NTNU – Trondheim
Norwegian University of
Science and Technology

Theoretical and numerical study of polymer flooding

Sebastian Osvaldo Benavides

Petroleum Geoscience and Engineering
Submission date: June 2015
Supervisor: Ole Torsæter, IPT

Norwegian University of Science and Technology
Department of Petroleum Engineering and Applied Geophysics

Preface

Initially I thought I would have at least a month to write down this part. Who would have thought I would end up writing it the day before the deadline, still cursing my brain for getting stuck in endless Matlab loops and saturating the body limits by increasing the quality of the thesis, hoping for no faults in place and predicting an equivalent product. The story goes like this. Arriving at the starting line and being notified I now belonged to the breed that enjoyed testing things in a lab environment, I headed out to meet my professor and advisor. We discussed a variety of interesting viable topics in this vast and different branch and agreed to leave the lab out of it this time. The plan we agreed to consisted of the work being done in half the time, ten weeks. Some say ignorance is a bliss but that sealed my future. In retrospective I may have enjoyed it more, tested things I later found in my bibliography and avoided to be trapped in my programming as I had been rusting away for some time prior to this. Which is why, thanks to my professor, for not giving up.

Acknowledgement

I would like to thank my family for being there when I needed them the most, not only my direct family but also my *entire* family. Those close to me now and those that are still close to me, far away. I thank my friends and thank the department and the people in the administration for being understanding and giving me another chance. I also would like to thank who, once asked to, believed in me and made sure I came back. Finally, I give my thanks to my advisor for providing interesting discussions, hinting me when I was going in the wrong direction and at times, telling me what I was lacking.

S.B.

Summary

In this work, a theoretical and numerical study of polymer flooding has been realized. It consists of a literature study of polymer flooding, the parameters affecting its performance and the implication of gel formation in oil fields. Includes lab studies and field experiences to date. It concludes with a theoretical review of flow in porous media that has been expanded to three-dimensional rectangular coordinates, followed by numerical simulations in both Matlab and Eclipse, with the code created to run the simulation in Matlab included.

Sammendrag

I dette arbeidet, en teoretisk og numerisk studie av polymer oversvømmelse har blitt gjennomført. Den består av et litteratur studie av polymer oversvømmelse, de parameterne som påvirker oppførselen av denne og implikasjonen av en gelé struktur i olje felter. Den konkluderer med et teoretisk gjennomgåelse av strømning i porøse medier som har blitt utvidet til tre dimensjoner i rektangulære koordinater, etterfulgt av numerisk simuleringer i både Matlab og Eclipse, med koden laget for å kjøre simuleringen i Matlab inkludert.

Contents

List of Figures

List of Tables

1 Introduction

Demands from the oil sector have been increasing, what was a huge gain several years ago might not even be worth to develop right now. Even more so for mature fields where the recovery grows smaller with each passing day. Enhanced oil recovery methods are required to deal with those demands, injection of water, brine, or even gas, to push the more oil out and faster. Some have to resort to fracturing, injection of surfactant and others to polymer flooding and gel conformance treatments.

1.1 Why polymer flooding?

This topic was selected because of the possibility to ascertain, to a good degree of agreement between simulations and field experiences, how the polymer solution behaves and enhances production. Understanding the dimensions of the field, properties of the reservoir and how the parameters affect the fluid and rock properties opens up for a whole new sort of alternatives in choosing how to produce the fields. Where recovery enhanced by this method is directly related to that understanding of the aforementioned.

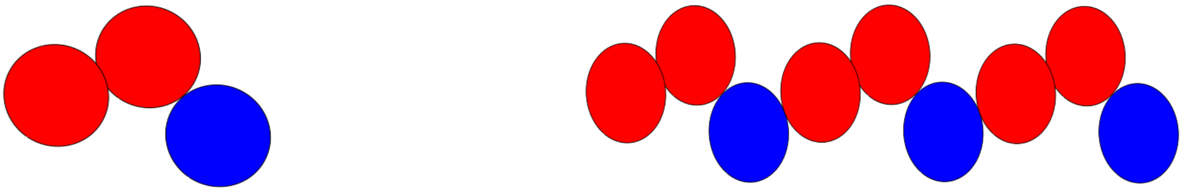
The work presented here extends from the basic understanding of what polymer is, what affects it and why, through what the literature has found about it and what makes it a reasonable solution to the problems as well as how reliable it is. An effort is also done in trying to predict how it interacts in different scenarios inside a reservoir.

