University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Petroleum Engineering / Reservoir Technology	Spring semester, 2013 Open
Writer: Ole Skretting Helvig	……………………………………… (Writer's signature)

Faculty supervisor:

Steinar Evje

Title of thesis:

Models for water flooding, imbibition and coupled fracture-matrix flow in a fractured reservoir

Credits (ECTS): 30

Key words: Models for water flooding Buckley-Leverett theory Capillary pressure correlations Capillary pressure effect Transfer term Numerical simulation	Pages: 49 + enclosure: 17 Stavanger, 17.06.2013

Models for water flooding, imbibition and coupled fracture-matrix flow in a fractured reservoir

A Master-thesis in petroleum engineering

By Ole Skretting Helvig

University of Stavanger

Spring 2013

Abstract

The study of reservoir performance during waterflooding is important to reservoir engineers. Numerous analytical, semi-analytical and numerical flow models with different assumptions have been submitted over the years, aiming to model and describe reservoir behavior and flow dynamics during an oil/water displacement process. Studying the effect that controlling forces such as viscous, gravity and capillary forces have on water saturation profiles, breakthrough time and oil recovery are important features of these models.

In this thesis a derivation and an analytical solution procedure of the Buckley-Leverett equation is presented.

A Buckley-Leverett model that includes capillary pressure is derived and solved numerically. The effect capillary forces have on saturation profiles, breakthrough time and oil recovery will be illustrated for different capillary pressures correlations and by varying a dimensionless number consisting of controlling flow parameters such as injection flow rate, fluid viscosity, length of porous media and capillary pressure.

Also a derivation and numerical solution of a model for coupled fracture-matrix flow in fractured reservoir will be presented. Modified Buckley-Leverett theory including a time dependent transfer term that takes into account fluid exchange rate between matrix and fracture is used to simulate this waterflooding process. A demonstration of fracture, matrix and total oil recovery will be illustrated for a given case. Additionally, some effects that strong versus weak spontaneous imbibition have on fracture saturation profiles, breakthrough time and oil recovery will be investigated.

Context

1 Introduction 6

1.1 Objectives 6
1.2 Acknowledgement 6

2 The two-phase porous media equation in 1D 7

2.1 Derivation of the model 8
  2.1.1 Derivation of the Buckley-Leverett equation from mass balance 8
  2.1.2 Dimensionless variables 11
  2.1.3 Corey relative Permeabilities 12
  2.1.4 Fractional Flow 13
2.2 Analytical solution of the Buckley-Leverett equation 14
  2.2.1 Analytic solution procedure 14
2.3 Numerical solution 20

3 The Buckley-Leverett equation including capillary pressure 21

3.1 Derivation of the model 21
3.2 Dimensionless variables 25
3.3 Discretization of the model 26
  3.3.1 Discretization at the boundary 27
3.4 Stability 28
3.5 Capillary pressure correlations 29
  3.5.1 Capillary pressure correlation for mixed-wet reservoirs 29
  3.5.2 A new versatile capillary pressure correlation 30
3.6 Water saturation profiles- Effects of including capillary pressure 32
3.7 Capillary end effects 36

4 A new model with modified Buckley-Leverett theory……………………………………………37

4.1 Derivation of the modified Buckley-Leverett Model (BLM)……………………………37
4.2 Dimensionless form of BLM……………………………………………………………………39
4.3 Discretization of BLM…………………………………………………………………………40
4.4 Numerical investigation………………………………………………………………………44

References……………………………………………………………………………………………48

Appendix

A Matlab codes for analytic solution 50
B Matlab codes for numerical solution of the Buckley-Leverett equation including capillary pressure 55
C Matlab codes for numerical solution of the modified Buckley-Leverett equation 61

1 Introduction

1.1 Objectives

The thesis is divided into mainly 3 chapters.

- In chapter two the Buckley-Leverett equation is derived and an analytical solution procedure is presented.
- In chapter three the model in chapter two is expanded to also account for capillary forces. The model is derived again and numerical solutions are produced to demonstrate this capillary effect.
- The fourth and last chapter presents a model for water injection in fractured porous media using modified Buckley-Leverett theory, and some characteristic effects are studied.
- The thesis also includes an Appendix where Matlab codes used in order to obtain figures are listed. Reference to appendix is seen as, for example, [A1] in the figure text.

1.2 Acknowledgement

The thesis was written with excellent guidance from Steinar Evje at University of Stavanger.

2 The two-phase porous media equation in 1D

This chapter consists mainly of previous work from [1], except for section 2.3. Some minor modifications have been made and all figures have been somewhat improved.

[1] We will in this chapter consider a line between the injection well and the producing well, with length L and constant cross section A. As injected water follows the pathway, we will study how the saturation changes as a function of both distance, x, and time, t, using the non-linear hyperbolic partial differential equation, the Buckley-Leverett equation.

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial x} = 0$$

Several assumptions are needed in order to derive the model and obtain an analytical solution. These are

- Immiscible flow of two incompressible fluids, water and oil.
- Homogeneous, incompressible reservoir, i.e. constant porosity φ and constant absolute permeability, K.
- Capillary pressure is zero.
- Gravity forces are neglected.
- Reservoir initially filled with oil only.
- Constant injection rate of water at one end and production at the other end.
- A unique physical solution exists.
- Mass is conserved.

2.1 Derivation of the model

2.1.1 Derivation of the Buckley-Leverett equation from mass balance

[1] Consider an infinitesimal element of rock having porosity φ, an area A, and a length Δx in the direction of flow as shown in figure 1.

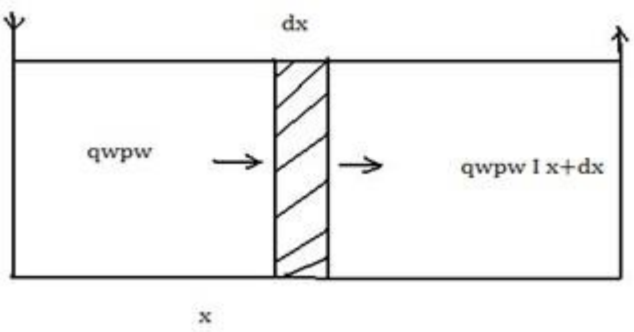


Figure 1 Mass flow rate of water through a linear volume element Aφdx.

The mass rate of water entering the element at x is $(q_w\rho_w)_x$. Where q_w and ρ_w denotes volume rate and density respectively. At $x + \Delta x$, the mass rate of water leaving the element is $(q_w\rho_w)_{x+\Delta x}$. The mass rate of water accumulating is $A\varphi\Delta x\frac{\partial}{\partial t}(S_w\rho_w)$, where S_w is the water saturation.

Due to mass balance, the mass of water entering the element in the rock, minus the mass of water leaving must be equal to the rate of water accumulation in the element. We thereby have:

$$(q_w\rho_w)_x - (q_w\rho_w)_{x+\Delta x} = A\varphi\Delta x\frac{\partial}{\partial t}(S_w\rho_w). \tag{2.1.1}$$

For a given function f, we have by definition, when $\Delta x \to 0$:

$$\frac{f(x,t) - f(x+\Delta x, t)}{\Delta x} = -\frac{\partial f}{\partial x}. \tag{2.1.2}$$

Hence we can write:

$$\frac{\partial}{\partial x}(q_w\rho_w) + A\varphi\frac{\partial}{\partial t}(S_w\rho_w) = 0. \tag{2.1.3}$$

Due to the assumption that the fluid are incompressible, the densities, ρ_w and ρ_o, will be constant, both as function of time and distance. This leads us to:

$$\frac{\partial}{\partial x}(q_w) + A\varphi\frac{\partial}{\partial t}(S_w) = 0, \quad \frac{\partial}{\partial x}(q_o) + A\varphi\frac{\partial}{\partial t}(S_w) = 0. \tag{2.1.4}$$

Darcy's equations for oil and water respectively, are written as

$$q_o = -\frac{Kk_{ro}A}{\mu_o}\left(\frac{\partial p_o}{\partial x}\right), \tag{2.1.5}$$

$$q_w = -\frac{Kk_{rw}A}{\mu_w}\left(\frac{\partial p_w}{\partial x}\right), \tag{2.1.6}$$

where K is the absolute permeability, k_{ro} and k_{rw} are the relative permeabilities for oil and water respectively, and where p_o and p_w are the phase pressures for oil and water respectively.

Inserting equation (2.1.5) and (2.1.6) into equation (2.1.4) gives

$$\frac{\partial}{\partial x}\left(\frac{kk_{rw}}{\mu_w}\frac{\partial p}{\partial x}\right) = \varphi\frac{\partial S_w}{\partial t}, \tag{2.1.7}$$

$$\frac{\partial}{\partial x}\left(\frac{kk_{ro}}{\mu_o}\frac{\partial p}{\partial x}\right) = \varphi\frac{\partial S_o}{\partial t}. \tag{2.1.8}$$

Next we introduce the fluid mobility, λ_w and λ_o, as functions of effective permeability divided by viscosity.

$$\lambda_w(S) = \frac{Kk_{rw}(S)}{\mu_w}\ , \lambda_o(S) = \frac{Kk_{ro}(S)}{\mu_o}.$$

By doing so, equation (2.1.7) and (2.1.8) becomes

$$\frac{\partial}{\partial x}\left(\lambda_w\frac{\partial p}{\partial x}\right) = \varphi\frac{\partial S_w}{\partial t}\ , \quad \frac{\partial}{\partial x}\left(\lambda_o\frac{\partial p}{\partial x}\right) = \varphi\frac{\partial S_o}{\partial t}.$$

Using the constraint; $S_w + S_o = 1$ and by adding the above equations we obtain

$$\frac{\partial}{\partial x}\left(\lambda_T\frac{\partial p}{\partial x}\right) = 0, \qquad \lambda_T = \lambda_w + \lambda_o,$$

which implies that

$$u = -\lambda_T \frac{\partial p}{\partial x}, \tag{2.1.9}$$

is independent of position x, i.e. constant.

The Darcy velocities, or fluid flux are written as

$$u_w = -\lambda_w \frac{\partial p}{\partial x}, \quad u_o = -\lambda_o \frac{\partial p}{\partial x}.$$

$u = u_w + u_o$ is the total flux and with (2.1.9) yields

$$\frac{\partial p}{\partial x} = -\frac{u}{\lambda_T}. \tag{2.1.10}$$

Eliminating the pressure term in the mass conservation equation for water using relation (2.1.10), give us the Buckley-Leverett equation. Mass balance of water is expressed by

$$\varphi \frac{\partial S_w}{\partial t} - \frac{\partial}{\partial x}(\lambda_w \frac{\partial p}{\partial x}) = 0,$$

and the result after substitution becomes

$$\varphi \frac{\partial S}{\partial t} + u \frac{\partial f(S)}{\partial x} = 0, \tag{2.1.11}$$

where $S = S_w$ is the water saturation and $f(S)$ denotes water fractional flow function defined by

$$f(S) = \frac{\lambda_w(S)}{\lambda_t(S)} = \frac{k_{rw}(S)/\mu_w}{k_{rw}(S)/\mu_w + k_{ro}(S)/\mu_o}. \tag{2.1.12}$$

2.1.2 Dimensionless variables

[1] It is useful to introduce dimensionless position x_D and dimensionless time t_D

Let $x_D = \frac{x}{L}, and\ t_D = \frac{u}{\varphi L} t.$

We have that

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial x_D}\frac{\partial x_D}{\partial x} = \frac{\partial f}{\partial x_D}\frac{1}{L}.$$

Equation (2.1.11) becomes

$$\frac{\varphi L}{u}\left(\frac{\partial S}{\partial t}\right) + \frac{\partial f}{\partial x_D} = 0, \tag{2.1.13}$$

and

$$\frac{\partial S}{\partial x} = \frac{\partial S}{\partial t_D}\frac{\partial t_D}{\partial x} = \frac{\partial S}{\partial t_D}\frac{u}{\varphi L}.$$

Inserting the above equation, and rearranging, equation (2.1.11) can be written as

$$\frac{\partial S}{\partial t_D} + \frac{\partial f(S)}{\partial x_D} = 0.$$

Without the subscript D, the dimensionless Buckley-Leverett equation is written as

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial x} = 0. \tag{2.1.14}$$

From now on, unless otherwise stated, dimensionless variables are used, implying that

$0 \leq x \leq 1$ and $t \geq 0$.

2.1.3 Corey relative permeabilities

[1] Corey relative permeabilities are an often used approximation of relative permeabilities. This uses few parameters that can be defined. Normalized Corey type relative permeabilities are specified using expressions

$$k_{ro} = KRO(1-S)^{no}, \qquad (2.1.15)$$

$$k_{rw} = (KRW)S^{nw}. \qquad (2.1.16)$$

Since we have neglected residual saturations, we may simplify by setting the end point permeabilities, $KRO = \ KRW = 1$. The homogenous reservoir assumption also suggests constant values for the Corey exponents, no and nw. They are normally chosen with no and $nw \geq 1$. If Corey relative permeabilities are used, $f(S)$ is completely determined by specifying Corey exponents nw, no and viscosity ratio, M.

In this chapter, we set $M = 0.5$ and $nw = no = 2$, as these are common values.

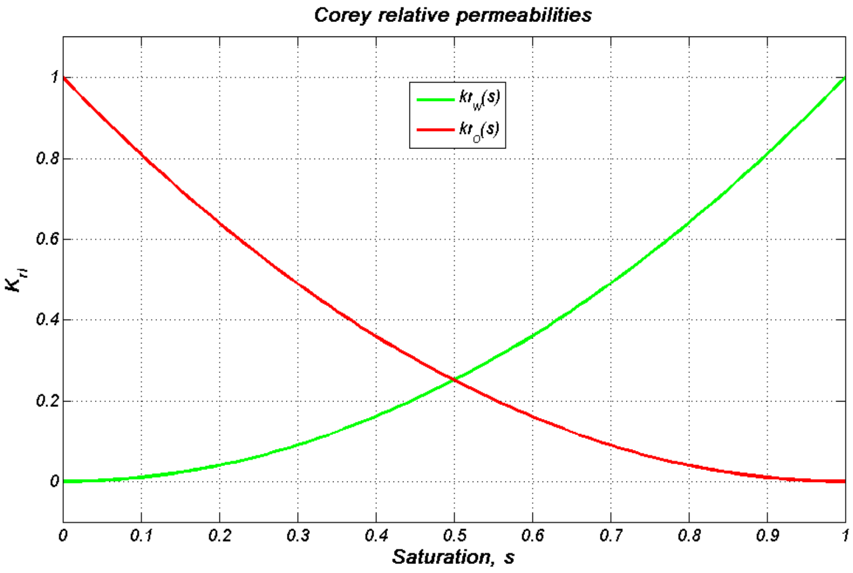


Figure 2:[A1] Plot of Corey relative permeabilities, given by equation (2.1.15) and (2.1.16).

2.1.4 Fractional flow

[1] We want the fractional flow equation (2.1.12) to be a function of the water saturation, S. Rearranging (2.1.12) we obtain

$$f(S) = \frac{k_{rw}}{k_{rw} + Mk_{ro}}, \tag{2.1.17}$$

where $k_{rw} = S^{nw}$ and $k_{ro} = (1 - S)^{no}$ and also viscosity ratio, $M = \frac{\mu_w}{\mu_o}$.

The flux function then becomes

$$f(S) = \frac{S^{nw}}{S^{nw} + M(1 - S)^{no}}, \tag{2.1.18}$$

and the derivative might be written as

$$f'(S) = \frac{M(1 - S)^{nw-1} * S^{no-1}[no(1 - S) + S_{nw}]}{(S^{no} + M(1 - S)^{nw})^2}. \tag{2.1.19}$$

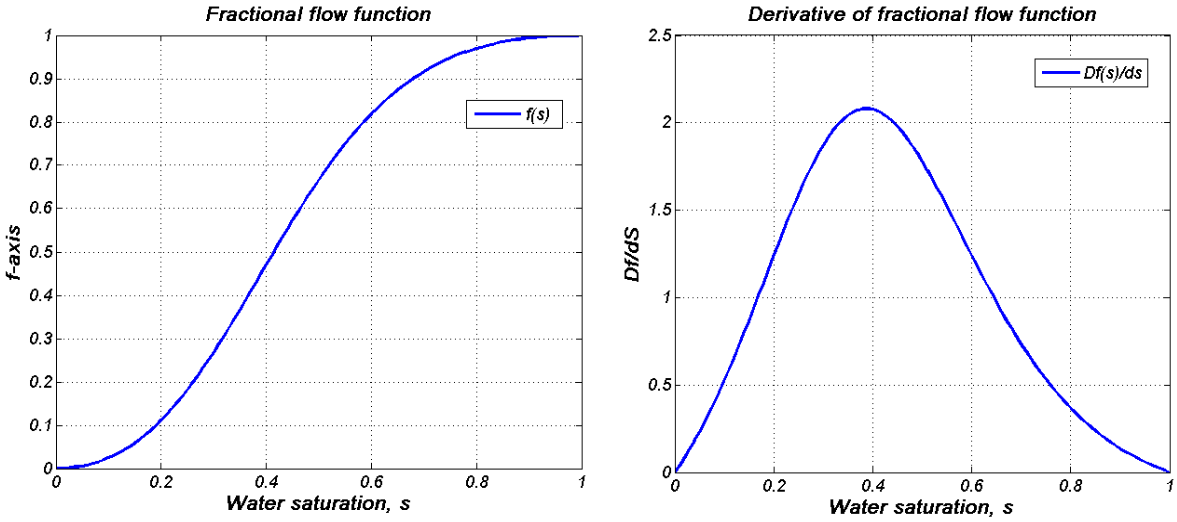


Figure 3:[A2] Fractional flow curve, equation (2.1.18) (left) and its derivative [A3], equation (2.1.19) (right).

