtx3=zeros(tsteps-1,xgrid);             % total relative mobility with correction
    mu_gfx3=zeros(tsteps,xgrid);                 % gas apparent viscosity with foam with correction
    mu_gfsx3=zeros(tsteps,1);                    % average gas apparent viscosity  with correction
over the whole core

    %   4th correction
    lambda_rtx4=zeros(tsteps-1,xgrid);             % total relative mobility with correction
    mu_gfx4=zeros(tsteps,xgrid);                 % gas apparent viscosity with foam with correction
    mu_gfsx4=zeros(tsteps,1);                    % average gas apparent viscosity  with correction
over the whole core
```

### Initial & Boundary Conditions

```
    Sw(:,:)=1;                                  % initially the core is fully water saturated
    Sg(:,:)=0;                                  % initially no gas has been injected in the core
    u_g(:,1)=u_t;                               % injection rate at the first boundary
    lambda_rt(:,:)=1;                           % fill total relative mobility matrix with initial
value's

    %   for calculations with correction
    Swx(:,:)=1;                                 % initially the core is fully water saturated
    Sgx(:,:)=0;                                 % initially no gas has been injected in the core
    u_gx(:,1)=u_t;                              % injection rate at the first boundary
    lambda_rtx(:,:)=1;                          % fill total relative mobility matrix with initial
value's

    %   for calculations with 2nd correction
    Swx2(:,:)=1;                                 % initially the core is fully water saturated
    Sgx2(:,:)=0;                                 % initially no gas has been injected in the core
    lambda_rtx2(:,:)=1;                          % fill total relative mobility matrix with initial
value's

    %   for calculations with 3rd correction
    Swx3(:,:)=1;                                 % initially the core is fully water saturated
    Sgx3(:,:)=0;                                 % initially no gas has been injected in the core
    lambda_rtx2(:,:)=1;                          % fill total relative mobility matrix with initial
value's

    %   for calculations with 4th correction
    Swx4(:,:)=1;                                 % initially the core is fully water saturated
    Sgx4(:,:)=0;                                 % initially no gas has been injected in the core
    lambda_rtx4(:,:)=1;                          % fill total relative mobility matrix with initial
value's
```

### Calculate Sg and Sw (without correction)

```
    for t=1:tsteps-1;
        for i=1:xgrid;
         %  Calculate rel. perms, fractional flow, mobility and gas viscosity

[fw(t,i),u_g(t,i+1),mu_gf(t,i),krw(t,i),krgnf(t,i),krgf(t,i),lambda_w(t,i),lambda_gf(t,i),lambda_rt(t
,i)]=fracflowf(Sw(t,i),Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
         %  Calculate gas & water saturation at t+1
         Sg(t+1,i)=(Sg(t,i))+(u_g(t,i)-u_g(t,i+1))*dt/(dx*phi);
         Sw(t+1,i)=1-Sg(t+1,i);
        end
    end
```

### Calculate Mu_g,avg,app (without correction)

```
 for t=1:tsteps;
     for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfs(t,1)=mean(mu_gf(t,1:xgrid));
     end
 end
```

### Create contour plots without corrections

```
    figure(3)
    subplot(2,1,1)
    contSw=contour(tD(1:tsteps), linspace(0,L,xgrid), Sw', 750);
    clabel(contSw, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
    %title('Contour plot of uncorrected water saturations')
    grid on
    hold on

    figure(4)
    subplot(3,1,1)
    contRt=contour(tD(1:tsteps-1), linspace(0,L,xgrid), lambda_rt', 500);
    clabel(contRt, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
    %title('Contour plot of uncorrected total relative mobilities')
    grid on
    hold on
```

### Interpolate lab data (Ma et al.) and Calculate correction factor cx

```
Simulation parameters
    PV=5;                                       % Untill which PV to run the interpolation, has to
correspond to tsteps above (Note: Will change automatically for last entry in xsmooth array)
    stepsize=(PV/(tsteps-1));

    %   Smoothed data of Ma et al. (used for calculation)
    xsmooth=[0, 0.5, 0.6116, 0.7503, 0.8521, 0.8977, 0.9368, 0.9791, 1.002, 1.022, 1.051, 1.074,
1.117, 1.151, 1.196, 1.23, 1.28, 1.302, 1.332, 1.348, 1.367, 1.385, 1.392, 1.437, 1.457, 1.475,
1.511, 1.677, 2, PV];
    ysmooth=[5.588, 7.939, 8.47, 9.226, 10.08, 10.67, 11.43, 12.64, 13.52, 14.43, 16.09, 17.64,
21.11, 24.32, 28.88, 32.31, 36.1, 39.3, 40.11, 39.44, 38.62, 37.54, 36.9, 35.16, 33.72, 32.81, 32,
28.07, 25.67, 24];

    interpx=0:stepsize:PV;                      % Calculate size to match amount of calculated
Mu_g,avg,app
    interpy=interp1(xsmooth,ysmooth,interpx);   % Interpolate lab data points to get data on every
interpx point

    labdata=transpose(interpy);                 % change from row to column vector for matrix
operation for cx
    cx=labdata./mu_gfs;                         % Calculate correction factors for every data point