1.2 Structure of the work

At the start is the introduction, which include a brief introduction of the need of polymer flooding and why it is interesting. In addition, the structure of the work presented. After that comes a description of what polymer, polymer flooding and polymer gel is, introducing factors that describe their behavior. It is followed by lab and field studies reporting how different selections of polymer and polymer gel systems vary with different parameters. Finally, a theoretical review of porous media is extended to three dimensional rectangular coordinates to cope with vertical flow between different vertical sections in the reservoir and fractional flow simulations are carried out in Matlab and compared to simulations in Eclipse, one of the more common software used in reservoir simulation.

2 Polymer

A polymer is a chain of linked monomers.“The polymer should have a molecular weight greater than 200 and at least 8 or more repeating units” (Clark, 1984). The types of monomers conforming it, their arrangement as well as the length of the chain then decide the properties of the polymer. Classified as either synthetic polymers or biopolymers. While biopolymers viscosify better in high salinity waters, Polyacrylamides, a type of synthetic polymer, gives better results in low salinity waters (Needham & Doe, 1987). A visualization of a monomer and a polymer is presented in Fig. 1.


(a) Monomer

(b) Polymer extension

Figure 1: Visualization of (a) a monomer and (b) a very short extension of a polymer chain.

When choosing a polymer for use in polymer flooding or gel formation, the properties of interest are the weight, given by the extension of the chain and the types of monomers that constitute it, the resilience of the chain extension to external forces, experienced by the injection velocity of the flow but also to pressure differentials in the reservoir, and the viscosifying properties they carry at different temperatures. Furthermore, the polymer has to be soluble in the current solution, which is usually water. Polymers are usually non-Newtonian and are referred to as a dilatant or pseudoplastic fluids, but depending on the reservoir conditions and the amount of shear stress experienced they may behave like a Newtonian fluid (Seright, 2010).

The viscosifying properties of the polymer in a solution are known to be defined by its size and concentration, as by definition, viscosity μ is the resistance to shearing flow expressed as

$$\frac{F}{A} = \mu u_{shear}, \qquad\qquad P = \frac{ma}{A} = \mu u_{shear}, \tag{1}$$

where F is the force applied over the area A, m is the mass, a is the acceleration, P is the pressure and $\boldsymbol{u_{shear}}$ is the shear velocity. Moreover, considering a constant acceleration, the mass determines how viscous the fluid is.

A unit used to quantify the polymer molecule in terms of mass is the Dalton. It is equal to $1.66x10^{-27}kg$ and is defined as ”1/12 times the mass of a free carbon 12 atom, at rest and in its ground state” (SI, 2006). It follows that the longer the polymer chain, the more components that constitute it, as consequence, more mass and thus a higher Dalton value. However, in real life field and test experiences, higher Dalton does not directly reflect higher viscosity values. These high Dalton values could be due to the polymer being composed of large and heavy monomers. As big and heavy as they may be, they would experience more resistance to flow and the bonds holding the polymer molecule as a unit could end up breaking, resulting in a considerable smaller polymer size (Yu et al., 2003) with a comparable Dalton value and effective viscosity. This mechanical degradation and is one of several ways the polymer can be deteriorated and viscosity affected that will be discussed in more detail later.

2.1 Polymer Flooding

Water flooding is the injection of an aqueous solution into a reservoir to increase production by pushing the oil out. What determines how effective this method is, is the mobility λ of the water relative to that of the oil, where the *Mobility ratio* M is defined as the mobility of the displacing fluid (water) behind the flood front to that of the displaced fluid (oil) ahead of the flood front (Baker, 1997; Dake, 1983).

$$M = \frac{M_{rw}}{M_{ro}} = \frac{k'_{r\alpha}}{\mu_\alpha}, \qquad \lambda_\alpha = \frac{k_\alpha}{\mu_\alpha}, \qquad \text{for } \alpha = w, o, \tag{2}$$

where $k'_{r\alpha}$ and μ_α are the *end point relative permeability* and *viscosity* of *phase* α, respectively.