2.2 Analytical solution of the Buckley-Leverett equation

[1] In this chapter we lay out a procedure for obtaining the analytical solution of equation (2.1.14). In simple steps this is to:

1. Determine the front saturation height S_f
2. Find the position x_f of S_f
3. Find the saturation distribution behind the front

2.2.1 Solution procedure

Determination of the front saturation height

[1] We draw the saturation movement by multiplying the derivative of the fractional flow function with elapsed time T. Unless otherwise stated, we set T=0.5.

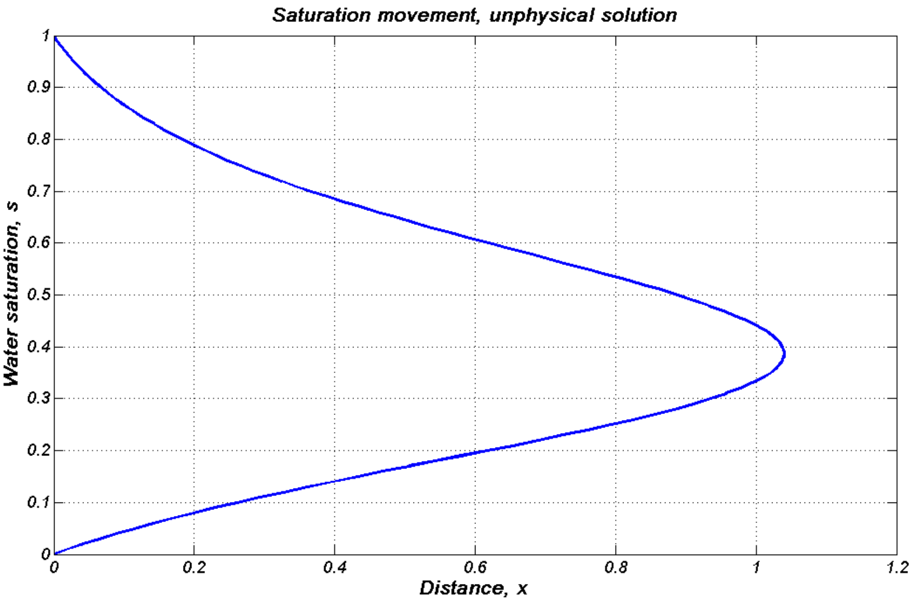


Figure 1: [A4] Saturation movement, unphysical solution.

Mathematically this suggest that for any given position x, there is multiple values for the saturation. Physically this cannot be true.

Unique solution

[1] In order to obtain a unique solution we use a material balancing argument. The amount of water injected in the well, must be equal in either solution, being physical or non-physical.

Looking at figure 5, we see that area $_{\text{non-physical}}$ can be found by

$$Area_{non-physical} = \int_{0}^{1} f'(S)Tds = T\big(f(1) - f(0)\big) = T,$$

and

$$Area_{physical} = S_f * x_f + \int_{S_f}^{1} f'(S)Tds = S_f f'(S_f)T + T\left(1 - f(S_f)\right).$$

So we set

$$Area_{physical} = Area_{non-physical}$$

$$S_f f'(S_f)T + T(1 - f(S_f) = T$$

$$T\left[S_f f'(S_f) + \left(1 - f(S_f)\right)\right] = T$$

$$S_f f'(S_f) + 1 - f(S_f) = 1.$$

And we find the relation

$$f'(S_f) = \frac{f(S_f)}{S_f}. \tag{2.2.1}$$

This means that if this is true, both areas are equal.

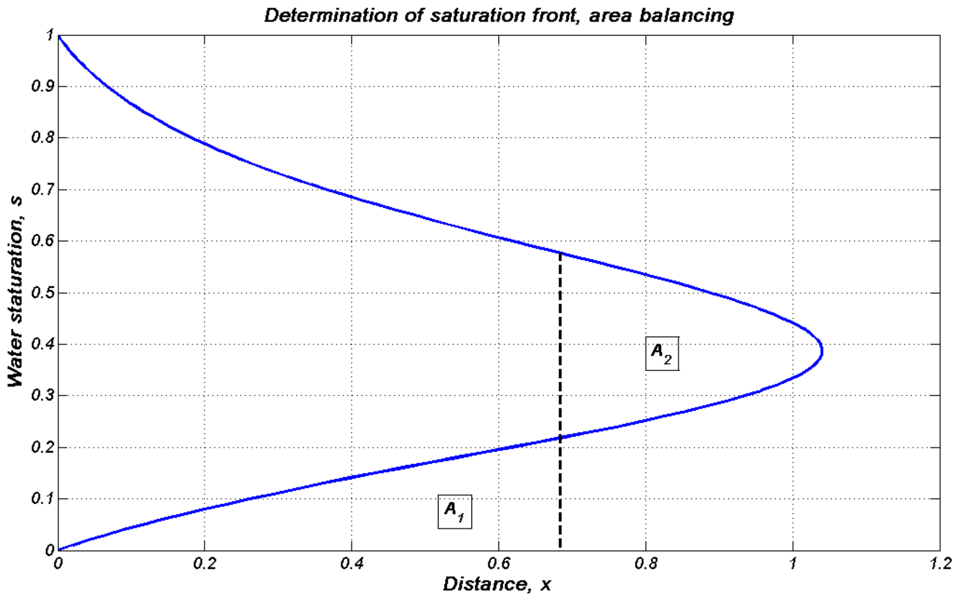


Figure 2: [A5] Area balancing

We can determine the front saturation height, S_f, from equation (2.2.1), by

$$g(S) = f'(S) - \frac{f(S)}{S} = 0. \tag{2.2.2}$$

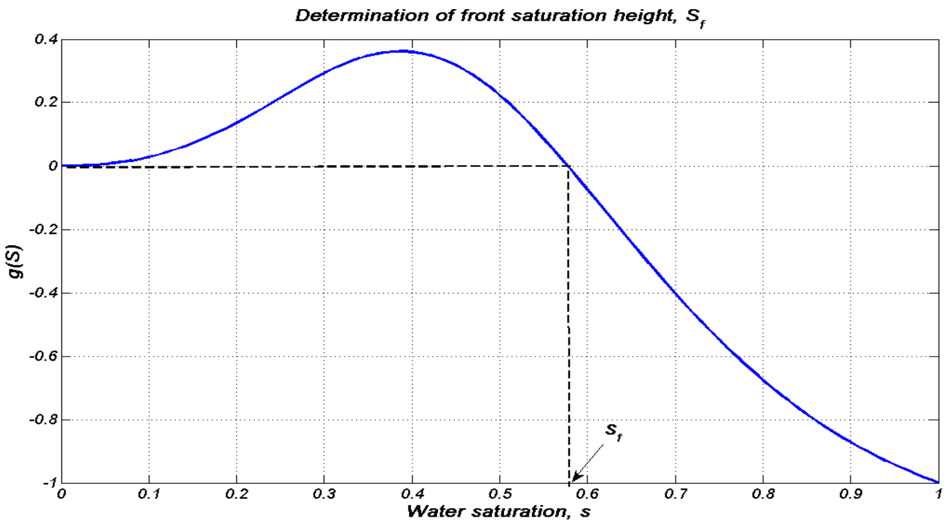


Figure 6: [A6] Determination of front saturation height

Where the function $g(S)$ intersect the horizontal line, $g(S) = 0$, gives the front saturation height S_f.

Remark: We can also determine S_f graphically by drawing the tangent line going from the origin to the inflexion point. The steepness of the tangent line is equal to the speed of which the saturation front is moving. The steepness is given by equation (2.2.1).

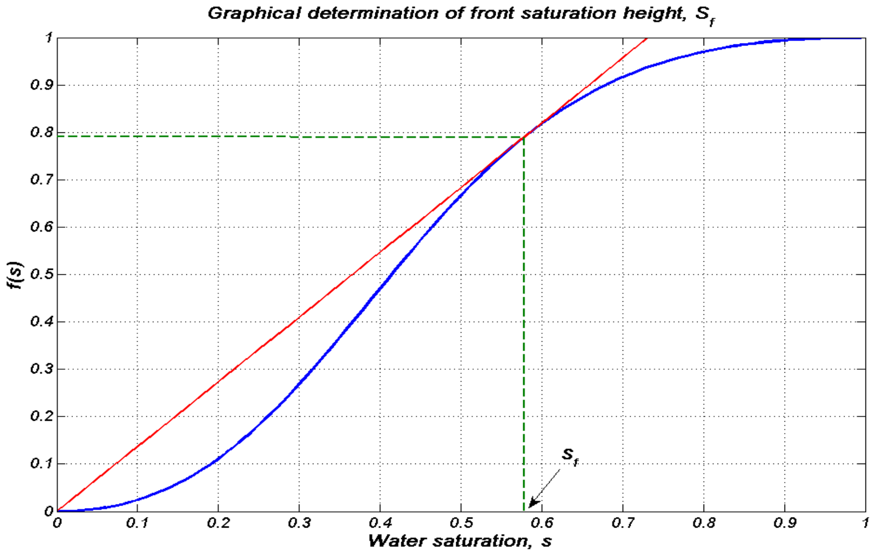


Figure 7: [A7] Graphical determination of front saturation

Obtaining the position of the front, x_f

[1] Since $f'(S_f)$ is the speed of the front, we find the position of the front by multiplying with elapsed time, T.

$$x_f = f'(S_f)T. \quad (2.2.3)$$

Saturation distribution behind the front

[1] We may also calculate saturation behind the front for a given position.

$$x = f'(S)T. \quad (2.2.4)$$

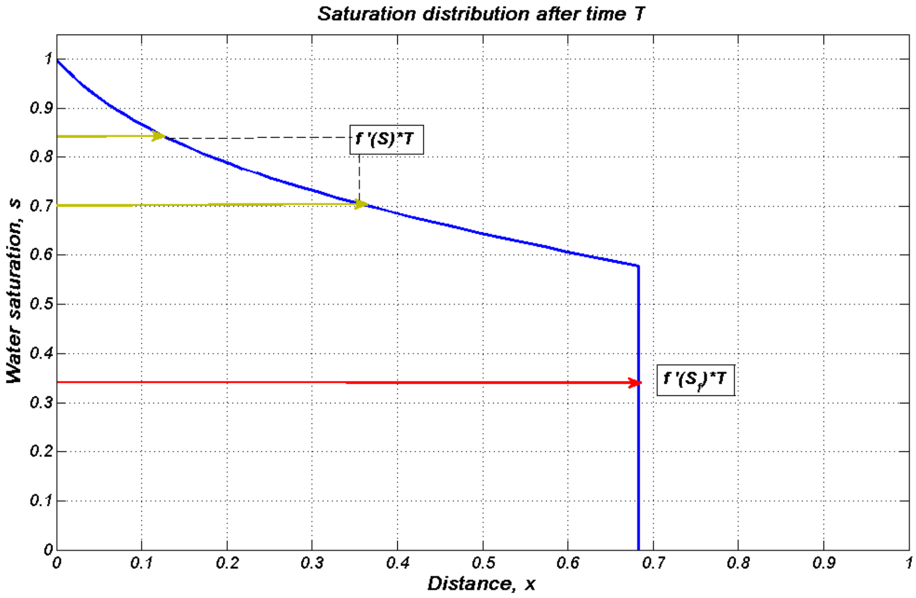


Figure 8: [A8] Saturation distribution after elapsed time, T

The solution

[2] Once the front saturation is obtained we can plot the solution at different values of T and see how the saturation distribution varies.

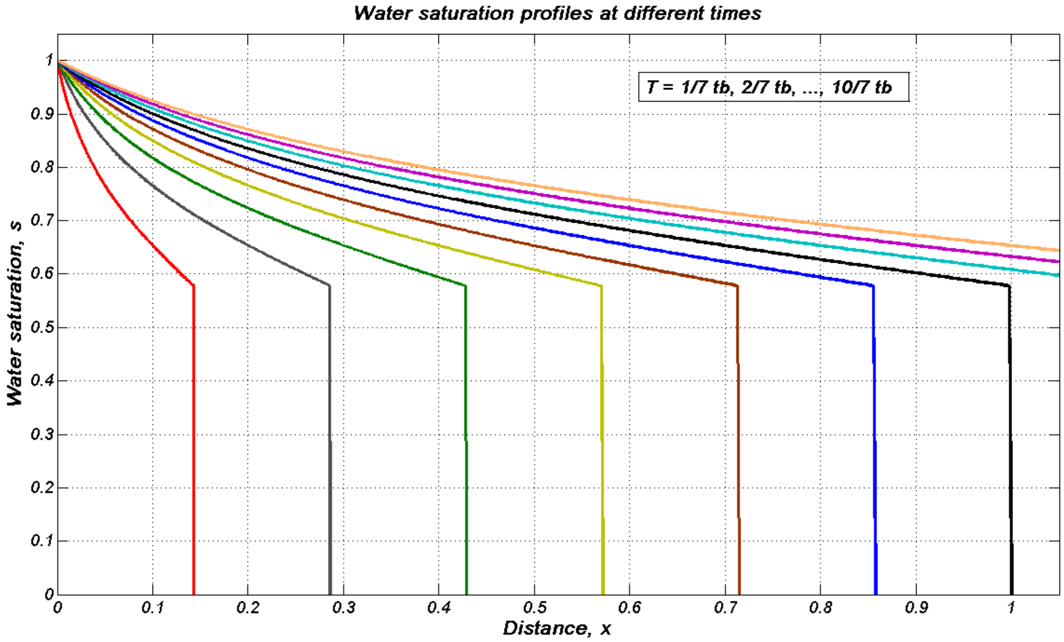


Figure 9: [A8] Saturation movement at different times

From (2.2.3) we know that the speed of which the saturations are moving is given by

$$v = f'(S) = \frac{x}{T}.$$

The point at which the water has reached the producing well is equal to 1, and since the highest speed of the saturation is at the front, we may introduce the breakthrough time as

$$T_{bt} = \frac{1}{v_f} = \frac{1}{f'(S_f)}. \tag{2.2.5}$$

Figure 9 shows the solution using the breakthrough time as reference. Note that the solution at $T = T_{bt}$ has its front at $x = 1$.

2.3 Numerical solution

[2] In order to study numerical solutions of the Buckley-Leverett model the relaxed scheme presented by Jin and Xin [22] in [2] is used. See references and appendix for a comprehensive description. This scheme has been tested for many different hyperbolic conversation laws and has proven itself to be accurate. A central in space-explicit in time discretization is used. A description of the discretization and the stability criterion will be given in the next model where capillary pressure is included.

The figure below shows a comparison of water saturations profiles obtained analytically and numerically with different number of nodes.

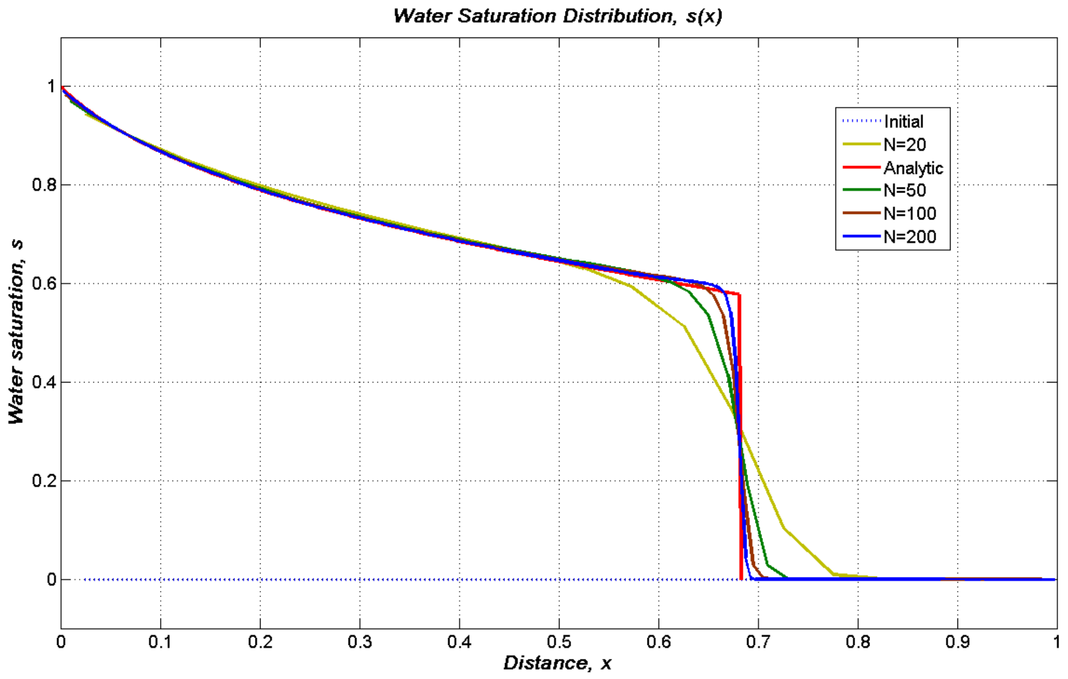


Figure 10: Comparison of analytical solution and numerical solutions with different number of nodes

From figure 10 we observe that the numerical solution is converging towards the analytical solution for increasing number of nodes. For nodes, $N \geq 100$, the numerical solution gets reasonable accurate. This explicit scheme has a fairly strict stability condition on the time step, especially when the capillary pressure term is included. $N = 100$ appear like a good choice to achieve acceptable accuracy and still maintain a low computational time. This value of N will be used in later analysis unless otherwise stated.

3 The Buckley-Leverett equation including capillary pressure

In the previous model only viscous forces were accounted for. We will now extend the model by adding another major element, namely capillary forces. This model will represent the physics of the water-oil displacement more accurately and will give a better understanding of the water injection process.