    %   Plot correction factors vs PVI
    figure(2)
    plot(tD(1:tsteps-1,1),cx(1:tsteps-1,1));
    grid on
   % title('Correction factors vs. PVI')
    xlabel('t_D (PVI)')
    ylabel('Correction factor')
    hold on
```

### Calculate Mu_g,avg,app (with correction)

```
    for t=1:tsteps;
       for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfsx(t,1)=mean(mu_gfx(t,1:xgrid));
       end
    end
```

### Plot Mu_g,avg,app results

```
  figure(1)
    plot(tD(1:tsteps-1,1),mu_gfs(1:tsteps-1,1), 'r--');       % Plot Mu_g,avg,app vs. PVI without
correction
    hold on

    figure(1)
    plot(tD(1:tsteps-1,1),mu_gfsx(1:tsteps-1,1), 'k--');      % Plot Mu_g,avg,app vs. PVI with
correction

    Ma                                                       % Load raw lab data points of (Ma et
al.)
    figure(1)
    plot(xMa,yMa,'k.')                                       % Plot raw lab data points of (Ma et
al.)

    %  Full set smoothed lab Data of Ma et al. (used only for plotting)
    xsmoothfull=[0, 0.5, 0.6116, 0.7503, 0.8521, 0.8977, 0.9368, 0.9791, 1.002, 1.022, 1.051, 1.074,
1.117, 1.151, 1.196, 1.23, 1.28, 1.302, 1.332, 1.348, 1.367, 1.385, 1.392, 1.437, 1.457, 1.475,
1.511, 1.677, 2, 5];
    ysmoothfull=[5.588, 7.939, 8.47, 9.226, 10.08, 10.67, 11.43, 12.64, 13.52, 14.43, 16.09, 17.64,
21.11, 24.32, 28.88, 32.31, 36.1, 39.3, 40.11, 39.44, 38.62, 37.54, 36.9, 35.16, 33.72, 32.81, 32,
28.07, 25.67, 24];

    figure(1)
    plot(xsmoothfull,ysmoothfull,'--ob')                      % Plot full smoothed curve and points
    title('Gas apparent viscosity vs. PV injected')
    legend('Calculated Mu_g,avg,app', 'Corrected Mu_g,avg,app', 'Lab data points (Ma et
al.)','Smoothed lab data', 'Location','SE')
    xlabel('Dimensionless time (TPV)')
    ylabel('Foam average apparent viscosity (cP)')
    grid on
```

### Create contour plots with 1st correction

```
    figure(3)
    subplot(3,1,2)
    contSwx1=contour(tD(1:tsteps), linspace(0,L,xgrid), Swx', 500);           % 1st Corrected  water
saturations
    clabel(contSwx1);
    xlabel('t_D (PVI)')
    ylabel('x_D')
    %title('Contour plot with 1st corrected water saturations')
    grid on
    hold on

    figure(4)
    subplot(3,1,2)
    contRtx1=contour(tD(1:tsteps-1), linspace(0,L,xgrid), lambda_rtx', 500);    % 1st Corrected total
relative mob's
    clabel(contRtx1, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
    %title('Contour plot with 1st corrected total relative mobilities')
    grid on
    hold on
```

### 2nd correction

```
    cx2=((labdata./mu_gfsx).*cx);                            % Calculate 2nd correction factors for
every data point

    %   Calculate Sgx and Swx (with 2nd correction factor)
    for t=1:tsteps-1;
        for i=1:xgrid;
         %  Calculate rel perms, fractional flow, mobility and gas viscosity
```

```
[fwx(t,i),u_gx(t,i+1),mu_gfx2(t,i),krwx(t,i),krgfx(t,i),lambda_rtx2(t,i)]=fracflowfcx(cx2(t,1),Swx(t,
i),Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
```