Polymer flooding is the injection of a solution, usually water with an enhanced viscosity by means of a high molecular weight polymer concentration. This injection can affect the permeability of the formation through polymer adsorption or the entrapment of polymer in the pore conducts or holes. As such, the inflicted effects the polymer on the reservoir may be quantified by the *resistance factor* R_f and the *residual resistance factor* R_{rf}.

In the case of evaluating the difference between a water flood and a polymer flood, the resistance factor is defined as the ratio of the mobility of the water solution or solvent of the polymer solution, to the mobility of the polymer solution, in the same reservoir rock (Norman & Smith, 1999)

$$R_f = \frac{\lambda_{solvent}}{\lambda_{polymer}}. \tag{3}$$

However, in the case of evaluating the effects of the polymer solution in terms of displaced fluid, the resistance factor would be the ratio of the mobility of the displaced fluid to the mobility of the fluid doing the displacement. Which in the case of no permeability reduction, is just the ratio of viscosity of the displacing fluid to the displaced fluid (Sorbie & Seright, 1992). On the other hand, the residual resistance factor represents the change in permeability resulting from the polymer injection. Defined as the permeability of the reservoir rock before injection to the permeability after injection

$$R_{rf} = \frac{k_{before}}{k_{after}}, \tag{4}$$

where the fluid used to measure the permeability is the same in both cases, and is usually just brine.

3 Polymer Gel

Polymer gel is the result of polymers forming bonds with a crosslinker and being held together by it. These bonds form a network of strongly intraconnected polymer molecules. A crosslinker molecule is a molecule able to attach itself to a polymer molecule by being more electrically attractive than the extension replaced. In other words, it acts as a bridge by forming a stable chemical bond with the polymer molecules. Depending on the crosslinker used, a single crosslinker molecule may connect two or more polymer molecules and depending on the type of crosslinker used, polymer gels may be classified as organically or inorganically crosslinked (Al-Muntasheri et al., 2007). The crosslinking process will be covered in Section 3.2

In the injection of a polymer solution into a rock formation, the polymer network may grow big enough to clog the pore throats and prevent subsequent flow. A visualization of this interaction can be seen in Fig. 2

(a) Gel

(b) Formation

Figure 2: Visualization of (a) a small gel structure and (b) flow through the formation clogged with the gel structure.

Polymer gel may increase production in the following ways. By shutting off (a) zones already drained for oil, (b) high permeable water bearing zones and (c) zones with fractures. As such, it opens up options to selectively redirect flow. Though, in the worst case scenario, the opposite may happen, namely blocking off the access to oil bearing zones, which is why there has to be a way to reverse the gelation process.

3.1 Gel Systems

There are several ways to place the gel structure, from here on referred to as *gelant*, into the formation. The more common one is injecting the polymer solution together with the crosslinker. After some time the gelant will form, and that gelation time will depend on the type of polymer, the solution it is dissolved in, the type of crosslinker and the in-situ conditions. This system however has a weakness. Injecting the polymer solution together with the crosslinker means that the gelation process may start already before entering the formation. That means some combinations of polymer and crosslinkers may be impractical. Another way to place the gelant, in a way that overcomes the weakness of the previous one, is to inject the polymer solution first and the crosslinker after. The downside with this method is reliably injecting the crosslinker into the same place as the polymer. Even if that were the case, some polymer will not get in contact with the crosslinker as at the very least, the extremes would not connect. Aside from the aforementioned methods, Liu et al. (2004) has been testing and trying a new method based on forming the gelant prior to injection. Allowing to calmly prepare the gelant in surface facilities with the appropiate characteristics to satisfy the needs of the desired intervention. This gelant is later cut into small enough granule sizes, typical values of $2 - 5mm$ and of spherical form. These may be screened and later injected, with the size of the granules determining over which permeable zones they flow and expanding $60 - 80$ times in size when in contact with water. This way, uncertainties of the gelant forming, and the characteristics of the gel formed in-situ are overcome.