3.1 Derivation of the model

[1, 2] Recall from derivation of the ordinary Buckley-Leverett model that:

$$\frac{\partial}{\partial x}(q_w) + A\varphi\frac{\partial}{\partial t}(S_w) = 0, \qquad \frac{\partial}{\partial x}(q_o) + A\varphi\frac{\partial}{\partial t}(S_w) = 0. \tag{3.1.1}$$

Darcy's equations for oil and water respectively, are written as

$$q_w = -\frac{Kk_{rw}A}{\mu_w}\left(\frac{\partial p_w}{\partial x}\right), \tag{3.1.2}$$

$$q_o = -\frac{Kk_{ro}A}{\mu_o}\left(\frac{\partial p_o}{\partial x}\right). \tag{3.1.3}$$

Inserting Eq. (3.1.2) and (3.1.3) into Eq. (3.1.1) gives

$$\frac{\partial}{\partial x}\left(\frac{Kk_{rw}}{\mu_w}\frac{\partial p_w}{\partial x}\right) = \varphi\frac{\partial S_w}{\partial t}, \tag{3.1.4}$$

$$\frac{\partial}{\partial x}\left(\frac{Kk_{ro}}{\mu_o}\frac{\partial p_o}{\partial x}\right) = \varphi\frac{\partial S_o}{\partial t}. \tag{3.1.5}$$

We get the Darcy velocities, u_o and u_w, from Darcy laws as follows:

$$u_w = -\lambda_\mathrm{w}\left(\frac{\partial p_w}{\partial x}\right), \tag{3.1.6}$$

$$u_o = -\lambda_\mathrm{o}\left(\frac{\partial p_o}{\partial x}\right), \tag{3.1.7}$$

where

$$\lambda_w(S) = \frac{Kk_{rw}(S)}{\mu_w} \ , \qquad \lambda_o(S) = \frac{Kk_{ro}(S)}{\mu_o}.$$

By substituting Eq. (3.1.6) and (3.1.7) into Eq. (3.1.4) and (3.1.5) we obtain

$$\varphi\frac{\partial S_w}{\partial t} + \frac{\partial u_w}{\partial x} = 0, \qquad \varphi\frac{\partial S_o}{\partial t} + \frac{\partial u_o}{\partial x} = 0\ . \tag{3.1.8}$$

Now we introduce capillary pressure $P_c(S)$ defined as the difference between oil and water pressure

$$P_c(S) = p_o(S) - p_w(S) \tag{3.1.9}$$

The total velocity, u_T, is given by

$$\begin{aligned} u_T = u_w + u_o &= -\left(\lambda_w\frac{\partial p_w}{\partial x} + \lambda_o\frac{\partial p_o}{\partial x}\right) \\ &= -\left(\lambda_w\frac{\partial p_0}{\partial x} - \lambda_w\frac{\partial P_C}{\partial x} + \lambda_o\frac{\partial p_o}{\partial x}\right) \\ &= -\left(\lambda_T\frac{\partial p_o}{\partial x} - \lambda_w\frac{\partial P_c}{\partial x}\right) = \lambda_w\frac{\partial P_c}{\partial x} - \lambda_T\frac{\partial p_o}{\partial x}, \end{aligned} \tag{3.1.10}$$

where the total mobility

$$\lambda_T = \lambda_w + \lambda_o.$$

Using the constraint; $S_w + S_o = 1$ and by adding the two equations from (3.1.8) we obtain

$$\frac{\partial}{\partial x}u_T = \frac{\partial}{\partial x}\left(\lambda_w\frac{\partial P_c}{\partial x} - \lambda_T\frac{\partial p_o}{\partial x}\right) = 0, \tag{3.1.11}$$

which implies that the velocity, u_T, is independent of position, i.e. constant and is determined from boundary conditions.

From equation (3.1.10) it follows that

$$p_o = \int^x \frac{1}{\lambda_T}\left(\lambda_w\frac{\partial P_c}{\partial x} - u_T\right)dx, \tag{3.1.12}$$

which can be used to obtain p_o once water saturation S is known.

We can rewrite the continuity equation for water, Eq. (3.1.8),using $u_w = -\lambda_\mathrm{w}\left(\frac{\partial p_o}{\partial x} - \frac{\partial P_c}{\partial x}\right)$

$$\varphi\frac{\partial \mathrm{S_w}}{\partial \mathrm{t}} - \frac{\partial}{\partial x}\left(\lambda_\mathrm{w}\frac{\partial p_o}{\partial x}\right) + \frac{\partial}{\partial x}\left(\lambda_\mathrm{w}\frac{\partial P_c}{\partial x}\right) = 0. \tag{3.1.13}$$

From equation (3.1.10) it follows that

$$\frac{\partial p_o}{\partial x} = \frac{\lambda_w}{\lambda_T}\left(\frac{\partial P_c}{\partial x}\right) - \frac{u_T}{\lambda_T}. \tag{3.1.14}$$

Inserting the above equation into Eq. (3.1.13) we get

$$\varphi\frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x}\left(\lambda_w\left[\frac{u_T}{\lambda_T} - \frac{\lambda_w}{\lambda_T}\left(\frac{\partial P_c}{\partial x}\right)\right]\right) + \frac{\partial}{\partial x}\left(\lambda_w\frac{\partial P_c}{\partial x}\right) = 0. \tag{3.1.15}$$

Recall that the fractional flow function $f(S)$ are defined as

$$f(S) = \frac{\lambda_w(S)}{\lambda_t(S)} = \frac{k_{rw}(S)/\mu_w}{k_{rw}(S)/\mu_w + k_{ro}(S)/\mu_o}. \tag{3.1.16}$$

Using Eq. (3.1.16) in Eq. (3.1.15) gives that

$$\varphi\frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x}(u_T f(S)) - \frac{\partial}{\partial x}\left(\lambda_w f(S)\frac{\partial P_c}{\partial x} - \lambda_w\frac{\partial P_c}{\partial x}\right) = 0. \tag{3.1.17}$$

From Eq. (3.1.16) we see that

$$\lambda_w f(S) - \lambda_w = -\lambda_o f(S),$$

and by using the equation above, Eq. (3.1.17) can be written in the form

$$\varphi\frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x}(u_T f(S)) = -\frac{\partial}{\partial x}\left(\lambda_o f(S)\frac{\partial P_c}{\partial x}\right). \tag{3.1.18}$$

In view of Eq. (3.1.18) a model has been obtained of the form

$$\varphi \frac{\partial S}{\partial t} + u_T \frac{\partial f(S)}{\partial x} = \frac{\partial}{\partial x}\left(a(S)\frac{\partial P_c}{\partial x}\right), \tag{3.1.19}$$

where $f(S)$ is given by Eq. (3.1.16) and the diffusion coefficient $a(S)$ is given by

$$a(S) = -\lambda_o f(S) \tag{3.1.20}$$

The figure below shows the fractional flow function plotted together with the corresponding relative permeabilities and a plot of the diffusion coefficient for a given set of parameters ($nw = 3,\ no = 2,\ M = 0.5,\ \mu_o = 2$)

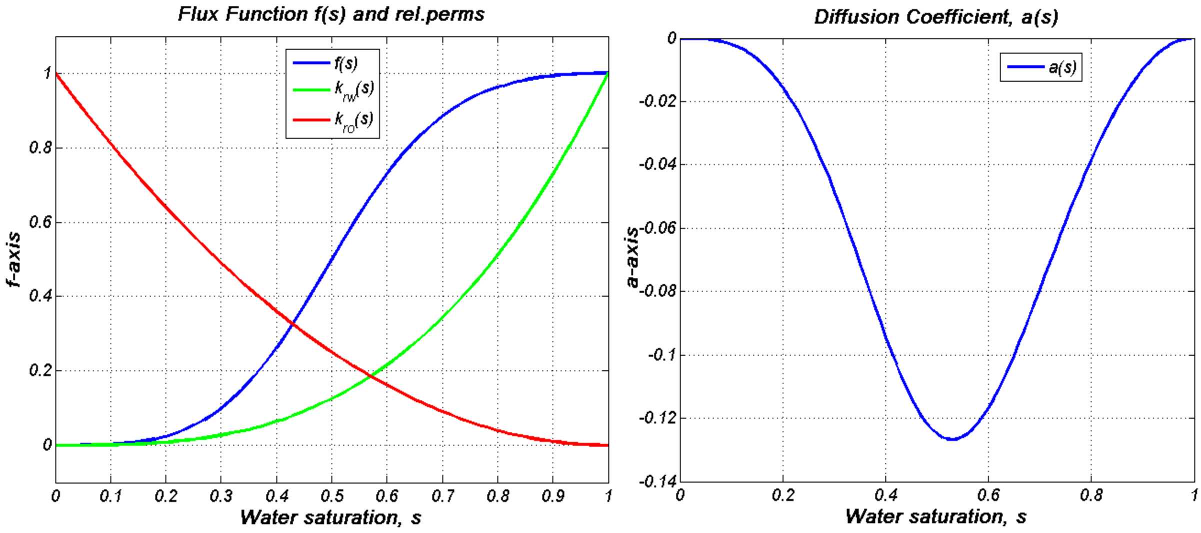


Figure 11: [A1,A2] Flux function, f(s), with corresponding relative permeabilities and [A9] diffusion coefficient, a(s). nw=3, no=2, M=0.5

3.2 Dimensionless variables

[1,2] Recall that

$$\frac{\partial S}{\partial t_D} + \frac{\partial f(S)}{\partial x_D} = 0$$

for the ordinary Buckley-Leverett model where the capillary pressure effects is neglected, i.e. $P_c = 0$, and where $x_D = \frac{x}{L}, and\ t_D = \frac{u}{\varphi L} t$.

We now introduce reference capillary pressure $P_{c,r}$, and reference viscosity μ_r, and define dimensionless capillary pressure $P_{c,D}$ and dimensionless oil viscosity μ_{oD} as

$$P_{c,D} = \frac{P_c}{P_{c,r}}, \qquad \mu_{oD} = \frac{\mu_o}{\mu_r},$$

Then the dimensionless Buckley-Leverett equation including capillary pressure can be written as (without the subscript D)

$$\frac{\partial \mathrm{S}}{\partial \mathrm{t}} + \frac{\partial f(S)}{\partial x} = \varepsilon \frac{\partial}{\partial x}\left(a(S)\frac{\partial P_c}{\partial x}\right), \qquad \varepsilon = \frac{P_{c,r}}{L u_T \mu_r}, \tag{3.2.1}$$

where

$$a(s) = -\frac{k_{ro}}{\mu_{oD}} f(S). \tag{3.2.2}$$

The dimensionless, characteristic number, ε, relates the viscous and capillary forces and depends on reference capillary pressure and viscosity, length of porous media and total injection flow velocity. This number determines which effect the capillary forces will have on the displacement process.

The function, $a(s)$, is called the diffusion coefficient and depends on relative permeability of oil, dimensionless viscosity of oil and the fractional flow function.

3.3 Discretization of the model

[1,2] We are not able to compute analytical solutions of equation (3.2.1). So in order to solve and obtain solutions for this model we need discretize the model and make use of a numerical scheme.

First we discretize the problem's domain. This is done by dividing the domain into a uniform grid in space and time.

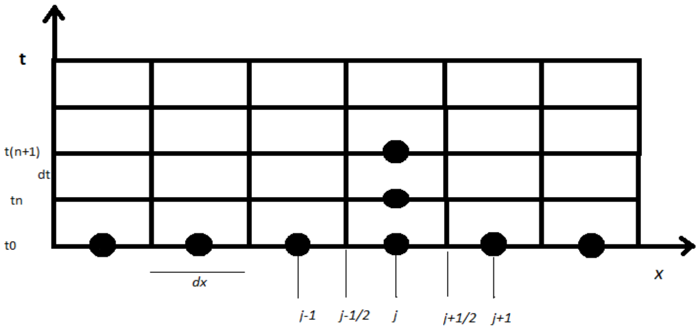


Figure 12: Discretization in space and time

The grid is divided into K special cells with length $\Delta x = 1/K$. We associate $x_{j+1/2}$ with the cell interface for $j = 0, \dots, K$ and x_j with the cell center for $j = 1, \dots, K$. Likewise for N time steps of length $\Delta t = 1/N$ where $t^n = n\Delta t$ for $n = 1, \dots, N$.

We may write the model

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial x} = \varepsilon \frac{\partial}{\partial x}\left(a(S)\frac{\partial P_c}{\partial x}\right),$$

in a more general form

$$S_t + f(S)_x = \varepsilon(a(S)P_c(S)_x)_x, \qquad S(x,0) = S_0(x). \tag{3.3.1}$$

A discretization in the interior domain of the equation above is given by

$$\frac{S_j^{n+1} - S_j^n}{\Delta t} + \frac{1}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right) = \frac{\varepsilon}{\Delta x}\left(a_{j+1/2}^n \frac{P_{c\,j+1}^{\,n} - P_{c\,j}^{\,n}}{\Delta x} - a_{j-1/2}^n \frac{P_{c\,j}^{\,n} - P_{c\,j-1}^{\,n}}{\Delta x}\right), \tag{3.3.2}$$

for cell $j = 2, \dots, K-1$.

And a solution for the unknown S_j^{n+1} can be written as

$$S_j^{n+1} = S_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right) + \frac{\varepsilon\Delta t}{\Delta x}\left(a_{j+1/2}^n \frac{{P_c}_{j+1}^n - {P_c}_j^n}{\Delta x} - a_{j-1/2}^n \frac{{P_c}_j^n - {P_c}_{j-1}^n}{\Delta x}\right) \tag{3.3.3}$$

where

$$a_{j\pm1/2}^n = \frac{1}{2}\left(a_{j\pm1}^n + a_j^n\right), \tag{3.3.4}$$

and the function F takes the form

$$F_{j+1/2}^n = \frac{1}{2}\left(F\left(S_j^n\right) + F\left(S_{j+1}^n\right)\right) - \frac{1}{2}D\left(S_{j+1}^n - S_j^n\right), \tag{3.3.5}$$

where D is an appropriate value such that $D \geq \max[f'(S)]$ and the function F is called the numerical flux and is an approximation of $f(S)$.

3.3.1 Discretization at the boundary

For cell $j = 1$ the following discretization is used

$$\frac{S_1^{n+1} - S_1^n}{\Delta t} + \frac{1}{\Delta x}\left(F_{1+1/2}^n - F_{1/2}^n\right) = \frac{\varepsilon}{\Delta x}\left(a_{1+1/2}^n \frac{{P_c}_2^n - {P_c}_1^n}{\Delta x} - a_{1/2}^n \frac{{P_c}_1^n - {P_c}_0^n}{\Delta x/2}\right), \tag{3.3.6}$$

where

$${P_c}_0^n = P_c(S=1)^n, \quad a_{1/2}^n = a_1^n, \quad F_{1/2}^n = F^n(S=1). \tag{3.3.7}$$

For cell $j = K$

$$\frac{S_K^{n+1} - S_K^n}{\Delta t} + \frac{1}{\Delta x}\left(F_{K+1/2}^n - F_{K-1/2}^n\right) = \frac{\varepsilon}{\Delta x}\left(a_{K+1/2}^n \frac{{P_c}_{K+1}^n - {P_c}_K^n}{\Delta x/2} - a_{K-1/2}^n \frac{{P_c}_K^n - {P_c}_{K-1}^n}{\Delta x}\right), \tag{3.3.8}$$

where

$${P_c}_{K+1}^n = P_c(S_K)^n, \quad a_{K+1/2}^n = a^n(S_K), \quad F_{K+1/2}^n = F^n(S_K). \tag{3.3.9}$$

3.4 Stability

[1,2] In order to have stability when using explicit numerical schemes, we are required to apply the necessary condition known as the Courant-Friedrich-Lewy, CFL-condition.

CFL stability conditions

$$\frac{\Delta t}{\Delta x}\max[f'(S)] \leq 1 \qquad and \qquad \frac{\Delta t}{\Delta x^2}\max[a(S) * P_c'(S)] * \varepsilon \leq \frac{1}{2}, \qquad (3.4.1)$$

where $\max[f'(S)]$ is an estimate for the maximum value of the derivative to the fractional flow function $f(S)$ and where $\max[a(S) * P_c{}'(S)]$ is the product of the absolute maximum value of the diffusion coefficient $a(S)$ multiplied with the absolute maximum value of the derivative to the capillary pressure function $P_c(S)$. Δt and Δx represents the time steps and space steps respectively.

Both CFL conditions have to be satisfied to ensure that the numerical scheme is stable. In most cases the strictest condition will be the second order condition $\frac{\Delta t}{\Delta x^2}\max[a(S) * P_c'(S)] * \varepsilon \leq \frac{1}{2}$.

3.5 Capillary pressure correlations

3.5.1 Capillary Pressure Correlation for Mixed-Wet Reservoirs

$$P_{ci} = C^* \left[\frac{c_{wi}}{\left(\frac{S_w - S_{wR}}{1 - S_{wR}} \right)^{a_{wi}}} + \frac{c_{oi}}{\left(\frac{S_o - S_{oR}}{1 - S_{oR}} \right)^{a_{oi}}} \right] \qquad (3.5.1)$$

[3] An imbibition curve from $S_{wR}$to S_{oR} is obtained by the equation above, where a_{wi}, a_{oi}, c_{wi}, c_{oi} and C^* are constants that must be specified. The constraints on the constants are that a_{wi}, a_{oi}, c_{wi} are positive numbers and c_{oi} is a negative number. S_{wR} and S_{0R} are residual saturations (in this thesis $S_{wR} = S_{0R} = 0$).

If we set $a_{wi} = a_{oi} = c_{wi} = |c_{oi}| = 1$, equation (3.5.1) reduces to:

$$P_{ci} = C^* \left[\frac{1}{(S_w)} - \frac{1}{(S_o)} \right].$$

Unless otherwise stated, $a_{wi} = 0.5,\ a_{oi} = 0.5,\ c_{wi} = 1,\ c_{oi} = -1,\ C^* = 0.1$.