```
         %  Calculate gas & water saturation at t+1
         Sgx(t+1,i)=(Sgx(t,i))+(u_gx(t,i)-u_gx(t,i+1))*dt/(dx*phi);
         Swx(t+1,i)=1-Sgx(t+1,i);
        end
    end

    for t=1:tsteps;
        for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfsx2(t,1)=mean(mu_gfx2(t,1:xgrid));
        end
    end


   figure(1)
    plot(tD(1:tsteps-1,1),mu_gfsx2(1:tsteps-1,1), 'c:');      % Plot Mu_g,avg,app vs. PVI with
correction
```

### 3rd correction

```
    cx3=((labdata./mu_gfsx2).*cx2);                         % Calculate 3rd correction factors for
every data point

    %   Calculate Sgx and Swx (with 3rd correction factor)
    for t=1:tsteps-1;
        for i=1:xgrid;
          %  Calculate rel perms, fractional flow, mobility and gas viscosity

[fwx(t,i),u_gx(t,i+1),mu_gfx3(t,i),krwx(t,i),krgfx(t,i),lambda_rtx3(t,i)]=fracflowfcx(cx3(t,1),Swx(t,
i),Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
          %  Calculate gas & water saturation at t+1
          Sgx(t+1,i)=(Sgx(t,i))+(u_gx(t,i)-u_gx(t,i+1))*dt/(dx*phi);
          Swx(t+1,i)=1-Sgx(t+1,i);
        end
    end

    %   Calculate Mu_g,avg,app (with 3rd correction)
    for t=1:tsteps;
        for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfsx3(t,1)=mean(mu_gfx3(t,1:xgrid));
        end
    end

     figure(1)
         plot(tD(1:tsteps-1,1),mu_gfsx3(1:tsteps-1,1), 'm--');      % Plot Mu_g,avg,app vs. PVI with
correction
```

### 4th correction

```
    cx4=((labdata./mu_gfsx3).*cx3);                         % Calculate 4th correction factors for
every data point

    %   Calculate Sgx and Swx (with 4th correction factor)
    for t=1:tsteps-1;
        for i=1:xgrid;
         %  Calculate rel perms, fractional flow, mobility and gas viscosity

[fwx(t,i),u_gx(t,i+1),mu_gfx4(t,i),krwx(t,i),krgfx(t,i),lambda_rtx4(t,i)]=fracflowfcx(cx4(t,1),Swx(t,
i),Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
         %  Calculate gas & water saturation at t+1
         Sgx(t+1,i)=(Sgx(t,i))+(u_gx(t,i)-u_gx(t,i+1))*dt/(dx*phi);
         Swx(t+1,i)=1-Sgx(t+1,i);
        end
    end

    %   Calculate Mu_g,avg,app (with 4th correction)
    for t=1:tsteps;
        for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfsx4(t,1)=mean(mu_gfx4(t,1:xgrid));
        end
    end

    figure(1)
    plot(tD(1:tsteps-1,1),mu_gfsx4(1:tsteps-1,1), 'g-.');      % Plot Mu_g,avg,app vs. PVI with
correction
    legend('Calculated Mu_g,avg,app', 'Corrected Mu_g,avg,app', 'Lab data points (Ma et
al.)','Smoothed lab data', '2nd correction', '3rd correction', '4th correction', 'Location','NE')
```

### Create contour plots (with 4th correction)

```
    figure(3)
    subplot(2,1,2)
    contSwx4=contour(tD(1:tsteps), linspace(0,L,xgrid), Swx', 750);           % 4th Corrected  water
saturations
    clabel(contSwx4, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
    %title('Contour plot with 4th corrected water saturations')
    grid on

    figure(4)
    subplot(3,1,3)
    contRtx4=contour(tD(1:tsteps-1), linspace(0,L,xgrid), lambda_rtx4', 500);    % 4th Corrected
total relative mob's
    clabel(contRtx4, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
    %title('Contour plot with 4th corrected total relative mobilities')
    grid on
```

### Plot correction factors

```
    figure(2)
    plot(tD(1:tsteps-1,1),cx2(1:tsteps-1,1), 'c');
    hold on
    xlabel('t_D (PVI)')
    ylabel('Correction factor')

    figure(2)
    plot(tD(1:tsteps-1,1),cx3(1:tsteps-1,1), 'm');
    hold on
    xlabel('t_D (PVI)')
    ylabel('Correction factor')

    figure(2)
    plot(tD(1:tsteps-1,1),cx4(1:tsteps-1,1), 'k');
    hold on
    grid on
    xlabel('t_D (PVI)')
    ylabel('Correction factor')

figure(2)
    axis([0 3 0 5]);
    legend('1st Correction', '2nd Correction', '3rd Correction', '4th Correction')
```

## FRACFLOW FUNCTION WITH CORRECTION FACTOR