3.2 Bonding

The crosslinking processes discussed here are through (a) H-bonding, (b) covalent bonding, (c) ionic bonding, (d) bonding by exchange of ligands.

Moorhouse et al. (1998) studied the interaction between the biopolymer guar and a crosslinker based on zirconium. It was compared to the performance of the borate crosslinker. At the time of the study, for use in fracturing applications, guar and carboxymethyl hydroxypropyl guar (CMHPG) polymer was the more probable choice. As for crosslinker, the metallic choice was either boron or zirconium. Hydrogen bonding was postulated to exist for non-ionic guars with the zirconium chelate in the pH range 4-10. For ionic guar however, the bonding was postulated to be covalent and to occur between ph 8-11. It was also believed that the guar formed hydrogen bonds with the borate molecule, and unlike the hydrogen bonding of the zirconium, when broken,these would rapidly heal (Moorhouse et al., 1998).

Based on this, the author believes gelation at some point relied primarily on the weak hydrogen bondings. And if the guar indeed forms hydrogen bonds with the borate crosslinker, the presented plots indicate that it is feasible to stop the flow using a gelant that is held together with hydrogen bonds, in this case, at $pH = 11$ (Moorhouse et al., 1998, See: Plot 7).
Unfortunately, no other records of hydrogen bonding as the main bonding type between polymer and crosslinker were found so validity of the postulate on hydrogen bonding is raised.

Hardy et al. (1999) studied the performance of polyacrilamide t-butyl acrylate (PAtBA) polymer with polyethylene imine (PEI) as crosslinker under different conditions. They found a crosslinking reaction between PAtBA and PEI that does not require hydrolysys or thermolysis and would still form a stable covalent bond them. This makes it possible to form a stable gel, regardless of the hydrolysis degree and independent of high temperatures, which an increase of, would open up linking sites for metallic crosslinkers. This type of bonding is common in organically crosslinked polymers, which are the preferred gel system at high temperatures because the covalent bonding holds the gel stable (Al-Muntasheri et al., 2007; Moradi-Araghi, 1999). But has also been reported to to be a viable choice at low to very low temperatures by adjusting salt concentrations (Reddy et al., 2012)

Unlike organically crosslinked polymers, inorganically and metal based crosslinker experience a ionic bonding type with the polymer. This would happen between a trivalent cation present in a Chromium (C_r^{+3}) based crosslinker and a carboxylate (R-COO$^-$) extension of a partially hydrolyzed polyacrylamide (PHPA) (Al-Muntasheri et al., 2007). And regarding the stability of the bonding, for the case of metal based crosslinker, better at lower temperatures (Moradi-Araghi, 1999).

Moradi-Araghi (1999) and Lockhart (1994) discuss the possibility of ligands extensions on metallic crosslinkers as an alternative to the simple soluble metal crosslinker. The metal being initially attached to a ligand would delay the rate of reaction between the polymer and the cation as it has to be separated from the ligand before any reaction with the polymer takes place. This introduces selectivity in the crosslinking process (Moradi-Araghi, 1999; Lockhart, 1994). Basically the ligand acts as a screening factor. Consider the Chromium based crosslinker (C_r^{+3}) screened with acetate ($CH_3CO_2{}^-$) ligands. With a PHPA polymer, a ligands exchange would take place. For gelation to occur, the carboxylate of the polymer would instead need to replace the acetate. Which being a step that previously did not exist, already delays the process, but is reported to increase the ph range for which gelation may occur. And if a more stable ligand is present, “that” ligand would react with the C_r^{+3}, leaving gelation for when “that” type of ligand has been exhasuted, making the gelation process reversible, as injecting a more stable ligand would revert the gelation process (Lockhart, 1994).

Gelation speed, conditions for gelation to start and strength of the gel structure is given by the type of bond and the surrounding, in terms of salinity, pH, amount of polymer, amount of crosslinker, other species (ligands) and temperature. Basically, the bonds and the types of polymer and crosslinker react differently to these.
Hydrolysis is the degradation of a molecule by water. For a PHPA polymer, the degree of hydrolyzation is the degree to which the amine groups (R-NH_2CO) of the polyacrylamide (PA) transforms to $R\text{-}COO^-$. The more $R\text{-}COO^-$ extensions the PHPA has, the more scusceptible to crosslinking and gelation. For the polymer to undergo hydrolysis, it has to be dissolved in an alkaline solution, usually concentrated with sodium hydroxide (NAOH). Thermolysis however, achieves the same end result but by being exposed to high enough temperatures (Borling et al., 1994).