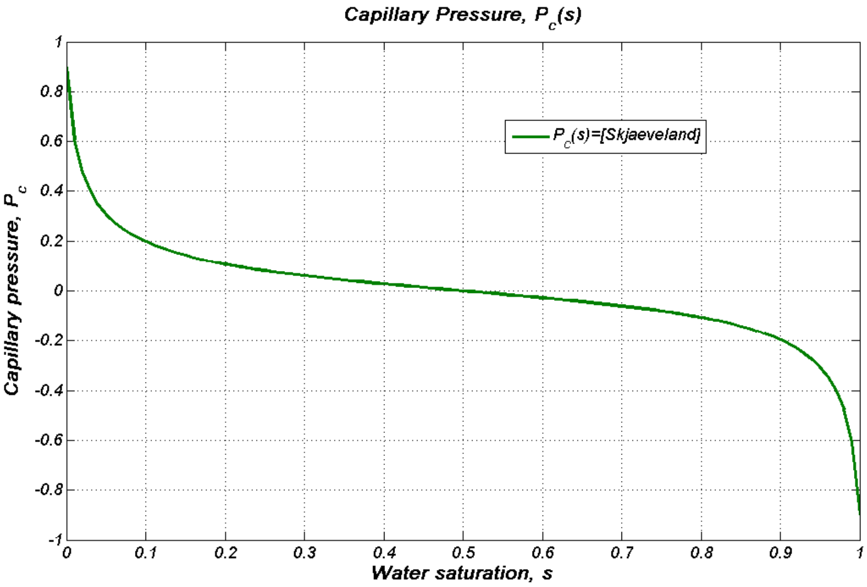


Figure 13: [A10] Secondary imbibition capillary pressure curve using Skjaeveland correlation with the given parameters

3.5.2 A New Versatile Capillary Pressure Correlation

[4] The new correlation [4] has a characteristic capillary pressure shape, but instead of having infinite values at the residuals which can cause problems for reservoir modeling, it has finite values that must be specified. "By using a finite correlation for the capillary pressure one avoids the mathematical problems of an infinite correlation without a significant sacrifice in accuracy when modeling the capillary transition zone. Despite the addition of parameters, the LET correlation remains easily accessible and applicable for full field reservoir simulations and engineering".

The spontaneous water saturation, which is the water saturation where the capillary pressure is zero, is determined by an imbibition test in the laboratory and is used as a defined parameter in this correlation.

The structure of the mathematical LET function F(x) is:

$$F(S_{wn}) = \frac{{S_{wn}}^{L}}{{S_{wn}}^{L} + E(1 - S_{wn})^{T}}, \tag{3.5.2}$$

where the parameter L describes the lower part of the curve and the parameter T describes the upper part in a similar way. The parameter E specifies the position of the slope of the curve. The normalized water saturation is defined as $S_{wn} = \frac{S_w - S_{wi}}{1 - S_{wi} - S_{orw}}$.

Since the secondary imbibition curve for an oil-water system consist of a positive (spontaneous) part and a negative (forced) part, two LET functions are combined to address this process. The versatile LET correlation for secondary imbibition capillary pressure can be written as

$$P_{cow}^{i} = P_{cow}^{si} * F_{cow}^{si} + P_{cow}^{fi} * F_{cow}^{fi}, \tag{3.5.3}$$

where P_{cow}^{si} and P_{cow}^{fi} are the preset maximum and minimum values of the spontaneous and forced capillary pressure respectively. The first term in Eq. (3.5.3) represent the positive spontaneous part of the imbibition curve, where the LET function is given by

$$F_{cow}^{si} = \frac{(1 - S_{wn})^{L_{ow}^{si}}}{(1 - S_{wn})^{L_{ow}^{si}} + E_{ow}^{si} {S_{wn}}^{T_{ow}^{si}}}, \tag{3.5.4}$$

and the second term in Eq. (3.5.3) represents the negative forced part of the imbibition curve, where the LET function is

$$F_{cow}^{fi} = \frac{{S_{wn}}^{L_{ow}^{fi}}}{{S_{wn}}^{L_{ow}^{fi}} + E_{ow}^{fi}(1 - S_{wn})^{T_{ow}^{fi}}}. \tag{3.5.5}$$

By substituting equation (3.5.4) and (3.5.5) into equation (3.5.3) we can write the secondary imbibition capillary pressure LET correlation as

$$P_{cow}^{i} = \frac{P_{cow}^{si}(1 - S_{wn})^{L_{ow}^{si}}}{(1 - S_{wn})^{L_{ow}^{si}} + E_{ow}^{si}{S_{wn}}^{T_{ow}^{si}}} + \frac{P_{cow}^{fi}{S_{wn}}^{L_{ow}^{fi}}}{{S_{wn}}^{L_{ow}^{fi}} + E_{ow}^{fi}(1 - S_{wn})^{T_{ow}^{fi}}}. \tag{3.5.6}$$

By rearranging the equation above, the E-parameter in the spontaneous imbibition term can be calculated by

$$E_{ow}^{si} = -\left[\frac{P_{cow}^{si}\left({S_{wsin}}^{L_{ow}^{fi}} + E_{ow}^{fi}(1 - S_{wsin})^{T_{ow}^{fi}}\right)}{P_{cow}^{fi}{S_{wsin}}^{L_{ow}^{fi}}} + 1\right] * \frac{(1 - S_{wsin})^{L_{ow}^{si}}}{{S_{wsin}}^{T_{ow}^{si}}}, \tag{3.5.7}$$

where the spontaneous water saturation S_{wsin} normally is defined from imbibition tests in the laboratory.

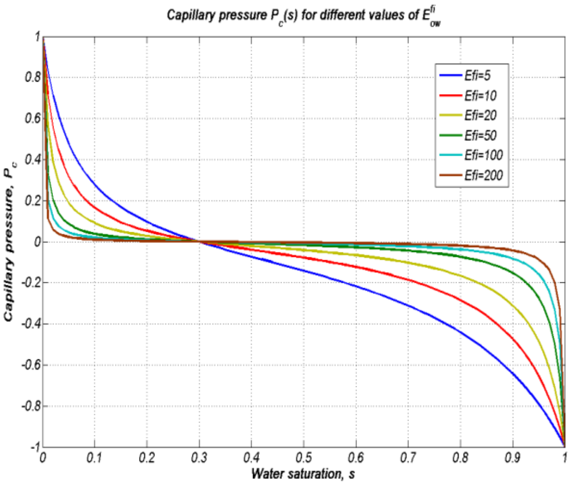


Figure 14: [A10] Variation in the forced E-parameter, E_{ow}^{fi}

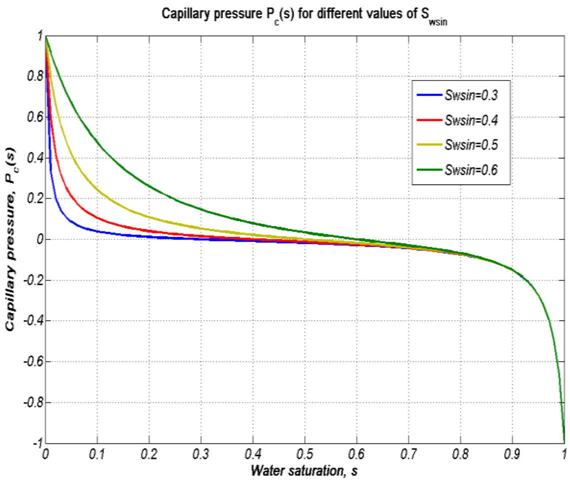


Figure 15: [A10] Variation of the spontaneous saturation, S_{wsin}

Remark: This is a simplified version of the LET correlation for the secondary imbibition capillary pressure. A constant term called the *threshold pressure*, which is used to raise or lower the slow varying part of the curve, has been neglected in equations (3.5.3, 3.5.6 and 3.5.7).

3.6 Water saturation profiles - Effects of including capillary pressure

Figure 16 shows a comparison of water saturation distributions, obtained by solving the modified Buckley-Leverett model given in Eq. (3.2.1) numerically, with different capillary pressure correlations. It also contains an analytical (exact) solution and a numerical solution obtained from the first BL model given by Eq. (2.1.14), where the capillary pressure is assumed to be zero.

Figure 17 shows the associated capillary pressure curves. The yellow curve is not described in the paper, but is simply given by [7] $P_c = -Bln(S)$ where B is a constant that must be specified. In this paper $B = 0.217$.

Unless otherwise stated, the parameters from the table are used in this chapter.

Parameters	***[Lomeland]***	***[Skjaeveland]***
$Nodes = 100$	$P_{cow}^{si} = 1$	$c_{wi} = 1$
$M = 0.5$	$P_{cow}^{fi} = -1$	$c_{oi} = -1$
$Nw = 3$	$E_{ow}^{fi} = 50$	$a_{wi} = 0.5$
$No = 2$	$L_{ow}^{si} = 2$	$a_{oi} = 0.5$
$T = 0.5$	$L_{ow}^{fi} = 1$	$C^* = 0.1$
$\varepsilon = 0.833$	$T_{ow}^{si} = T_{ow}^{fi} = 1$	
	$S_{wsin} = 0.5$	

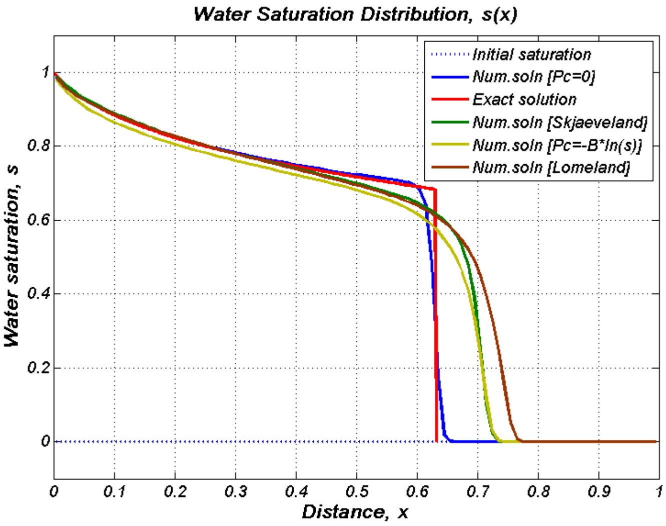


Figure 16: Comparison of water saturation distributions

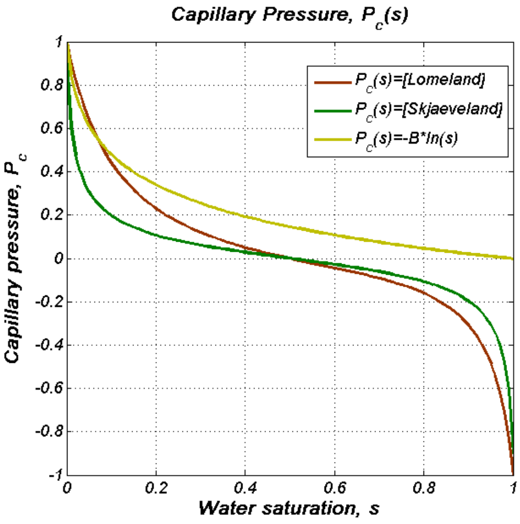


Figure 17: Corresponding capillary pressure curves, $E_{ow}^{fi} = 20$

It is observed from Figure 16 that the saturation profile produced from the numerical solution, where capillary pressure is neglected, seems to have a relative sharp front. Once we include a capillary pressure term we see that the displacement front tends to smooth/smear out. This smearing effect depends strongly on the parameter ε, the greater the value of ε the more the front gets smeared out.

To further demonstrate the effects of including capillary pressure we use the LET-correlation with varying values of the empirical parameter E_{ow}^{fi} and the spontaneous water saturation S_{wsin}. Figure 18 and 19 illustrates the water saturation behavior and associated capillary pressure curves, by varying E_{ow}^{fi} and S_{wsin} respectively.

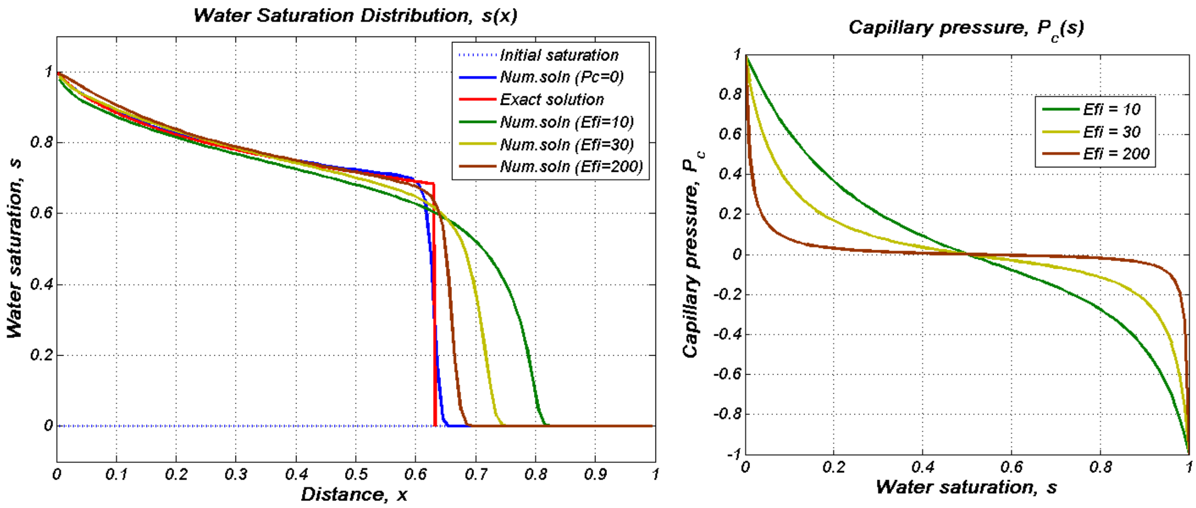


Figure 18: Comparison of water saturation distributions with corresponding capillary pressure curves (varying E_{ow}^{fi})

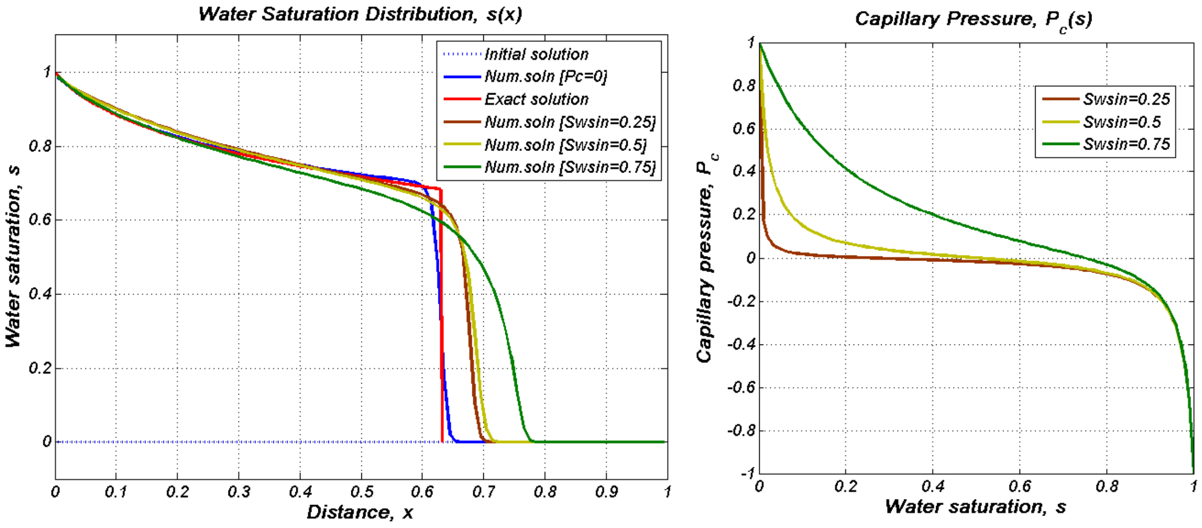


Figure 19: Comparison of water saturation distributions with corresponding capillary pressure curves (varying S_{wsin}), $L_{ow}^{si} = 1$

From figure 18 and 19 we can observe that the shape of the capillary pressure curve is important for the associated saturation profile. For decreasing values of E_{ow}^{fi} and/or increasing values of S_{wsin}, the capillary pressure curve becomes more “elongated”, thus less variation in the derivative of $P_c(S)$. This tends to smooth the displacement front and results in a lower average saturation, but a longer travelled distance. The increase in travelled distance is relatively greater than the decrease in average water saturation and as a result, oil recovery was observed to increase before breakthrough (figure 20, left). Note that an increase in oil recovery (before breakthrough) was observed for all saturation profiles that included capillary pressure. Due to higher sweep efficiency, the solution with neglected capillary effects has the highest recovery at breakthrough. After breakthrough, as more water is injected, the recoveries are observed to behave similarly (illustrated in figure 20 (right)).

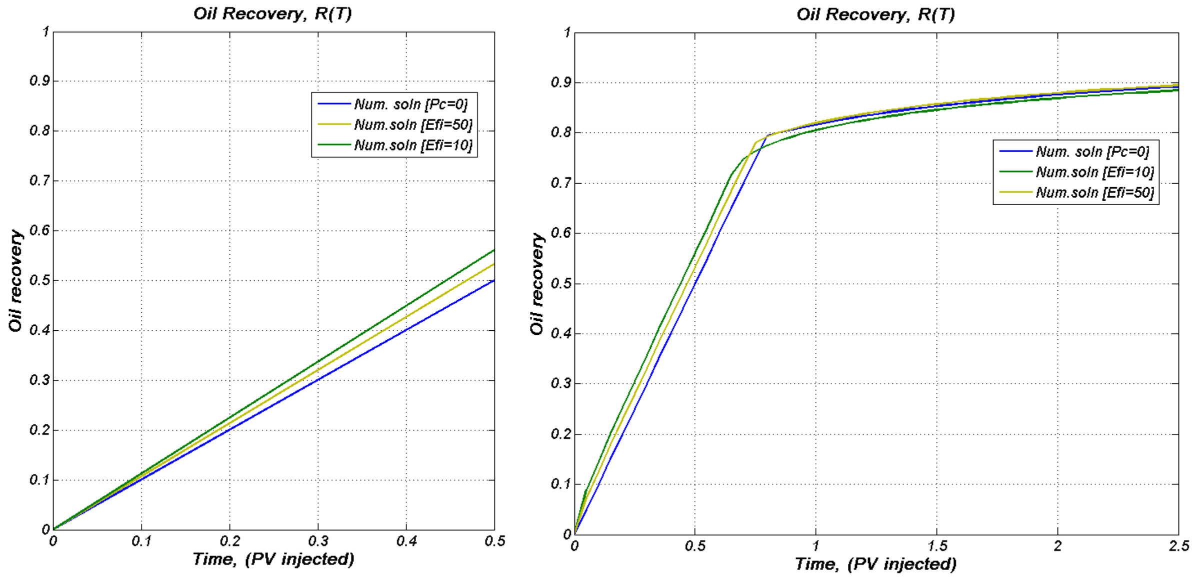


Figure 20: Oil recovery before breakthrough, $S_{wsin} = 0.5$, $E_{ow}^{fi} = [50, 10]$ (left) and after breakthrough (right) with and without Pc-effect

[6] The importance of the capillary forces in the displacement process are mainly determined by the dimensionless number, $\varepsilon = \frac{P_{c,r}}{L u_T \mu_r}$. A reduction in injection rate, porous media length and/or reference viscosity i.e. reduced oil viscosity, will result in increased capillary contribution to displacement process. Figure 21 illustrates this for increasing values of ε. The LET-correlation has been used to express the capillary pressure curve.