```
function [fwx,ugx,mu_gfx,krwx,krgfx,lambda_rtx] =
fracflowfcx(cx,Swx,Swc,Sg_r,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_water,mu_gas,u_t)

% Calculate krw and krg (Brooks-Corey correlation)
if Swx>Swc;
       krwx=k_rwo*((Swx-Swc)/(1-Swc-Sg_r))^nw;
   else
       krwx=0;
end

if Swx<1-Sg_r;
        krgnfx=k_rgo*((1-Swx-Sg_r)/(1-Swc-Sg_r))^ng;
     else
        krgnfx=0;
end

% Calculate F2 (Water saturation dependent function)
    F2x=0.5+(atan(epdry*(Swx-fmdry)))/pi;

% Calculate FM (Mobility reduction factor)
    FMx=1/(1+fmmob*F2x*cx);

% Calculate gas rel perm with foam
    krgfx=krgnfx*FMx;

% Calculate the mobilities and total mobility
    lambda_gfx=krgfx/(mu_gas);
    lambda_wx=krwx/(mu_water);
    lambda_rtx=lambda_wx+lambda_gfx;

% Calculate fractional flow
    fwx=1/(1+(1/lambda_wx)*(lambda_gfx));

% Calculate water and gas superficial velocity
    uwx=fwx*u_t;
    ugx=u_t-uwx;

% Calculate gas apparent viscosity [cP]
    mu_gfx=1/((lambda_wx)+(lambda_gfx));
end
```

## MODEL WITH LINEAR CORRECTION FUNCTIONS

```
clear all; close all; clc

%%_________________________________________________________________________
%   Two-phase incompressible model to run simulations on Dyanamic SAG Foam Floods.
%   The model applies Buckley-Leverett flow for foam simulation. Gas
%   relative viscosities (or pressure drop) are calculated in the presence of foam
%   when injected into a fully water-saturated core sample.
%   The determined pressure drop and lab data is used to calculate a
%   correction factor to fit the late peak in the lab data of Ma et al.
%%_________________________________________________________________________
```

**Parameters**

```
Core sample parameters
    phi=0.36;                                   % Porosity
    L=1;                                        % size of the core used in the lab [ft]
    u_t=20;                                     % total velocity [ft/d]

    %   Corey exponents and rel.perm coefficients
    %   (from Fitting Foam Simulation Model Paramters for SAG Foam Applications, Persoff et al.
(1991); SPE 165282)
```

```
    mu_w=1.0;                                    % Water viscosity [cP]
    Swc=0.07;                                    % residual water saturation
    k_rwo=1;                                     % end point water relative permeability
    nw=1.96;                                     % Corey exponent for water relative permeability

    mu_g=0.02;                                   % Gas viscosity [cP]
    k_rgo=1;                                     % end point gas relative permeability
    ng=2.29;                                     % Corey exponent for gas relative permeability
function
    Sgr=0.0;                                     % residual gas saturation

    %   Discretisation parameters
    dt=1/(10*(24*60*60));                        % time step [s]
    tsteps=77760;                                % amount of time steps to reach certain PV, has to
correspond with PV below (3.5 PV=54432, 2 PV=31104)
    xgrid=100;                                    % amount of grid cells
    dx=L/xgrid;                                  % size of grid cells

    %   Foam parameters
    fmmob=4.72e4;                                % reference gas mobility reduction factor for wet
foams
    fmdry=0.101;                                 % water saturation in vicinity of which foam
collapses
    epdry=425;                                   % abruptness of foam dry-out effect

    %   Build matrices for calculations of Sw and Sg
    lambda_rt=zeros(tsteps-1,xgrid);               % total relative mobility
    krw=zeros(tsteps,xgrid);                     % water relative permeability
    krgnf=zeros(tsteps,xgrid);                   % gas relative permeability without foam
    krgf=zeros(tsteps,xgrid);                    % gas relative permeability in the presence of foam
    fw=zeros(tsteps,xgrid);                      % water fractional flow
    Sw=zeros(tsteps,xgrid);                      % water saturation
    Sg=zeros(tsteps,xgrid);                      % gas saturation
    u_g=zeros(tsteps,xgrid);                     % gas superficial velocity [m s^-1]
    mu_gf=zeros(tsteps,xgrid);                   % gas apparent viscosity with foam
    mu_gfs=zeros(tsteps,1);                      % average gas apparent viscosity over the whole core
    tD=zeros(tsteps,1);                          % dimensionless time (PVI)