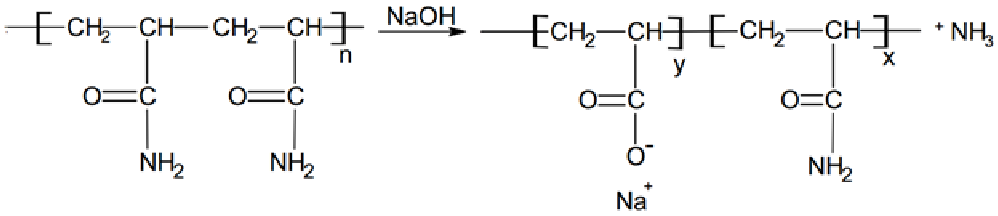


Figure 3: Alkaline hydrolysis of amide groups (Al-Muntasheri et al., 2007)

3.3 Salinity and pH

Salinity is the amount of salts dissolved in an aqueous solution, in other words its concentration. Salts being ionic compounds dissolve as ions of a positive and negative charge. In an aqueous solution, as long as salts dissolve in the solution, varying the concentration will affect the pH of the solution. This happens because salts alter the balance between the hydrogen ions (H^+) and hydroxide anions (OH^-) of the solution when dissolving.

Salinity's importance with relation to gel systems lies in the shielding experienced by carboxylate or negative charged extensions of the polymer by positively charged ions. As these extensions are the ones responsible for gelation, any way to prevent them from reaching the crosslinker would result in a delay of the gelation process. In a polymer solution with some concentration of sodium hydroxide, the dissolved sodium (Na^+) would find its way, due to the nature of the charges involved, and rest in the surroundings of the negative charged extensions of the polymer. These extensions are believed to repel each other, resulting in extended polymer chains with easy access points for the crosslinker on its sides but would, with the presence of positive charges surrounding them, result in a formless, bent and crouched polymer with hard to access extensions. As a result, the induction point for gelation is delayed (Al-Muntasheri et al., 2007; Romero-Zeron et al., 2004; Hardy et al., 1999; Reddy et al., 2003, 2012). Aside from a delay in gelation, one can distinguish a peculiar steepness from these studies. One possible explanation for it is the solution being at equilibrium regarding its charge, followed by the gelation process occurring nearly simultaneous throughout the solution.

PH influences the gel characteristics and the process of gelation similar to how the salinity does. This factor accounts for the concentration of hydrogen ions $[H^+]$ in the solution. So depending on the salts dissolved in the solution this value can shift considerably. A gelant of a polyacrylamide polymer is heavily reliant on the pH of the solution as it depends on the hydroxide $[OH^-]$ to undergo hydrolysis. In which case hydrolysis would happen faster if the solution were to be more alkaline. That would result in an increase of the extensions available for bonding of the polymer, which the crosslinker can adhere to at a given time.

4 Polymer flooding - Lab studies

A general idea of what polymers are, the viscosity effects they have in solutions, as well as the potential residual in the reservoir has been covered in Section 2. As to quantify these effects and find out to what degree they vary to a variation of different coefficients, studies have been carried out and a small fraction is gathered here. Studies that report on how much the viscosity varies from polymer to polymer and how resilient they are to changes in the surroundings, that can and are encountered in different reservoirs around the world.