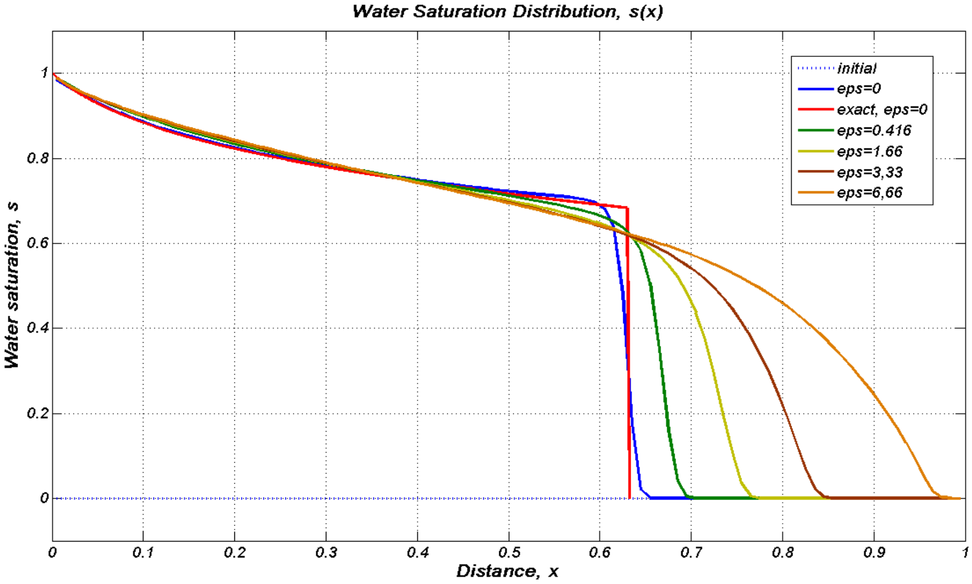


Figure 21: Comparison of saturation profiles with increasing capillary effect

For high numbers of ε, where capillary effects are dominant, there is less variation in the saturation gradients, resulting in more linearized saturation profiles. Figure 22 illustrates this at different dimensionless times, T. This result is in some agreement with results presented in [6] and references therein.

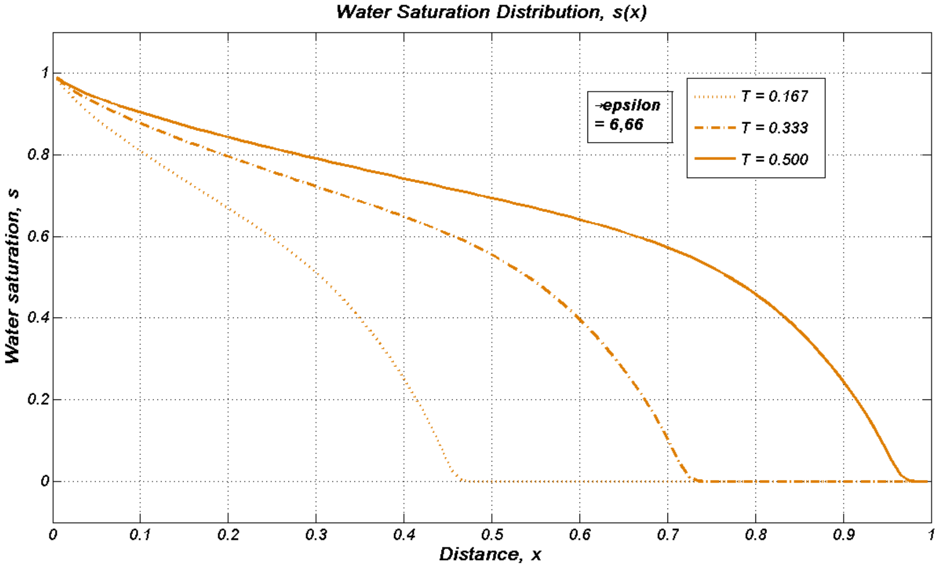


Figure 22: Saturation profiles at different times with dominant capillary effects

3.7 Capillary end effects

[8] In this section we also include capillary end effects in the numerical simulations. Capillary end effects are a phenomenon that arises mainly in experiments performed on small core samples, where the length of the “reservoir” is small. If the capillary pressure at the outlet of the core is set to be zero, the water saturation at the outlet takes the value that satisfy, $P_c(S) = 0$, i.e. the spontaneous water saturation S_{wsin}. Figure 23 illustrate this effect for different spontaneous water saturations.

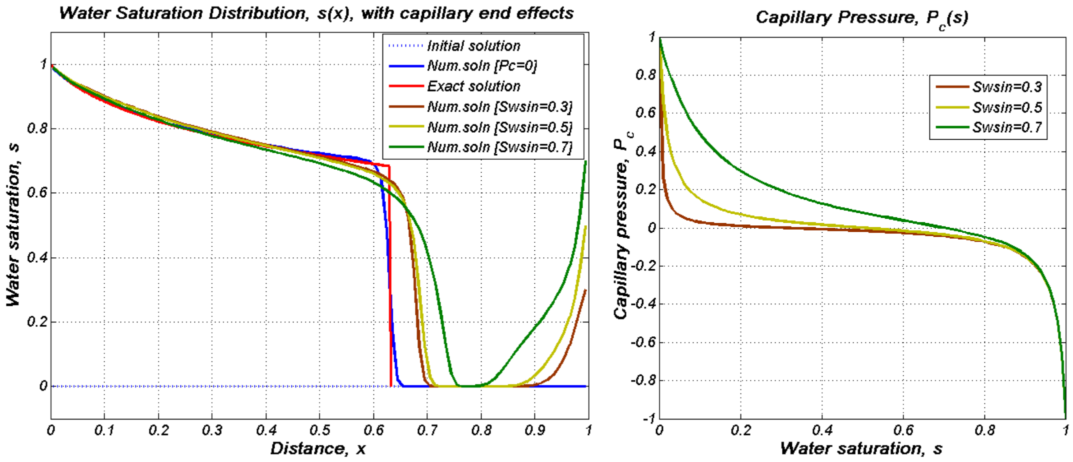


Figure 23: Comparison of saturation profiles with capillary end effects for different spontaneous water saturations

In figure 24 we observe that the recovery before breakthrough is similar to recovery illustrated in figure 20, but after breakthrough the gap between the curves tends to increase a bit which is the opposite of the curves in figure 20. “This is a result of capillary end effects where the porous media try to retain some water at the end of the media in an attempt to maintain equilibrium across the outlet where the capillary pressure is set to be zero”.

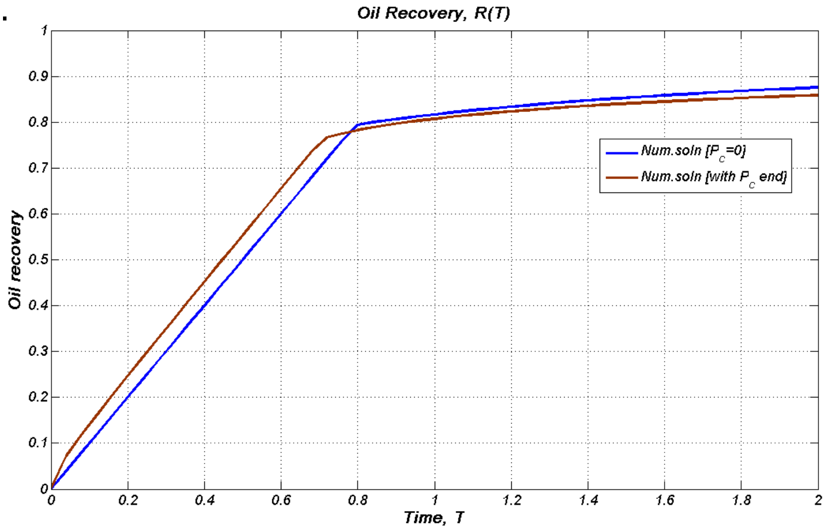


Figure 24: Recovery comparison with and without capillary end effects, $S_{wsin} = 0.3, L_{ow}^{si} = 1, E_{ow}^{fi} = 20$

4. A new model with modified Buckley-Leverett theory

[5] A new Buckley-Leverett model was developed by Terez and Firoozabadi [5] for studying water injection in water-wet fractured porous media. This model includes a time dependent transfer term that takes into account fluid exchange rate between matrix and fracture. This modified model covers both cocurrent and countercurrent imbibition and is computationally very efficient.

They performed experiments on Berea sandstone, Kansas outcrop chalk and Austin chalk. The experimental results from Berea sandstone and Kansas chalk, (see paper), were compared and matched with implicit fine grid simulations by Eclipse simulator and the modified Buckley-Leverett model, BLM. The results showed overall a good match between experimental data and both simulations, but the computationally time were significantly less for the BLM.

The BLM is used to explore the imbibition effect (both cocurrent and countercurrent) in a single block and to predict water injection in different types of multiblock systems.

4.1 Derivation of the modified Buckley-Leverett Model (BLM)

[5] Recall from the previous model that

$$\varphi \frac{\partial \mathrm{S}}{\partial \mathrm{t}} + u_T \frac{\partial f(S)}{\partial x} = \frac{\partial}{\partial x}\left(a(S)\frac{\partial P_c}{\partial x}\right), \tag{4.1.1}$$

In this model fracture capillary pressure is assumed to be zero, thus the right hand side of Eq. (4.1.1) vanishes. A transfer term, q_w, that represents the fluid exchange rate between matrix and fracture is included and parameters considering fracture properties are added.

The modified model can be written as

$$\varphi_e^f \frac{\partial \mathrm{S}_w^f}{\partial \mathrm{t}} + u_T \frac{\partial f(S_w^f)}{\partial x} = -q_w, \tag{4.1.2}$$

where S_w^f is the fracture water saturation, and φ_e^f is the effective fracture porosity. The effective fracture porosity is defined as the fracture volume divided by the total bulk volume of matrix and fracture.

The fracture fractional flow function, $f\left(S_w^f\right)$, is defined as

$$f\left(S_w^f\right) = \frac{\lambda_w^f\left(S_w^f\right)}{\lambda_t^f\left(S_w^f\right)} = \frac{k_{rw}^f\left(S_w^f\right)\big/\mu_w}{k_{rw}^f\left(S_w^f\right)\big/\mu_w + k_{ro}^f\left(S_w^f\right)\big/\mu_o} = \frac{1}{1+\frac{\mu_w}{\mu_o}\frac{k_{ro}^f\left(S_w^f\right)}{k_{rw}^f\left(S_w^f\right)}}, \tag{4.1.3}$$

where $k_{rw}^f\left(S_w^f\right)$ is the water relative permeability in the fracture and $k_{ro}^f\left(S_w^f\right)$ is the oil relative permeability. These are again given by:

$$k_{ro}^f\left(S_w^f\right) = KRO(1-S)^n, \tag{4.1.4}$$

$$k_{rw}^f\left(S_w^f\right) = KRW(S^n), \tag{4.1.5}$$

where KRO and KRW are the endpoint values. In this chapter the Corey exponent $n = 2$.

The expression for the transfer term $q_w(x,t)$ with variable fracture saturation can be written using Duhamel's principle:

$$\begin{aligned} q_w(x,t) = {} & \left[S_w^f(x,t_0)\right]^m R_1\sigma_1 e^{-\sigma_1(t-t_0)} \\ & + \left(\left[S_w^f(x,t_1)\right]^m - \left[S_w^f(x,t_0)\right]^m\right) R_1\sigma_1 e^{-\sigma_1(t-t_1)} \\ & + \cdots \left(\left[S_w^f(x,t_{n-1})\right]^m - \left[S_w^f(x,t_{n-2})\right]^m\right) R_1\sigma_1 e^{-\sigma_1(t-t_{n-1})} \\ & + \left[S_w^f(x,t_0)\right]^m R_2\sigma_2 e^{-\sigma_2(t-t_0)} \\ & + \left(\left[S_w^f(x,t_1)\right]^m - \left[S_w^f(x,t_0)\right]^m\right) R_2\sigma_2 e^{-\sigma_2(t-t_1)} \\ & + \cdots \left(\left[S_w^f(x,t_{n-1})\right]^m - \left[S_w^f(x,t_{n-2})\right]^m\right) R_2\sigma_2 e^{-\sigma_2(t-t_{n-1})}. \end{aligned} \tag{4.1.6}$$

"The time period from t_0 to t_n is divided into n parts. Letting $n \to \infty$ and using the definition of an integral and integrating by parts, one obtains"

$$\begin{aligned} q_w = {} & R_1\sigma_1\left(\left[S_w^f(x,t)\right]^m - \sigma_1\int_0^t \left[S_w^f(x,\tau)\right]^m e^{-\sigma_1(t-\tau)}d\tau\right) \\ & + R_2\sigma_2\left(\left[S_w^f(x,t)\right]^m - \sigma_2\int_0^t \left[S_w^f(x,\tau)\right]^m e^{-\sigma_2(t-\tau)}d\tau\right). \end{aligned} \tag{4.1.7}$$

The transfer term consists of a two terms; the first one account for the cocurrent imbibition and the second one for the countercurrent imbibition. R_1, R_2, σ_1 and σ_2 are fixed parameters for a given set of matrix capillary pressure and matrix Corey relative parmeabilities. Only σ_1 varies with the injection rate.

The parameters σ_1 and σ_2 represents the imbibition rate of the matrix blocks. These parameters can be estimated using fine grid simulation of a single field-size block.

Calculations performed by Terez and Firoozabadi suggests that the saturation exponent, m, should take the value 0.5. See reference [5] for further details.

4.2 Dimensionless form of BLM

Like the two previous models, it is convenient to make the BLM dimensionless.

Recall that

$$\frac{\partial S}{\partial t_D} + \frac{\partial f(S)}{\partial x_D} = 0,$$

for the ordinary Buckley-Leverett model where the capillary pressure effects is neglected, i.e. $P_c = 0$, and where $x_D = \frac{x}{L}, and\ t_D = \frac{u}{\varphi L} t$.

Since the transfer term consists of two terms we can divide it into $q_{w,1}$ and $q_{w,2}$ which represents the cocurrent imbibition rate and the countercurrent imbibition rate respectively.

We use the expressions above in Eq. (4.1.2), and introduce two new dimensionless numbers, ε_1 and ε_2, that depends on the length of porous media, total injection flow velocity and the transfer term parameters R_1, R_2, σ_1 and σ_2.

The dimensionless BLM equation can be written as (without the subscript D)

$$\frac{\partial S_w^f}{\partial t} + \frac{\partial f(S_w^f)}{\partial x} = -\left(\varepsilon_1 q_{w,1} + \varepsilon_2 q_{w,2}\right), \qquad \varepsilon_1 = \frac{L(R_1\sigma_1)}{u_T}, \qquad \varepsilon_2 = \frac{L(R_2\sigma_2)}{u_T}. \tag{4.2.1}$$

4.3 Discretization of BLM

Recall that a discretization of the left hand side of Eq. (4.2.1) can be written as

$$\frac{\partial \mathrm{S}}{\partial \mathrm{t}}+\frac{\partial f(S)}{\partial x}\approx\frac{S_j^{n+1}-S_j^n}{\Delta t}+\frac{1}{\Delta x}\left(F_{j+1/2}^n-F_{j-1/2}^n\right), \tag{4.3.1}$$

where the function F takes the form

$$F_{j+1/2}^n=\frac{1}{2}\Big(F\big(S_j^n\big)+F\big(S_{j+1}^n\big)\Big)-\frac{1}{2}D\big(S_{j+1}^n-S_j^n\big), \tag{4.3.2}$$

and where D is an appropriate value such that $D\geq\max[f'(S)]$ and the function F is called the numerical flux and is an approximation of $f(S)$.

In the above and subsequent equations, the fracture water saturation subscripts f and w are dropped for the sake of brevity.

A discrete version of the transfer terms can be written in the form

$$q_{w,1}\big(S_j^n\big)=\left(\big(S_j^n\big)^m-\sigma_1\sum_{k=1}^{n}\Delta t*\big(S_j^k\big)^m*e^{-\sigma_1(t^n-k\Delta t)}\right) \tag{4.3.3}$$

$$q_{w,2}\big(S_j^n\big)=\left(\big(S_j^n\big)^m-\sigma_2\sum_{k=1}^{n}\Delta t*\big(S_j^k\big)^m*e^{-\sigma_2(t^n-k\Delta t)}\right). \tag{4.3.4}$$

By substituting Eqs. (4.3.1)-(4.3.4) into Eq. (4.2.1) we obtain a discretization of the BLM

$$\begin{aligned}\frac{S_j^{n+1}-S_j^n}{\Delta t}&+\frac{1}{\Delta x}\left(F_{j+1/2}^n-F_{j-1/2}^n\right)\\&=-\left[\varepsilon_1\left(\big(S_j^n\big)^m-\sigma_1\sum_{k=1}^{n}\Delta t*\big(S_j^k\big)^m*e^{-\sigma_1(t^n-k\Delta t)}\right)\right.\\&\quad\left.+\varepsilon_2\left(\big(S_j^n\big)^m-\sigma_2\sum_{k=1}^{n}\Delta t*\big(S_j^k\big)^m*e^{-\sigma_2(t^n-k\Delta t)}\right)\right],\end{aligned} \tag{4.3.5}$$

and a solution for the unknown S_j^{n+1} can be written as

$$S_j^{n+1} = S_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right) - \Delta t\left[\varepsilon_1 q_{w,1}\left(S_j^n\right) + \varepsilon_2 q_{w,2}\left(S_j^n\right)\right], \tag{4.3.6}$$

for cell $j = 2, \dots, K-1$ and $n = 1, \dots, N$. The transfer terms $q_{w,1}\left(S_j^n\right)$ and $q_{w,2}\left(S_j^n\right)$ are given by Eqs. (4.3.3)-(4.3.4).