    %   Build matrices for calculations of Swx and Sgx (with correction)
    lambda_rtx=zeros(tsteps-1,xgrid);            % total relative mobility with correction
    krwx=zeros(tsteps,xgrid);                    % water relative permeability with correction
    krgfx=zeros(tsteps,xgrid);                   % gas relative permeability in the presence of foam
with correction
    fwx=zeros(tsteps,xgrid);                     % water fractional flow with correction
    Swx=zeros(tsteps,xgrid);                     % water saturation with correction
    Sgx=zeros(tsteps,xgrid);                     % gas saturation with correction
    u_gx=zeros(tsteps,xgrid);                    % gas superficial velocity with correction [m s^-1]
    mu_gfx=zeros(tsteps,xgrid);                  % gas apparent viscosity with foam with correction
    mu_gfsx=zeros(tsteps,1);                     % average gas apparent viscosity  with correction
over the whole core

    %   Build viscosity matrices for linear cx corrections
    mu_gfxlin=zeros(tsteps,xgrid);               % gas apparent viscosity with foam with correction
    mu_gfsxlin=zeros(tsteps,1);                  % average gas apparent viscosity  with correction
over the whole core
    mu_gfxlin2=zeros(tsteps,xgrid);              % gas apparent viscosity with foam with correction
    mu_gfsxlin2=zeros(tsteps,1);                 % average gas apparent viscosity  with correction
over the whole core
    lambda_rtx1=zeros(tsteps-1,xgrid);            % total relative mobility with correction
    lambda_rtx2=zeros(tsteps-1,xgrid);            % total relative mobility with correction
```

### Initial & Boundary Conditions

```
    Sw(:,:)=1;                                   % initially the core is fully water saturated
    Sg(:,:)=0;                                   % initially no gas has been injected in the core
    u_g(:,1)=u_t;                                % injection rate at the first boundary

    %   for calculations with correction
    Swx(:,:)=1;                                  % initially the core is fully water saturated
    Sgx(:,:)=0;                                  % initially no gas has been injected in the core
    u_gx(:,1)=u_t;                               % injection rate at the first boundary

    %   for calculations with 2nd correction
    Swx2(:,:)=1;                                  % initially the core is fully water saturated
    Sgx2(:,:)=0;                                  % initially no gas has been injected in the core
    u_gx2(:,1)=u_t;                               % injection rate at the first boundary
```

## Calculate Sg and Sw (without correction)

```
    for t=1:tsteps-1;
        for i=1:xgrid;
         %  Calculate rel. perms, fractional flow, mobility and gas viscosity

[fw(t,i),u_g(t,i+1),mu_gf(t,i),krw(t,i),krgnf(t,i),krgf(t,i),lambda_w(t,i),lambda_gf(t,i),lambda_rt(t
,i)]=fracflowf(Sw(t,i),Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
         %  Calculate gas & water saturation at t+1
         Sg(t+1,i)=(Sg(t,i))+(u_g(t,i)-u_g(t,i+1))*dt/(dx*phi);
         Sw(t+1,i)=1-Sg(t+1,i);
        end
    end
```

## Calculate Mu_g,avg,app (without correction)

```
    for t=1:tsteps;
        for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfs(t,1)=mean(mu_gf(t,1:xgrid));
        end
     end

%    mu_gfs=importdata('mu_gfs.mat');               % Load precalculated mu_gfs to save time (x=10)
```

## Create contour plots (without correction)

```
  figure(3)
    subplot(3,1,1)
    contSw=contour(tD(1:tsteps), linspace(0,L,xgrid), Sw', 1000);           % uncorrected water
saturations
   clabel(contSw, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
   % title('Contour plot of uncorrected water saturations')
    grid on
    hold on

  figure(4)
    subplot(3,1,1)
    contRt=contour(tD(1:tsteps-1), linspace(0,L,xgrid), lambda_rt', 1000);    % uncorrected total
relative mob's
    clabel(contRt, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
  %  title('Contour plot of uncorrected total relative mobilities')
    grid on
    hold on
```

## Interpolate lab data (Ma et al.) and Calculate correction factor cx

```
Simulation parameters
    PV=5;                                        % Untill which PV to run the interpolation, has to
correspond to tsteps above (Note: Will change automatically for last entry in xsmooth array)
    stepsize=(PV/(tsteps-1));

    %   Smoothed data of Ma et al. (used for calculation)
    xsmooth=[0, 0.5, 0.6116, 0.7503, 0.8521, 0.8977, 0.9368, 0.9791, 1.002, 1.022, 1.051, 1.074,
1.117, 1.151, 1.196, 1.23, 1.28, 1.302, 1.332, 1.348, 1.367, 1.385, 1.392, 1.437, 1.457, 1.475,
1.511, 1.677, 2, PV];
    ysmooth=[5.588, 7.939, 8.47, 9.226, 10.08, 10.67, 11.43, 12.64, 13.52, 14.43, 16.09, 17.64,
21.11, 24.32, 28.88, 32.31, 36.1, 39.3, 40.11, 39.44, 38.62, 37.54, 36.9, 35.16, 33.72, 32.81, 32,
28.07, 25.67, 24];
```