4.1 Viscosity vs Shear velocity

Of the more general characteristics of polymers, a desire to know or confirmation about how viscosity varies with shear velocity is pursued. This dependency tells us about the mobility in different sections of the reservoir and distinguishes zones that are depleted slowly, more closely to what a recovery by water injection than by polymer flooding would be. Baijal (1975) compared the depency of three different types of polymer to flow rate, calculated through the R_f, one of them being Polyacrilamide. His results show how each of them, with a similar slope, decreased with increasing flow rate, identifying a shear thinning behavior for all of the experimented polymers. Surpisingly, his calculations also indicate an increase on the effect of resistance factor with the distance of polymer molecules from the surface, as these results indicate that thicker polymer structures have a better effect on viscosity. Which would explain why bigger monomers suffer from higher mechanical degradation (Yu et al., 2003). His shear study results have later been corroborated by Yerramilli et al. (2013) making use of experimental rheological measures and a model, based on the Carreau viscosity model and extended to dependency on salinity and polymer concentration.

4.2 Viscosity vs Concentration

From the results presented in Yerramilli et al. (2013) one can conclude that higher concentrations of polymer turn the solution more viscous but also that the more concetrated solution responds more to an increase in shear rate. The lower end of the concentrations tested, 250 $ppm - 500\ ppm$ behaved mostly Newtonian upto a shear rate of $10^2\ s^{-1}$ as opposed to values greater than 500 *ppm*. The effect of non-Newtonian behavior of the polymer solution can be discerned to start appearing at lower shear rate values as the concentration of polymer increases. In other words the shear rate range for which their Newtonian behavior was present decreased with increasing concentration. The limit, for his particular polymer, was encountered at 5000 *ppm* which coincided with the highest concentration tested.

These results indicate that there is a critical point in the relation between shear rate and polymer concentration where the viscosity behavior of the polymer changes from independent to shear rate to dependent on it. [1]

[1] referred to in Section 6.2

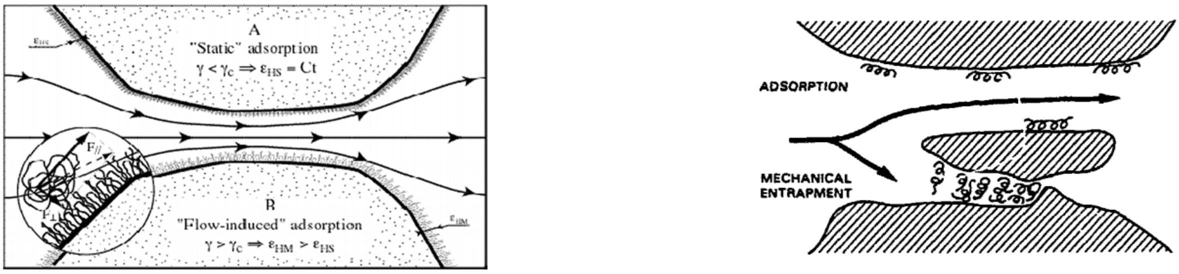
4.3 Viscosity vs Salinity

As for salinity concentration present in the solution, an increase in salinity concentration decreases the viscosity of the polymer solution Yerramilli et al. (2013). But the interaction is more interesting than just a straightforward decrease. The interaction between the polymer and salt ions in the formation of gellant described in Section 3.3, namely screening, also applies to polymer flooding and is the reason viscosity decreases, according to Yerramilli et al. (2013). Screening which has the effect of reducing the size of the polymer molecules which in accordance with the secondary results of Baijal (1975) should result in a reduced viscosity. And is exactly what the results indicate. And as is implied, by the decrease in size of the polymer molecules, the effects of shear thinning behavior with increasing shear rate ends up being less pronounced.