The boundary and initial conditions are

$$S(t, x = 0) = 1, \qquad S(t = 0, x) = 0. \tag{4.3.7}$$

To avoid having to summarize the $\sum_{k=1}^{n} \Delta t * \left(S_j^k\right)^m * e^{-\sigma(t^n - k\Delta t)}$ term for every new time step we will try to find another way to express this summation term within the transfer term q_w. In the following derivation, parameters that are not included in the summation term within the transfer term will be dropped for the sake of brevity. The time step size, Δt, and the saturation exponent, m, will also be excluded until the derivation is finished. Recall that the transfer term has been divided into two parts.

For time step n, an expansion of the summation term can be written as

$$\begin{aligned} \widehat{q_{w,1}^{n}} &= \sum_{k=1}^{n} \left(S_j^k\right) * e^{-\sigma_1(t^n - k\Delta t)} \\ &= S^1 e^{-\sigma_1 (n-1)\Delta t} + S^2 e^{-\sigma_1 (n-2)\Delta t} + \cdots S^{n-1} e^{-\sigma_1(\Delta t)} + S^n, \end{aligned} \tag{4.3.8}$$

$$\begin{aligned} \widehat{q_{w,2}^{n}} &= \sum_{k=1}^{n} \left(S_j^k\right) * e^{-\sigma_2(t^n - k\Delta t)} \\ &= S^1 e^{-\sigma_2 (n-1)\Delta t} + S^2 e^{-\sigma_2 (n-2)\Delta t} + \cdots S^{n-1} e^{-\sigma_2(\Delta t)} + S^n. \end{aligned} \tag{4.3.9}$$

At time step $n+1$ we get

$$\begin{aligned} \widehat{q_{w,1}^{n+1}} &= \sum_{k=1}^{n} \left(S_j^k\right) * e^{-\sigma_1(t^{n+1} - k\Delta t)} = \sum_{k=1}^{n} \left(S_j^k\right) * e^{-\sigma_1(t^{n+1} - k\Delta t)} + S^{n+1} e^{-\sigma_1(t^{n+1} - t^{n+1})} \\ &= S^1 e^{-\sigma_1 (n\Delta t)} + S^2 e^{-\sigma_1 (n-1)\Delta t} + \cdots S^{n-1} e^{-\sigma_1(2\Delta t)} + S^n e^{-\sigma_1(\Delta t)} + S^{n+1} \end{aligned} \tag{4.3.10}$$

$$\begin{aligned}\widehat{q_{w,2}^{n+1}} &= \sum_{k=1}^{n}\left(S_j^k\right) * e^{-\sigma_2\left(t^{n+1}-k\Delta t\right)} = \sum_{k=1}^{n}\left(S_j^k\right) * e^{-\sigma_2\left(t^{n+1}-k\Delta t\right)} + S^{n+1}e^{-\sigma_2\left(t^{n+1}-t^{n+1}\right)} \\ &= S^1 e^{-\sigma_2(n\Delta t)} + S^2 e^{-\sigma_2(n-1)\Delta t} + \cdots S^{n-1}e^{-\sigma_2(2\Delta t)} + S^n e^{-\sigma_2(\Delta t)} + S^{n+1}\end{aligned} \tag{4.3.11}$$

Note that; $S^{n+1}e^{-\sigma_1\left(t^{n+1}-t^{n+1}\right)} = S^{n+1}e^{-\sigma_2\left(t^{n+1}-t^{n+1}\right)} = S^{n+1}$

We now multiply Eqs. (4.3.10)-(4.3.11) with $e^{\sigma_1(\Delta t)}$ and $e^{\sigma_2(\Delta t)}$ respectively

$$\begin{aligned}\widehat{q_{w,1}^{n+1}}e^{\sigma_1(\Delta t)} = S^1 e^{-\sigma_1(n-1)\Delta t} + S^2 e^{-\sigma_1(n-2)\Delta t} + \cdots S^{n-1}e^{-\sigma_1(\Delta t)} + S^n \\ + S^{n+1}e^{\sigma_1(\Delta t)},\end{aligned} \tag{4.3.12}$$

$$\begin{aligned}\widehat{q_{w,2}^{n+1}}e^{\sigma_2(\Delta t)} = S^1 e^{-\sigma_2(n-1)\Delta t} + S^2 e^{-\sigma_2(n-2)\Delta t} + \cdots S^{n-1}e^{-\sigma_2(\Delta t)} + S^n \\ + S^{n+1}e^{\sigma_2(\Delta t)}.\end{aligned} \tag{4.3.13}$$

By comparing Eqs. (4.3.12)-(4.3.13) with Eqs. (4.3.8)-(4.3.9) we see that $\widehat{q_{w,1}^{n+1}}$ and $\widehat{q_{w,2}^{n+1}}$ can be written as

$$\widehat{q_{w,1}^{n+1}} = \widehat{q_{w,1}^{n}}e^{-\sigma_1(\Delta t)} + S^{n+1}e^{-\sigma_1(\Delta t)} = e^{-\sigma_1(\Delta t)}\left(\widehat{q_{w,1}^{n}} + S^{n+1}\right), \tag{4.3.14}$$

$$\widehat{q_{w,2}^{n+1}} = \widehat{q_{w,2}^{n}}e^{-\sigma_2(\Delta t)} + S^{n+1}e^{-\sigma_2(\Delta t)} = e^{-\sigma_2(\Delta t)}\left(\widehat{q_{w,2}^{n}} + S^{n+1}\right). \tag{4.3.15}$$

We now include the excluded parameters together with the expressions for the summation terms given in Eqs. (4.3.14)-(4.3.15) and we obtain new expressions for the transfer terms evaluated at time step $n + 1$

$$q_{w,1}\left(S_j^{n+1}\right) = \left(\left(S_j^{n+1}\right)^m - \sigma_1\Delta t\left(\widehat{q_{w,1}^{n}}e^{-\sigma_1(\Delta t)} + \left(S_j^{n+1}\right)^m e^{-\sigma_1(\Delta t)}\right)\right), \tag{4.3.16}$$

$$q_{w,2}\left(S_j^{n+1}\right) = \left(\left(S_j^{n+1}\right)^m - \sigma_1\Delta t\left(\widehat{q_{w,2}^{n}}e^{-\sigma_2(\Delta t)} + \left(S_j^{n+1}\right)^m e^{-\sigma_2(\Delta t)}\right)\right). \tag{4.3.17}$$

It is beneficial to evaluate the transfer term at time step n in order to maintain an explicit solution. This reduction from time step $n+1$ to n may result in some stability problems.

$$q_{w,1}\left(S_j^n\right) = \left(\left(S_j^n\right)^m - \sigma_1 \Delta t\left(\widehat{q_{w,1}^{n-1}} e^{-\sigma_1(\Delta t)} + (S_j^n)^m e^{-\sigma_1(\Delta t)}\right)\right), \qquad (4.3.18)$$

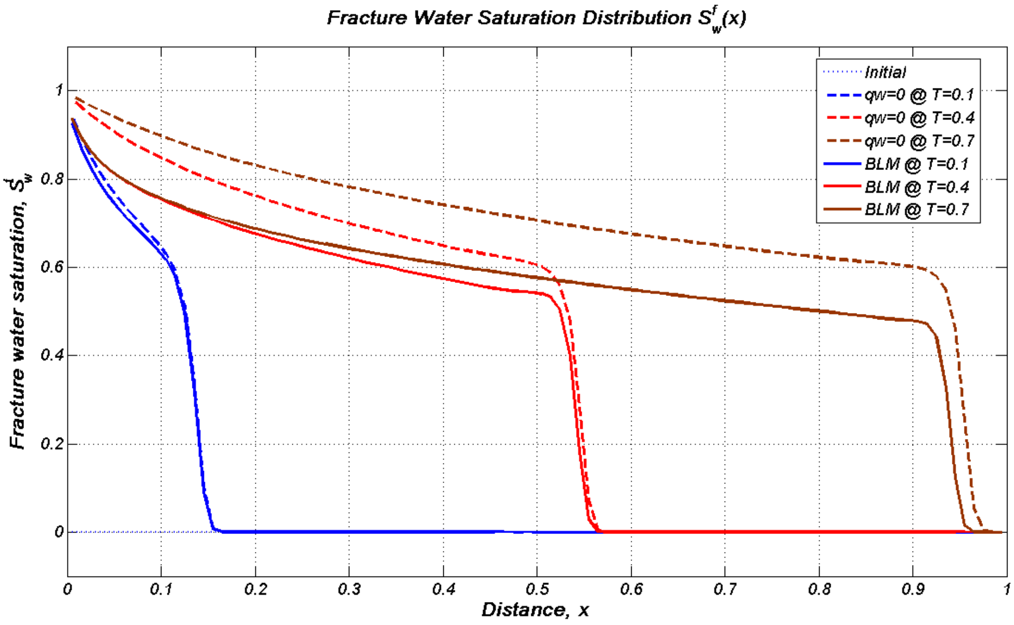
$$q_{w,2}\left(S_j^n\right) = \left(\left(S_j^n\right)^m - \sigma_2 \Delta t\left(\widehat{q_{w,2}^{n-1}} e^{-\sigma_2(\Delta t)} + (S_j^n)^m e^{-\sigma_2(\Delta t)}\right)\right). \qquad (4.3.19)$$

The substitution of Eqs. (4.3.18)-(4.3.19) into Eq. (4.3.6) completes the new formulation and a solution for S_j^{n+1} can be written as

$$S_j^{n+1} = S_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right) - \Delta t\left[\varepsilon_1 * q_{w,1}\left(S_j^n\right) + \varepsilon_2 * q_{w,2}\left(S_j^n\right)\right], \qquad (4.3.20)$$

for cell $j = 2, \dots, K-1$ and $n = 1, \dots, N$. The transfer terms $q_{w,1}\left(S_j^n\right)$ and $q_{w,2}\left(S_j^n\right)$ are given by Eqs. (4.3.18)-(4.3.19), the numerical flux F_j^n is given by Eq. (4.3.2) and the dimensionless constants ε_1 and ε_2 can be found in Eq. (4.2.1).

Figure 25 illustrates the effect the transfer term has on the fracture saturation profile.



Figur 25: Effect of transfer term on fracture water saturation profiles at different times, the dashed lines represents solutions for qw=0

It is observed that as the saturation front moves along the fracture water is lost to matrix due to spontaneous imbibition. This result in a decreasing fracture water front saturation and also tends to decrease the velocity which the front is traveling with, and consequently an increase in breakthrough time is observed. The parameters used to obtain figure 25 are presented in section 4.4.

4.4 Numerical investigation

[5, 9] We consider a water-wet porous media consisting of a fracture with porosity $\varphi^f = 1$ and matrix with porosity $\varphi^m = 0.2$. The media is initially saturated with oil and residual saturations are neglected i.e. $S_{wr} = S_{or} = 0$. The system considers immiscible flow of two incompressible fluids, water and oil. Fracture capillary pressure is assumed zero and the matrix capillary pressure and relative permeabilities are incorporated in the transfer term.

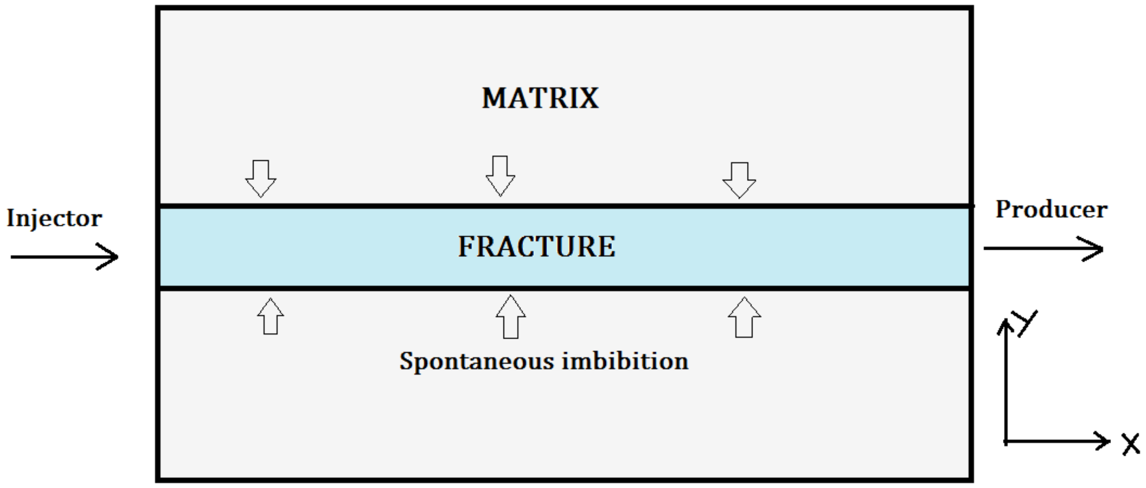


Figure 26: Cross section of the porous media seen from above

The length between injector and producer is set to $L_x = 1\ m$, and the length in y and z-direction is set to $L_y = L_z = 0.245\ m$. This gives a total volume, $V_T = 0.06\ m^3$, which can be separated into a matrix volume, $V_m = 0.05\ m^3$, and a fracture volume, $V_f = 0.01\ m^3$.

We want to express the volumes in terms of pore volumes. The total pore volume is given by

$$PV_T = PV_f + PV_m = V_f * \varphi^f + V_m * \varphi^m = 0.02\ m^3 \qquad (4.4.1)$$

Note that 1 fracture pore volume is equal to 1 matrix pore volume i.e. $PV_f = PV_m$.

For the first demonstration we will use the following constant input parameters: Injection rate $Q_T = \frac{1}{2} PV_T/day = 1.16 * 10^{-7}\ m^3/s$, injection velocity $u_T = Q_T/A_{yz} = 1.93 * 10^{-6}\ m/s$, fracture Corey exponent $n = 2$, fracture capillary pressure is assumed zero and the transfer term parameters are set to $R_1 = 0.10$, $R_2 = 0.02$, $\sigma_1 = 1.11 * 10^{-5}\ s^{-1}$, $\sigma_2 = 2.78 * 10^{-6}\ s^{-1}$, $m = 0.5$, and thus the dimensionless constants becomes $\varepsilon_1 = \frac{L(R_1\sigma_1)}{u_T} = 0.576$, $\varepsilon_2 = \frac{L(R_2\sigma_2)}{u_T} = 0.029$.

Remark: SI units have been used, as opposed to units used in [5]. Similar ratios between transfer term parameters presented in [5] have been used in this demonstration.

To study the oil recovery we can divide the total oil recovery into two parts, recovery from fracture, R_f, and recovery from matrix, R_m. For this case where $PV_f = PV_m = \frac{1}{2} PV_T$, we can write the total oil recovery as follows

$$R_T = R_f + R_m = \frac{A(t) * PV_f}{PV_T} + \frac{q(t) * PV_m}{PV_T} = \frac{1}{2} A(t) + \frac{1}{2} q(t), \qquad (4.4.1)$$

where $A(t) = \sum_{t=0}^{t} S_w^f * dx$ and $q(t) = \sum_{t=0}^{t} q_w * dx$, and where $dx = \frac{1}{N} = 0.01$.

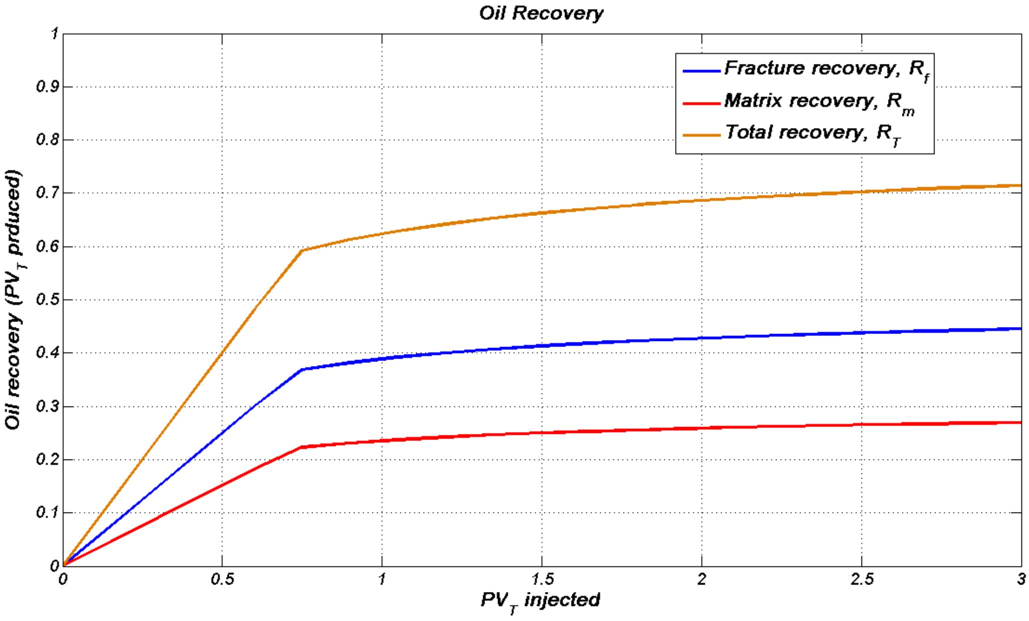


Figure 26: Fracture, matrix and total oil recovery

It is observed that after injection of 3 PV_T, ca. 89 % of the oil in the fracture has been produced and ca. 54 % matrix-oil has been produced due to spontaneous imbibition. It is observed that the fracture recovery rate is highest both before and after breakthrough.