```
    interpx=0:stepsize:PV;                        % Calculate size to match amount of calculated
Mu_g,avg,app
    interpy=interp1(xsmooth,ysmooth,interpx);   % Interpolate lab data points to get data on every
interpx point

    labdata=transpose(interpy);                 % change from row to column vector for matrix
operation for cx
    cx=labdata./mu_gfs;                         % Calculate correction factors for every data point

%     %   Plot correction factors vs PVI
    figure(2)
    plot(tD(1:tsteps-1,1),cx(1:tsteps-1,1), '--');
    grid on
    title('Correction factors vs. PVI')
    xlabel('PVI')
    ylabel('Correction factor cx')
```

**Calculate Sgx and Swx (with correction factor)**

```
    for t=1:tsteps-1;
        for i=1:xgrid;
         %  Calculate rel perms, fractional flow, mobility and gas viscosity

[fwx(t,i),u_gx(t,i+1),mu_gfx(t,i),krwx(t,i),krgfx(t,i),lambda_rtx(t,i)]=fracflowfcx(cx(t,1),Swx(t,i),
Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
         %  Calculate gas & water saturation at t+1
         Sgx(t+1,i)=(Sgx(t,i))+(u_gx(t,i)-u_gx(t,i+1))*dt/(dx*phi);
         Swx(t+1,i)=1-Sgx(t+1,i);
        end
    end
```

**Calculate Mu_g,avg,app (with correction)**

```
    for t=1:tsteps;
        for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfsx(t,1)=mean(mu_gfx(t,1:xgrid));
        end
    end
```

**Create contour plots (with correction)**

```
    figure(3)
    subplot(2,2,2)
    contSwx=contour(tD(1:tsteps), linspace(0,L,xgrid), Swx', 25);           % Corrected water
saturations
   % clabel(contSwx);
    xlabel('t_D (PVI)')
    ylabel('x_D')
    title('Contour plot of cx corrected water saturations')
    grid on
    hold on

  figure(4)
    subplot(2,2,2)
    contRtx=contour(tD(1:tsteps-1), linspace(0,L,xgrid), lambda_rtx', 300);    % Corrected total
relative mob's
    clabel(contRtx);
    xlabel('t_D (PVI)')
    ylabel('x_D')
    title('Contour plot of cx corrected total relative mobilities')
    grid on
    hold on
```

**Linear cx correction #1**

```
    cxlin(1:20840)=0:(1/20839):1;
    cxlin(1,20841:77760)=1;
    cx=transpose(cxlin);

   %   Calculate Sgx and Swx (with linear correction factor #1)
    for t=1:tsteps-1;
```

```
        for i=1:xgrid;
         %  Calculate rel perms, fractional flow, mobility and gas viscosity

[fwx(t,i),u_gx(t,i+1),mu_gfxlin(t,i),krwx(t,i),krgfx(t,i),lambda_rtx1(t,i)]=fracflowfcx(cx(t,1),Swx(t
,i),Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
         %  Calculate gas & water saturation at t+1
         Sgx(t+1,i)=(Sgx(t,i))+(u_gx(t,i)-u_gx(t,i+1))*dt/(dx*phi);
         Swx(t+1,i)=1-Sgx(t+1,i);
        end
    end

    for t=1:tsteps;
       for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfsxlin(t,1)=mean(mu_gfxlin(t,1:xgrid));
       end
    end
```

**Create contour plots (with linear correction factor #1)**

```
    figure(3)
    subplot(3,1,2)
    contSwx1=contour(tD(1:tsteps), linspace(0,L,xgrid), Swx', 1000);           % Corrected lin1 water
saturations
    clabel(contSwx1, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
   % title('Contour plot of lin#1 corrected water saturations')
    grid on
    hold on

  figure(4)
    subplot(3,1,2)
    contRtx1=contour(tD(1:tsteps-1), linspace(0,L,xgrid), lambda_rtx1', 1000);    % Corrected lin1
total relative mob's
    clabel(contRtx1, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
   % title('Contour plot of lin#1 corrected total relative mobilities')
    grid on
    hold on
```

**Linear cx correction #2**

```
    cxlin2(1:20840)=0:(1/(20839/1.35)):1.35;
    cxlinleftright=1:(1/(56919/0.35)):1.35;
    cxlin2(20841:77760)=fliplr(cxlinleftright);
    cx=transpose(cxlin2);