4.4 Adsorption

Adsorption is the adhesion of molecules to a surface, in this case, adhesion of polymer molecules to the reservoir rock surface as it flows through the pores. The surface in this case is the edge of the the boundary between the rock and the fluid in the proximity. This surface is "capable of adsorbing foreign atoms or molecules" Christmann (2012) as it outhermost bonds, on the edge, are unsaturated and thus unstable or open for exchange. For polymers the adsorption process is a physical process and the attraction to the surface mainly happens over van der Waals forces Christmann (2012); Holmen (2011) and Hydrogen bonding Yerramilli et al. (2013). The attraction forces may be very small and cover a short vertical lenght over the surface extension but the effects are increased, the higher the surface area is, encountered as the permeability decreases with a constant porosity value or as explained by Dang et al. (2011), "fine grained sands adsorb much more polymer than coarse grained sands". The reason adsorption plays a big role in polymer flooding is because a retention of polymer in the rock surface results in a decrease of effective radius for flow, in other words a decrease in permeability. This effect is accounted for with the residual resistance factor. A reduced permeability has the effect of a local increase in flow rate and hence shear rate but also the loss of polymer and subsequent lower concentration of the flowing polymer solution that leads to a weakened viscosity. Furthermore, flow-induced adsorption han an increased thickness of the adsorbed layer, compared to static adsorption Dang et al. (2011); Chauveteau et al. (2002). And as with the relation between surface area and permeability in mind, adsorption could potentially block off low permeability paths that may be rich in oil, confirmed by Zitha (2001). This phenomenom is explained by the polymer molecules stretching, increasing in length and as they are adsorbed, by being potentially equal or bigger to the smaller pore diameters in the vicinity, end up blocking them off. The Langmuir isotherm is used as base for calculating the amount of adsorption Yerramilli et al. (2013); Zitha (2001), which means that temperature is a variable that plays a big role, in terms of activation energy required for reactions to happen.

Dang et al. (2011) investigated the adsorption response of HPAM and recommends the use of high molecular weight polyacrilamides to reduce polymer adsorption as their results indicate that the adsorption mass increases with decreasing molecular weight. This may sound like a favorable course of action but there are several ways that the permeability can



(a) Adsorption

(b) Mechanical Entrapment

Figure 4: Different retention mechanisms of polymer molecules, (a) static and flow-induced adsorption (b) mechanical entrapment, Chauveteau et al. (2002); Huh et al. (1990).

be affected. One of them being mechanical entrapment. Polymer retention by mechanical entrapment would increase with molecular weight, as the bigger the polymer molecules the harder it is for those molecules to flow through the pores, specifically pore throats Huh et al. (1990). A visualization of these effects can be seen in Fig. 4. And coupled with the fact that adsorption increases with polymer concentration Zhang & Seright (2014), Fig. 5 one can conclude that polymer retention by adsorption increases with decreasing polymer molecule size and increasing concentration.

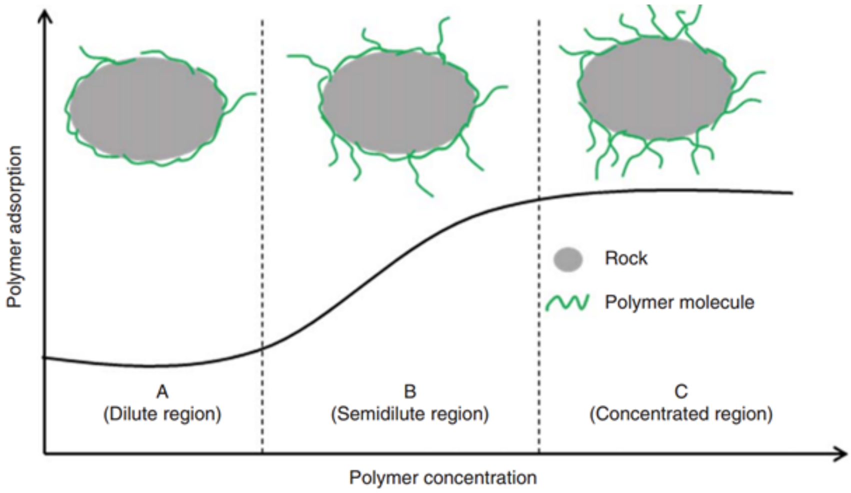


Figure 5: Proposed polymer-adsorption mechanism on the rock surface, Zhang & Seright (2014).

As for increasing pH, for the particular case of HPAM as the polymer of choice, Dang et al. (2011) reports that the adsortion decreases and discusses about the reasons for this happening, attributing it to the reduced screening effects by the higher amount of hydroxyls present in higher pH values which results in an increase in polymer molecule. Which as discussed earlier has a lower adsorption value compared to small molecules. Thus, the negatively charged rock surface, by excess of hydroxyls present in the water that attach to the rock surface, repel the vast amount of negatively charged carboxyls of the HPAM. And much like with pH but with the contrary effect, an increase in salinity results in a decrease of what would be the present hydroxyl which is reflected in smaller polymer molecule sizes and consequently higher adsorption value.