To study the difference between strong and weak spontaneous imbibition, simulations at different injection rates were performed. It is observed that a decrease in injection rate i.e. a decrease in injection velocity results in a proportional increase of the dimensionless numbers ε_1 and ε_2, which in turn increases the transfer term effect, resulting in a higher fracture-matrix fluid exchange i.e. stronger spontaneous imbibition.

From figure 27 we observe that as the injection rate decreases more water is lost to matrix and as a result a higher breakthrough time is obtained. Recall that dimensionless time is equal to pore volumes injected and that the solution at $T = T_{bt}$ has its front at $x = 1$.

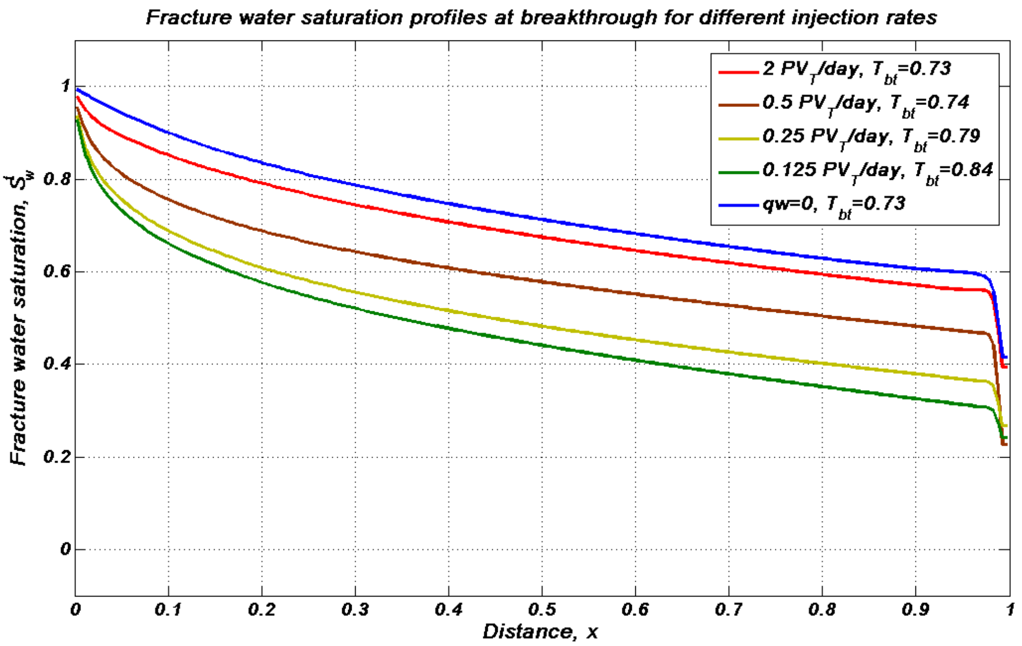


Figure 27: Comparison of fracture water saturation distribution for different injection rates at $T = T_{bt}$

Note that: The same parameters as in the previous example have been used to obtain figure 27 and 28. Only the injection velocity have been varied, except for the case when injection rate $Q_T = 0.125\ PV_T/day$. Recall that the cocurrent imbibition parameter σ_1 might vary with the injection rate based on observations made by Terez and Firoozabadi [5]. It was observed that simulations performed at $Q_T = 0.125\ PV_T/day$ with the given σ_1 produced unrealistic solutions. Thus σ_1 has been reduced by a factor of 1.6 at this injection rate.

Although water is lost to matrix and hence oil recovery from the fracture is reduced, relatively more oil is recovered from matrix at a given time due to spontaneous imbibition resulting in a higher total recovery compared with only recovery from fracture with transfer term set to zero (compare with blue line in figure 28). Figure 28 illustrates the increase in breakthrough time and recovery as injection rate is reduced.

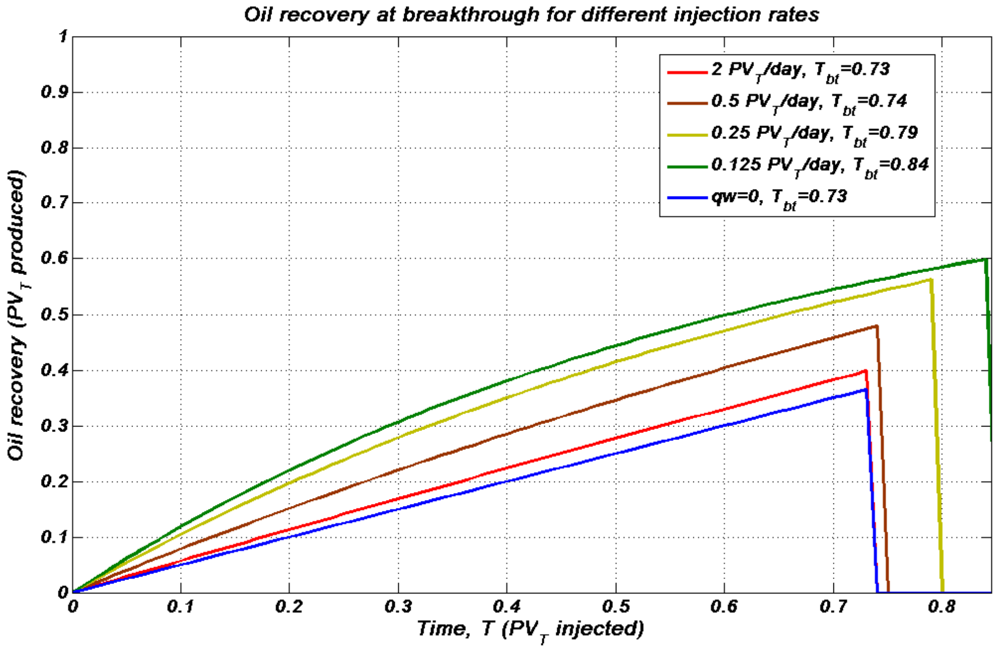


Figure 28: Oil recovery at breakthrough for different injection rates, strong vs. weak spontaneous imbibition

References

[1] “The two-phase porous media equation in 1D” A bachelor-thesis in petroleum engineering by Per Kristian Skretting & Ole Skretting Helvig, University of Stavanger, Spring 2011

[2] Liping Yu, Hans Kleppe, Terje Kaarstad and Svein M Skjaeveland, Univeristy of Stavanger. Steinar Evje and Ingebret Fjelde, International Research Institute of Stavanger (IRIS) “Modelling of Wettability Alteration Processes in Carbonate Oil Reservoirs”

[3] S.M. Skjaeveland, SPE, L.M. Siqveland, and A. Kjosavik, SPE, Stavanger College; W.L. Hammervold Thomas, SPE, Statoil; and G.A. Virnovsky, SPE, RF-Rogaland Research SPE 60900 “Capillary Pressure Correlation for Mixed-Wet Reservoirs”

[4] Frode Lomeland1, Einar Ebeltoft2 1 StatoilHydro ASA, Stavanger, 2 Reslab Integration AS, Bergen, Norway SCA2008-08 “A NEW VERSATILE CAPILLARY PRESSURE CORRELATION”

[5] Ivan E. Terez,* SPE, and Abbas Firoozabadi, SPE, Reservoir Engineering Research Inst. (RERI) SPE 56854 “Water Injection in Water-Wet Fractured Porous Media: Experiments and a New Model with Modified Buckley-Leverett Theory”

[6] Francisco J. Rosado-Vazquez and Edgar R. Rangel-German, National Autonomous University of Mexico, and Fernando Rodriquez-de la Garza, Pemex Exploration and Production SPE 108722 “Analysis of Capillary, Gravity and Viscous Forces Effects in Oil/Water Displacement”

[7] Dimo Kashchiev and Abbas Firzoozabadi, SPE, RERI SPE 87333 “Analytical Solutions for 1D Countercurrent Imbibition in Water-Wet Media”

[8] Douglas and Wagner 1958 “Calculation of Linear Waterflood Behavior Including the Effects of Capillary Pressure”

[9] Pål Østebø Andersen, Steinar Evje, Hans Kleppe, Svein Magne Skjæveland, University of Stavanger, ITPC 16957 “A Model for Wettability Alteration in Fractured Reservoirs”

[] Buckley, S., E., and Leverett, M., C.: Mechanism of Fluid Displacement in Sands, Trans. AIME, (1942) 146, 107.

[] Rapoport, L., A., and Leas, W., J.: Properties of Linear Waterfloods, Trans. AIME (1953) 198, 139

[] S. Jin and Z. Xin The relaxation schemes for systems of conservation laws in arbitrary space dimensions, Comm. Pure Appl. Math., 48 (1995), 235–276.

Appendix A

Matlab codes for analytic solution

```
load ParaScalar.DEF;
N     = ParaScalar(1,1);
T = ParaScalar(1,2);
a     = ParaScalar(1,3);
NSTEP = ParaScalar(1,4);
M     = ParaScalar(1,7);
no = ParaScalar(1,8);
nw = ParaScalar(1,9);


%% Ant Noder (N)   TIME     e f'   NSTEP       mu_o  mu_w  M  nw  no    T
%
%                 CASE I                             CASE analytical
     100            0.5       4      20          2    1   0.5  2   2    0.5
```

Corey relative permeabilities [A1]

```
function [fw,fo] = rel_perm(u,no,nw)

fw   = zeros(size(u));
fo   = zeros(size(u));

fw = u.^nw;
fo = (1-u).^no;

%Plot Corey relperms
figure(2)
s = 0:0.01:1;
[kw,ko] = rel_perm(s,no,nw);
plot(s,kw,'-g','LineWidth',2);hold on
plot(s,ko,'-r','LineWidth',2);hold on
drawnow;
title('corey relperms');
xlabel('s-axis')
ylabel('kri')
```

Fractional flow curve [A2]

```
function f=Fflux(u,M,no,nw)

f   = zeros(size(u));

[fw,fo] = rel_perm(u,no,nw);

f = fw./(fw+M*fo);
```

```
%Plot f(s)
s=0:0.01:1;
F = Fflux(s,M,no,nw);
plot(s,F,'-','LineWidth',2);hold on
axis([0 1 -0.1 1.1]);
drawnow;
grid on
xlabel('s-axis')
ylabel('f-axis')
hold off
```

[1] Fractional flow derivative [A3]

```
function f=Fflux_Df(s,M,no,nw)

f = zeros(size(s));

f(:) = ((M.*(((1-s(:).^(nw-1)).*s(:).^(no-1).*(no.*(1-
s(:))+s(:).*nw))))./((s(:).^no+M.*(1-s(:)).^nw)).^2

%plot Df
figure(3)
s = 0:0.01:1;
plot(s,Fflux_Df(s,M,no,nw),'-');hold on
drawnow;
title('flux function Df(s)');
xlabel('s-axis')
ylabel('Df-axis')
```

[1] Saturation movement, unphysical solution [A4]

```
function f = Fflux_DfT(s,M,no,nw,T)

f = zeros(size(s));

f(:) = T.*((M.*(((1-s(:).^(nw-1)).*s(:).^(no-1).*(no.*(1-
s(:))+s(:).*nw))))./((s(:).^no+M.*(1-s(:)).^nw)).^2;

% plot s(x) unphysical solution
figure(4)
plot(Fflux_DfT(s,M,no,nw,T),s),'-';hold on
drawnow;
title('Saturation movement, unphysical solution');
xlabel('x-axis')
ylabel('s-axis')
```

[1] Saturation movement, area balancing [A5]

```
function f = DfT_sol(s,M,no,nw,T)

f = zeros(size(s));

f(:) = T.*((M.*(((1-s(:).^(nw-1)).*s(:).^(no-1).*(no.*(1-
s(:))+s(:).*nw))))./((s(:).^no+M.*(1-s(:)).^nw)).^2;

% plot s(x) the solution with area

figure(5)
xstar = T*((M*(((1-sstar^(nw-1))*sstar^(no-1)*(no*(1-
sstar)+sstar*nw)))))/((sstar^no+M*(1-sstar)^nw))^2;
plot(DfT_sol(s,M,no,nw,T),s),'-';hold on
s = 0:0.01:sstar;
line([xstar;xstar],[0;sstar]);

drawnow;
title('Saturation movement Area balancing');
xlabel('x-axis')
ylabel('s-axis')
```

[1] Determination of front saturation height [A6]

```
function f = Fflux_tangent(s,M,no,nw)

f = zeros(size(s));

f(:) = (s(:).*((M.*(((1-s(:).^(nw-1)).*s(:).^(no-1).*(no.*(1-
s(:))+s(:).*nw)))) ...
    ./((s(:).^no+M.*(1-s(:)).^nw)).^2)-s(:).^no./(((s(:).^no)+(M.*(1-
s(:)).^nw)));

% plot determination of the front saturation height function
figure (6)
sstar = fzero(@(x) Fflux_tangent(x,M,no,nw),0.5);
fstar = sstar^no/((sstar^no)+(M*(1-sstar)^nw));
plot(s,Fflux_tangent(s,M,no,nw),'-');hold on
drawnow;
```

S_f = sstar = 0.5774

[1] Graphical determination of the front saturation [A7]

```
% plot flux function f(s)
figure(7)
s = 0:0.01:1;
plot(s,Fflux_f(s,M,no,nw),'-');hold on
drawnow;
title('flux function f(s)');
xlabel('s-axis')
ylabel('f-axis')

%tangent to fractional flow curve
sstar = fzero(@(x) Fflux_tangent(x,M,no,nw),0.5)
fstar = sstar^no/((sstar^no)+(M*(1-sstar)^nw))
% gradient a
a = fstar/sstar;
s = 0:0.01:1;
g = s(:).*a;
plot(s,g),('--g');hold on
```

[1] Saturation distribution after time T [A8]

```
% plot s(x) the solution,
figure (8)
xstar = T*((M*(((1-sstar^(nw-1))*sstar^(no-1)*(no*(1-
sstar)+sstar*nw)))))/((sstar^no+M*(1-sstar)^nw))^2;
s = sstar:0.01:1;
plot(DfT_sol(s,M,no,nw,T),s),'-';hold on
drawnow;
title('Saturation movement');
xlabel('x-axis')
ylabel('s-axis')
s = 0:0.01:sstar;
line([xstar;xstar],[0;sstar]);
```

Diffusion coefficient, a(s) [A9]

```
function f = diff_coeff(u,F,ko,mu_oD)

f   = zeros(size(u));
f = (-1/mu_oD).*(ko.*F);

% plot diffusion coefficient a(s)
subplot(2,2,3)
A = diff_coeff(s,F,ko,mu_oD);
plot(s,A,'-b','LineWidth',2);hold on
axis([0 1 -0.14 0]);
grid on
legend('a(s)')
title('Diffusion coefficient a(s)');
xlabel('s-axis')
ylabel('a-axis')
```

Capillary pressure correlations [A10]

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Capillary pressure functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Pc=cap_pressure(u,B,cw,co,aw,ao,Pmax,Pmin,Es,Ef,Ls,Lf,Ts,Tf)

Pc = zeros(size(u));

%Capillary pressure correlation for mixed-wet reservoirs

%Pc = B.*((cw./(u+0.01).^aw) + (co./((1-u+0.01).^ao)));

% [7]
%Pc = -B*log(u+0.01);

%A new versatile capillary pressure correlation,

Pc = ((Pmax.*((1-u).^Ls))./(((1-u).^Ls)+(Es.*((u).^Ts))))...
    +((Pmin.*((u).^Lf))./(((u).^Lf)+(Ef.*((1-u).^Tf))));


%E-parameter for spontaneous imbibition

function Es=Es(Swsin,Pmax,Pmin,Ef,Ls,Lf,Ts,Tf)

Es = zeros(size(Swsin));

Es = -((Pmax*((Swsin^Lf)+(Ef*((1-Swsin)^Tf))))/(Pmin*(Swsin^Lf))+1)...
    *(((1-Swsin)^Ls)/(Swsin^Ts));


% plot capillary pressure function Pc(s)
%Note: Constants need to be specified;
%B,cw,co,aw,ao,Pmax,Pmin,Ef,Ls,Lf,Ts,Swsin
figure(2)
subplot(2,2,1)
Es = Es(Swsin,Pmax,Pmin,Ef,Ls,Lf,Ts,Tf)
Pc = cap_pressure(s,B,cw,co,aw,ao,Pmax,Pmin,Es,Ef,Ls,Lf,Ts,Tf);
plot(s,Pc,'-r','LineWidth',2);hold off
drawnow;
grid on
legend('P_c(s)')
title('Capillary pressure, P_c(s)');
xlabel('s-axis')
ylabel('P_c-axis')
hold on
%Swsi
Swsi = fzero(@(s)
cap_pressure(s,B,cw,co,aw,ao,Pmax,Pmin,Es,Ef,Ls,Lf,Ts,Tf),0.4)
```

Appendix B

Matlab codes for numerical solution of the BL-equation including capillary pressure

Note: I got handed a complete numerical code and a solution procedure. All modifications conducted by me alone or in cooperation with my supervisor are noted with the symbol “**”.