    %   Calculate Sgx and Swx (with linear correction factor #2)
    for t=1:tsteps-1;
        for i=1:xgrid;
         %  Calculate rel perms, fractional flow, mobility and gas viscosity

[fwx(t,i),u_gx(t,i+1),mu_gfxlin2(t,i),krwx(t,i),krgfx(t,i),lambda_rtx2(t,i)]=fracflowfcx(cx(t,1),Swx(
t,i),Swc,Sgr,k_rwo,k_rgo,nw,ng,epdry,fmdry,fmmob,mu_w,mu_g,u_t);
         %  Calculate gas & water saturation at t+1
         Sgx(t+1,i)=(Sgx(t,i))+(u_gx(t,i)-u_gx(t,i+1))*dt/(dx*phi);
         Swx(t+1,i)=1-Sgx(t+1,i);
        end
    end

    for t=1:tsteps;
       for i=1:xgrid;
         tD(t,1)=u_t*dt*t/(L*phi);
         mu_gfsxlin2(t,1)=mean(mu_gfxlin2(t,1:xgrid));
       end
    end
```

**Create contour plots (with linear correction factor #2)**

```
    figure(3)
    subplot(3,1,3)
    contSwx2=contour(tD(1:tsteps), linspace(0,L,xgrid), Swx', 1000);           % Corrected lin2 water
saturations
   clabel(contSwx2, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
 %   title('Contour plot of lin#2 corrected water saturations')
    grid on

  figure(4)
    subplot(3,1,3)
    contRtx2=contour(tD(1:tsteps-1), linspace(0,L,xgrid), lambda_rtx2', 1000);    % Corrected lin2
total relative mob's
   clabel(contRtx2, 'manual');
    xlabel('t_D (PVI)')
    ylabel('x_D')
 %   title('Contour plot of lin#2 corrected total relative mobilities')
    grid on
```

**Plot Mu_g,avg,app results**

```
    figure(1)
    plot(tD(1:tsteps-1,1),mu_gfs(1:tsteps-1,1), 'r.');       % Plot Mu_g,avg,app vs. PVI without
correction
    hold on

    figure(1)
    plot(tD(1:tsteps-1,1),mu_gfsx(1:tsteps-1,1), 'g.');      % Plot Mu_g,avg,app vs. PVI with
correction

    Ma                                                       % Load raw lab data points of (Ma et
al.)
    figure(1)
    plot(xMa,yMa,'k.')                                       % Plot raw lab data points of (Ma et
al.)

    %   Full set smoothed lab Data of Ma et al. (used only for plotting)
    xsmoothfull=[0, 0.5, 0.6116, 0.7503, 0.8521, 0.8977, 0.9368, 0.9791, 1.002, 1.022, 1.051, 1.074,
1.117, 1.151, 1.196, 1.23, 1.28, 1.302, 1.332, 1.348, 1.367, 1.385, 1.392, 1.437, 1.457, 1.475,
1.511, 1.677, 2, 5];
    ysmoothfull=[5.588, 7.939, 8.47, 9.226, 10.08, 10.67, 11.43, 12.64, 13.52, 14.43, 16.09, 17.64,
21.11, 24.32, 28.88, 32.31, 36.1, 39.3, 40.11, 39.44, 38.62, 37.54, 36.9, 35.16, 33.72, 32.81, 32,
28.07, 25.67, 24];

    figure(1)
    plot(xsmoothfull,ysmoothfull,'--ob')                      % Plot full smoothed curve and points
    %title('Gas apparent viscosity vs. PV injected')
    legend('Model Mu_g,avg,app', 'Corrected Mu_g,avg,app', 'Lab data points (Ma et al.)','Smoothed
lab data (Ma et al.)', 'Location','NE')
    xlabel('Dimensionless time (TPV)')
    ylabel('Foam average apparent viscosity (cP)')
    grid on

    figure(1)
    plot(tD(1:tsteps-1,1),mu_gfsxlin(1:tsteps-1,1), 'c.');      % Plot Mu_g,avg,app vs. PVI with
linear correction #1
    plot(tD(1:tsteps-1,1),mu_gfsxlin2(1:tsteps-1,1), 'm.');      % Plot Mu_g,avg,app vs. PVI with
linear correction #2
    legend('Calculated Mu_g,avg,app', 'Corrected Mu_g,avg,app', 'Lab data points (Ma et
al.)','Smoothed lab data', 'Linear correction #1', 'Linear correction #2', 'Location','NE')
```

## LOAD DATA OF MA ET AL.