```
% Relaxed Scheme

function
%%[v2]=sol_relaxedScalar_ny(v0,dt,x,dx,e,M,no,nw,e2,eps,B,cw,co,aw,ao,mu_oD
,Pmax,Pmin,Es,Ef,Ls,Lf,Ts,Tf,Sws)

n = length(x);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bestem antall ordinaere tidssteg
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%dT = [0.75*dx/(e) , **0.25*dx^2/(e2)]
**dT = min(dT)

nstep = fix(dt/dT +1.5);
dT = dt/nstep;
fprintf(1,'\n----------------------------------------\n');
fprintf(1,'       Relaxed Scheme Scalar   \n');
fprintf(1,'------------------------------------------\n');
fprintf(1,'Antall steg               : %d\n', nstep);
fprintf(1,'Antall noder              : %d\n', length(x));
fprintf(1,'max Df                    : %f\n', e);
fprintf(1,'max (a(s)*DPc)            : %f\n', e2);
fprintf(1,'min(dT)                   : %f\n', dT);
fprintf(1,'epsilon                   : %f\n', eps);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Beregn loesning ved vanlig metode
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

v1          = zeros(size(v0));
v1_s        = zeros(size(v0));
w1_s_half   = zeros(size(v0));

v2          = zeros(size(v0));
f_s         = zeros(size(v0));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**kr_o       = zeros(size(v0));
**kr_w       = zeros(size(v0));
**a_s        = zeros(size(v0));
**Pc_s       = zeros(size(v0));
**a_s_half   = zeros(size(v0));
**DPc_s_half = zeros(size(v0));
**DPc_s_L    = zeros(size(v0));
**a_s_half_L = zeros(size(v0));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fpl          = zeros(size(v0));
fmi          = zeros(size(v0));
Dfpl         = zeros(size(v0));
Dfmi         = zeros(size(v0));
sigma_pl     = zeros(size(v0));
sigma_mi     = zeros(size(v0));

ts = cputime;

v1 = v0;

lambda = dT/dx;
i_dx   = 1/dx;

I=2:n-1;
J=1:n-1;
K=1:1;

for i=1:nstep

    %%%%%%%%%%%%%%
    %%  Step 1  %%
    %%%%%%%%%%%%%%
    v1_s = v1;
    v1_s_L = 1.0;

    f_s   = Fflux(v1_s,M,no,nw);
    f_s_L = Fflux(v1_s_L,M,no,nw);

   **[kr_w,kr_o] = rel_perm(v1_s,no,nw);
   **kr_o_L = rel_perm(v1_s_L,no,nw);

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
**a_s    = -(kr_o/mu_oD).*f_s;
**Pc_s    = cap_pressure(v1_s,B,cw,co,aw,ao,Pmax,Pmin,Es,Ef,Ls,Lf,Ts,Tf);

**a_s_L =  -(kr_o_L/mu_oD).*f_s_L;
**Pc_s_L =  cap_pressure(v1_s_L,B,cw,co,aw,ao,Pmax,Pmin,Es,Ef,Ls,Lf,Ts,Tf);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    fpl(K,:)  = f_s(K,:) + e*v1_s(K,:);
    fmi(K,:)  = f_s(K,:) - e*v1_s(K,:);
    Dfpl(K,J) = i_dx*(fpl(K,J+1)-fpl(K,J));
    Dfmi(K,J) = i_dx*(fmi(K,J+1)-fmi(K,J));

    fpl_L  = f_s_L + e*v1_s_L;
    fmi_L  = f_s_L - e*v1_s_L;
    Dfpl_L = i_dx*(fpl(K,1)-fpl_L);
    Dfmi_L = i_dx*(fmi(K,1)-fmi_L);
```

```
    %%%%%%%%%%%%%%%
    % sigma_pluss %
    %%%%%%%%%%%%%%%
    sigma_pl_L  = Sigma( Dfpl_L,Dfpl(K,1) );
    sigma_pl_LL = Sigma( 0,Dfpl_L );

    sigma_pl(K,1)  = sigma_pl_L;
    sigma_pl(K,n)  = 0;
    sigma_pl(K,I)  = Sigma( Dfpl(K,I-1),Dfpl(K,I) );

    %%%%%%%%%%%%%%%
    % sigma_minus %
    %%%%%%%%%%%%%%%
    sigma_mi(K,1)  = Sigma( Dfmi_L,Dfmi(K,1) );
    sigma_mi(K,n)  = 0;
    sigma_mi(K,I)  = Sigma( Dfmi(K,I-1),Dfmi(K,I) );


    % MUSCL Case
    w1_s_half(K,J)  = 0.5*( fpl(K,J) + 0.5*dx*sigma_pl(K,J) ) ...
                   + 0.5*( fmi(K,J+1) - 0.5*dx*sigma_mi(K,J+1) );

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Legg inn beskrivelse av a(s) og Pc(s)

    **a_s_half(K,J)   = 0.5*(a_s(K,J+1) + a_s(K,J));
    **DPc_s_half(K,J) = ((Pc_s(K,J+1) - Pc_s(K,J))/dx);

    **DPc_s_L = (Pc_s(K,1) - Pc_s_L)/(dx*0.5);
    **a_s_half_L = a_s(K,1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % flux at inlet end
    w1_s_half_L = f_s_L;

    % interior domain

**v2(K,I) = v1_s(K,I) - lambda*(w1_s_half(K,I) - w1_s_half(K,I-1))...
             + lambda*eps.*((a_s_half(K,I).*DPc_s_half(K,I)) -
            (a_s_half(K,I-1).*DPc_s_half(K,I-1)));

    % first cell

    **v2(K,1) = v1_s(K,1) - lambda*(w1_s_half(K,1) - w1_s_half_L)...
       + lambda*eps.*((a_s_half(K,1) * DPc_s_half(K,1)) - (a_s_half_L *
       DPc_s_L));


    % last cell
     %v2(1,n) = v2(1,n-1);

     %including capillary end effects
     **v2(1,n) = Sws;
```

```
%% Update to prepare for new local timestep  %%

 v1 = v2;     % relaxating scheme

end
etime = cputime - ts;
fprintf(1,'CPU-tid                         : %f\n', etime );
%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Van Leer limiter         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Sigma=Sigma(u,v)

Sigma = zeros(size(u));
L = abs(v) >= 0.0001;
theta = zeros(size(L));

Sigma(~L)=0;

theta(L)=u(L)./v(L);
Sigma(L)=v(L).*( (abs(theta(L))+theta(L))./(1+abs(theta(L))) );

%Sigma = zeros(size(u));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Solution Procedure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% grid in space
dx = 1.0/N;
x  = 0+0.5*dx:dx:1.0-0.5*dx;
n  = length(x);

% les inn initial data
u0 = zeros(1,n);
u0 = init_func(x);

 % plot initial data (sjekk)
figure(1)
subplot(2,2,2)
plot(x,u0(:),':','LineWidth',2);hold on
axis([0 1 -0.1 1.1]);
drawnow;
grid on
title('Initial Water Saturation s_0');
xlabel('x-axis')
ylabel('s-axis')

pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Solution Procedure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u     = zeros(1,n);
u_pre = zeros(1,n);

oil_recovery = zeros(1,NSTEP+1);

% first timestep
u_pre = u0;

dt = Time./NSTEP;
t  = 0:dt:Time;
oil_recovery(1)  = 0;

for step = 1:NSTEP
fprintf(1,'\n------------------------------------------\n');
fprintf(1,'Bestemmer lÖsning ved tid : %f\n', dt*step);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate solution
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**[u]=sol_relaxedScalar_ny(u_pre,dt,x,dx,a,M,no,nw,a2,epsi,B,cw,co,aw,ao,mu
_oD,Pmax,Pmin,Es,Ef,Ls,Lf,Ts,Tf,Swsi);
```

```
% calculate oilrecovery
oil_recovery(step+1) = dx*sum(u);

% plot oilrecovery
figure(1)
subplot(2,2,4)
plot(t,oil_recovery,'-g','LineWidth',2);hold off
title('Oil Recovery');
xlabel('time-axis')
ylabel('oilrecovery-axis')
axis([0 Time 0 1]);
grid on
drawnow;
hold on

% plot initial data + solution
figure(2)
subplot(2,2,4)
plot(x,u0(:),':','LineWidth',2);hold on
title('Water Saturation Distribution s(x)');
xlabel('x-axis')
ylabel('s-axis')
plot(x,u(:),'-','LineWidth',2);
axis([0 1 -0.1 1.1]);
grid on
drawnow;
hold on

% Oppdatering av initial data for hyperpolic part
u_pre = u;

pause

%%%%%%%%%%%%%%%%%%%%%%%
end  % time step loekke

% print to file
Tabell_1 = [x; u];
fid = fopen('num.data','w');
fprintf(fid,'%f %f\n', Tabell_1);
fclose(fid);

pause

% plot exact solution in the same plot as numerical solution
load sol.data
hold on
x1  = sol(:,1);
u1  = sol(:,2);
figure(2)
subplot(2,2,4)
plot(x1,u1,'-r','LineWidth',2);
grid on
%legend('initial saturation','exact solution','numerical solution')
hold on
```

Appendix C

Matlab codes for numerical solution of the modified Buckley-Leverett equation

Note: I got handed a complete numerical code and a solution procedure. All modifications conducted by me alone or in cooperation with my supervisor are noted with the symbol "**".

```
% Relaxed Scheme

function
**[v2,q1_s]=sol_relaxedScalar_qw_ny1(v0,q0,dt,x,dx,e,M,no,nw,sigma1,sigma2,
eps1,eps2,m)

% Relaxed Scheme

n = length(x);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bestem antall ordinaere tidssteg
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dT = 0.75*dx/(e);
nstep = fix(dt/dT +1.5);
dT = dt/nstep;
fprintf(1,'\n-----------------------------------------\n');
fprintf(1,'       Relaxed Scheme Scalar   \n');
fprintf(1,'-------------------------------------------\n');
fprintf(1,'Antall steg               : %d\n', nstep);
fprintf(1,'Antall noder              : %d\n', length(x));
fprintf(1,'max Df                    : %f\n', e);
fprintf(1,'dT                        : %f\n', dT);
fprintf(1,'Cocurrent constant        : %f\n', eps1);
fprintf(1,'Countercurrent constant   : %f\n', eps2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Beregn loesning ved vanlig metode
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


v1         = zeros(size(v0));
v1_s       = zeros(size(v0));
w1_s_half  = zeros(size(v0));

v2         = zeros(size(v0));
f_s        = zeros(size(v0));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


**q1          = zeros(size(v0));
**q1_s        = zeros(size(v0));
**q2          = zeros(size(v0));
**q1_loc_old = zeros(size(v0));
**q2_loc_old = zeros(size(v0));
**q1_loc_new = zeros(size(v0));
**q2_loc_new = zeros(size(v0));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fpl          = zeros(size(v0));
fmi          = zeros(size(v0));
Dfpl         = zeros(size(v0));
Dfmi         = zeros(size(v0));
sigma_pl     = zeros(size(v0));
sigma_mi     = zeros(size(v0));

ts = cputime;

v1 = v0;
**q1 = q0;

lambda = dT/dx;
i_dx   = 1/dx;


I=2:n-1;
J=1:n-1;
K=1:1;

for i=1:nstep

    %%%%%%%%%%%%%%
    %%  Step 1  %%
    %%%%%%%%%%%%%%
    v1_s = v1;
    v1_s_L = 1.0;

  **q1_s = q1;

    f_s   = Fflux(v1_s,M,no,nw);
    f_s_L = Fflux(v1_s_L,M,no,nw);

  **q1_loc_old = q0;
  **q2_loc_old = q0;
  **q1_s = SOURCE(v1_s,q1_loc_old,q2_loc_old,dT,sigma1,sigma2,eps1,eps2,m);

    fpl(K,:)  = f_s(K,:) + e*v1_s(K,:);
    fmi(K,:)  = f_s(K,:) - e*v1_s(K,:);
    Dfpl(K,J) = i_dx*(fpl(K,J+1)-fpl(K,J));
    Dfmi(K,J) = i_dx*(fmi(K,J+1)-fmi(K,J));

    fpl_L  = f_s_L + e*v1_s_L;
    fmi_L  = f_s_L - e*v1_s_L;
    Dfpl_L = i_dx*(fpl(K,1)-fpl_L);
    Dfmi_L = i_dx*(fmi(K,1)-fmi_L);

    %%%%%%%%%%%%%%%
    % sigma_pluss %
    %%%%%%%%%%%%%%%
    sigma_pl_L  = Sigma( Dfpl_L,Dfpl(K,1) );
    sigma_pl_LL = Sigma( 0,Dfpl_L )
    sigma_pl(K,1) = sigma_pl_L;
    sigma_pl(K,n) = 0;
    sigma_pl(K,I) = Sigma( Dfpl(K,I-1),Dfpl(K,I) )
```

```
    %%%%%%%%%%%%%%%
    % sigma_minus %
    %%%%%%%%%%%%%%%
    sigma_mi(K,1) = Sigma( Dfmi_L,Dfmi(K,1) );
    sigma_mi(K,n) = 0;
    sigma_mi(K,I) = Sigma( Dfmi(K,I-1),Dfmi(K,I) );

  **q1_s(K,I);

    % MUSCL Case
    w1_s_half(K,J)  = 0.5*( fpl(K,J) + 0.5*dx*sigma_pl(K,J) ) ...
                   + 0.5*( fmi(K,J+1) - 0.5*dx*sigma_mi(K,J+1) );


    % flux at inlet end
    w1_s_half_L = 1.0;

    % interior domain

  **v2(K,I) = v1_s(K,I) - lambda*(w1_s_half(K,I) - w1_s_half(K,I-1)) -
              dT*q1_s(K,I);

    % first cell
  **v2(K,1) = v1_s(K,1) - lambda*(w1_s_half(K,1) - w1_s_half_L) -
              dT*q1_s(K,1);

    % last cell
     v2(1,n) = v2(1,n-1);

   **q1_loc_new = exp(-sigma1*dT)*(q1_loc_old+v2);
   **q2_loc_new = exp(-sigma2*dT)*(q2_loc_old+v2);

%% Update to prepare for new local timestep  %%

 v1 = v2;    % relaxating scheme
**q1_loc_old = q1_loc_new;
**q2_loc_old = q2_loc_new;
**q1 = q1_s;
**q1 = q2;

end
etime = cputime - ts;
fprintf(1,'CPU-tid                   : %f\n', etime );
%%%%%%%%%%%%%%%%%%%%%%%
function Sigma=Sigma(u,v)

Sigma = zeros(size(u));
L = abs(v) >= 0.0001;
theta = zeros(size(L));

Sigma(~L)=0;

theta(L)=u(L)./v(L);
Sigma(L)=v(L).*( (abs(theta(L))+theta(L))./(1+abs(theta(L))) );

%Sigma = zeros(size(u));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Transfer term qw
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**function qw=SOURCE(u,q1_loc_old,q2_loc_old,dT,sigma1,sigma2,eps1,eps2,m)

**qw = zeros(size(u));

**qw = ((eps1*((u.^m)-(sigma1*dT*((q1_loc_old*exp(-
        sigma1*dT))+((u.^m)*(exp(-sigma1*dT)))))))+...
       (eps2*((u.^m)-(sigma2*dT*((q2_loc_old*exp(-
        sigma2*dT))+((u.^m)*(exp(-sigma2*dT))))))));


%Viscous and transfer term parameters
L    = 1;
%L    = 0.4553 %Berea single slab
u_t  = 0.5*(3.8564*10^-6) %Demonstration case 0.5 PVt/day

%Transfer term constants
sigma1 = 1/(25*3600); %Demonstration case @ 0.5 PVt/day
sigma2 = 1/(100*3600); %Demonstration case @ 0.5 PVt/day
R1 = 0.10; %Demonstration case @ 0.5 PVt/day
R2 = 0.02; %Demonstration case @ 0.5 PVt/day
m = 0.5;

eps1 = (L*R1*sigma1)/u_t;
eps2 = (L*R2*sigma2)/u_t;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Solution Procedure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% grid in space
dx = 1.0/N;
x  = 0+0.5*dx:dx:1.0-0.5*dx;
n  = length(x);

% les inn initial data
u0 = zeros(1,n);
u0 = init_func(x);
**q0 = zeros(1,n);
**q0 = init_func_qw(x);

% plot initial data (sjekk)
figure(1)
subplot(2,2,2)
plot(x,u0(:),':','LineWidth',2);hold on
axis([0 1 -0.1 1.1]);
drawnow;
grid on
title('Initial Water Saturation s_0');
xlabel('x-axis')
ylabel('s-axis')

pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Solution Procedure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u     = zeros(1,n);
u_pre = zeros(1,n);

%Legg inn qw og qw_pre
**qw   = zeros(1,n);
**qw_pre = zeros(1,n);

oil_recovery = zeros(1,NSTEP+1);

% first timestep
u_pre = u0;
**qw_pre = q0;

dt = Time./NSTEP;
t  = 0:dt:Time;
oil_recovery(1) = 0;

for step = 1:NSTEP
fprintf(1,'\n------------------------------------------\n');
fprintf(1,'Bestemmer lÖsning ved tid : %f\n', dt*step);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate solution
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**[u,qw]=sol_relaxedScalar_qw_ny1(u_pre,qw_pre,dt,x,dx,a,M,no,nw,sigma1,
                                  sigma2,eps1,eps2,m);

% calculate oilrecovery
%oil_recovery(step+1) = dx*sum(u)
**oil_recovery(step+1) = dx*sum(u)*0.5+dx*sum(qw)*0.5;
%oil_recovery(step+1) = dx*sum(qw)*0.5

% plot oilrecovery
figure(1)
subplot(2,2,4)
plot(t,oil_recovery,'-g','LineWidth',2);hold on
title('Oil Recovery');
xlabel('time-axis')
ylabel('oilrecovery-axis')
axis([0 Time 0 1]);
grid on
drawnow;
hold on

% plot initial data + solution
figure(1)
subplot(2,2,3)
plot(x,u0(:),':','LineWidth',2);hold on
title('Water Saturation Distribution s(x)');
xlabel('x-axis')
ylabel('s-axis')
plot(x,u(:),'-','LineWidth',2); hold on
axis([0 1 -0.1 1.1]);
grid on
drawnow;
hold on

% Oppdatering av initial data for hyperpolic part
u_pre = u;
**qw_pre = qw;
%q1_loc_old=q1_loc_new

pause

%%%%%%%%%%%%%%%%%%%%%%%
end  % time step loekke
```