```
% Initialize variables.
filename = 'Ma.csv';  % Change this to correct directory where file is located
delimiter = {'\t',';'};
startRow = 2;

% Read columns of data as strings:
% For more information, see the TEXTSCAN documentation.
formatSpec = '%s%s%[^\n\r]';

% Open the text file.
fileID = fopen(filename,'r');

% Read columns of data according to format string.
% This call is based on the structure of the file used to generate this
% code. If an error occurs for a different file, try regenerating the code
% from the Import Tool.
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, 'HeaderLines' ,startRow-1,
'ReturnOnError', false);

% Close the text file.
fclose(fileID);

% Convert the contents of columns containing numeric strings to numbers.
% Replace non-numeric strings with NaN.
raw = [dataArray{:,1:end-1}];
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=[1,2]
    % Converts strings in the input cell array to numbers. Replaced non-numeric
    % strings with NaN.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Create a regular expression to detect and remove non-numeric prefixes and
        % suffixes.
        regexstr = '(?<prefix>.*?)(?<numbers>([-]*(\d+[\.]*)+[\,]{0,1}\d*[eEdD]{0,1}[-
+]*\d*[i]{0,1})|([-]*(\d+[\.]*)*[\,]{1,1}\d+[eEdD]{0,1}[-+]*\d*[i]{0,1}))(?<suffix>.*)';
        try
            result = regexp(rawData{row}, regexstr, 'names');
            numbers = result.numbers;

            % Detected commas in non-thousand locations.
            invalidThousandsSeparator = false;
            if any(numbers=='.');
                thousandsRegExp = '^\d+?(\.\d{3})*\,{0,1}\d*$';
                if isempty(regexp(thousandsRegExp, '.', 'once'));
                    numbers = NaN;
                    invalidThousandsSeparator = true;
                end
            end
            % Convert numeric strings to numbers.
            if ~invalidThousandsSeparator;
                numbers = strrep(numbers, '.', '');
                numbers = strrep(numbers, ',', '.');
                numbers = textscan(numbers, '%f');
                numericData(row, col) = numbers{1};
                raw{row, col} = numbers{1};
            end
        catch me
        end
    end
end


% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells
raw(R) = {NaN}; % Replace non-numeric cells

% Allocate imported array to column variable names
```

```
xMa = cell2mat(raw(:, 1));
yMa = cell2mat(raw(:, 2));

% Clear temporary variables
clearvars filename delimiter startRow formatSpec fileID dataArray ans raw numericData col rawData row
regexstr result numbers invalidThousandsSeparator thousandsRegExp me R;

% Plot data
figure(5)
plot(xMa,yMa, 'ko', 'MarkerFaceColor','k','MarkerSize', 7)
legend('Experimental data', 'Location','NE')
xlabel('Dimensionless time (TPV)', 'fontsize', 12)
ylabel('Foam apparent viscosity (cP)', 'fontsize', 12)
grid on
axis([0 5 0 45])
```

## SMOOTHED DATA OF MA ET AL.

```
 %  Smoothed Experintal Data of Ma et al. (2013)
xsmooth=[0, 0.5, 0.6116, 0.7503, 0.8521, 0.8977, 0.9368, 0.9791, 1.002, 1.022, 1.051, 1.074, 1.117,
1.151, 1.196, 1.23, 1.243, 1.302, 1.332, 1.348, 1.367, 1.385, 1.403, 1.421, 1.437, 1.457, 1.475,
1.493, 1.511, 1.547, 1.559, 1.601, 1.68, 1.709, 1.746, 2, 2.5, 3, 3.5, 5];

ysmooth=[5.588, 7.939, 8.47, 9.226, 10.08, 10.67, 11.43, 12.64, 13.52, 14.43, 16.09, 17.64, 21.11,
24.32, 28.88, 32.31, 36.1, 39.3, 40.11, 39.44, 38.62, 37.54, 37, 36.36, 35.16, 33.72, 32.81, 31.73,
30.74, 29.87, 28.48, 27.2, 25.84, 25.27, 24.58, 24, 24, 24, 24, 24];

%   Plot Data points and smoothed curve
figure(5) [x,ind] = sort(x); plot(x,y,'b.',x,y(ind),'r-')
xlabel('PV'); ylabel('Average apparent viscosity (cp)')

%  Load original experimental data and add to plot
Ma
hold on
figure(5) plot(xMa,yMa, 'g.') title('Original and smoothed data from Ma et al.') legend('Selected
Data points','Smoothed Curve', 'Original data Ma et al.', 'Location','NE') grid on axis([-1 5 0 45])
```