4.5 Other flow rate effects

An increase in polymer flow rate is often represented by an increase in shear rate. This increase, as presented by Ogunberu & Asghari (2004) reflects an increase the adsorption and hence reduces the RRF. This effect is desirable but the effects of the reduced permeability can be negated if the subsequent brine or polymer solution is injected at a too high rate, in other words too high shear rate. The RRF will experience an increase to a peek, at a critical shear rate. For the studies performed by Ogunberu & Asghari (2004) that peek will be around an injection rate of 1.5 $\frac{mL}{min}$ of brine and then progressively diminish with increasing shear rate. What makes this permeability reduction so valuable is that the relative permeability of water is considerably reduced compared to the relative permeability of oil, in other words, its the injectant flow volume that is reduced (Ogunberu & Asghari, 2004). This is confirmed, attributing it to the flow of water and oil being transported on separate networks and as a result, only the aqueous phase experiencing the effects of the adsorption, (Schneider & Owens, 1982). Where the polymer is in contact with the walls of the formation as opposed to the the oil phase. This thoughts and results are confirmed and corroborated by (Zaitoun & Bertin, 1998) making a distinction for water-wet cores and oil-wet cores, where the result is the same, relative permeability of water is reduced. What is interesting though is that the results imply that adsorption in the oil-wet core formation reduces the residual oil saturation. In theory this is possible but no other author has corroborated this statement.

4.6 Mechanical Degradation

Mechanical degradation is the act of breaking of the structure by physical forces, for a polymer solution in a rock formation this may experienced by the shear forces between them. The flow near the injection well is referred to as extensional flow. As stated by Argillier et al. (2013), a higher degree degraded polymer solution experiences lower extensional pressure differentials than less degraded polymers. It is also shown how the apparent relative viscosity after decreasing with interstitial velocity, at some point slightly increases, the effect or rheo-tickening. This effect is negated as the degradation in a polymer increases, meaning that using an already degraded polymer, can increase the injectivity of polymer. Good in the cases where a degradation does not alter or reduce the viscosity too much, which is the case of the test polymer HPAM 360S, Argillier et al. (2013). Mechanical degradation is believed to affect the viscosity properties of a polymer as molecules are separated. Seright & Seheult (2008) reports that xanthan is a good resistant to shear stress, which being a polymer that exhibits pseudoplastic behavior would be really bad for conformance problems. In the tests performed, its viscosity decreased evenly and no effect of degradation was seen. Tests for types of HPAM were also conducted. As it was exposed to shear stress, it maintained its viscosity properties at low values and then exhibited pseudoplastic behavior until it reached the point of mechanical degradation where the viscosity decreased even more. The types tested experienced a viscosity loss of 15% to 64% at a forced rate of $300m/day$.

4.7 Polymer flooding - Field studies

A field case is presented in de Melo et al. (2005). Some details of it will be included here to show and discuss what is experienced in the process of polymer flooding. The polymer flooding project is situated in Brazil where the polymer of choice was the HPAM. Tests were conducted to evaluate the proper level of hydrolyzation, and MW of the polymer based on the reservoir properties. Since the HPAM polymer is of pseudoplastic rheological behavior, where the viscosity decreases with increasing shear rate, tests had to be performed with representative velocities that the polymer would experience in the reservoir. This value was considered low, representative of a low flow velocity and was equal to $7.3s^{-1}$. The studies returned a linear increase of viscosity with polymer concentration, under the representative shear stress and at standard temperature, with distilled water. Under reservoir conditions however, the viscosity values calculated decreased considerably, and the increment with polymer concentration were more similar with the power function of the concentration than linearly with the concentration. Meaning that the high temperatures in the reservoir had a great impact on the viscosity. When the injection process began, at times the polymer concentrations varied, a corresponding variation was recorded for the pressure. This meant that the polymer was working as intended. The first from the three fields was treated for 4 years, the results varied. Less than 30% of the wells presented positive results. The results were considered modest at most, by injecting a total of 1.1 pore volume and producing only $\approx 8250m^3$, only 2.8% of the oil in place at the start of the project. A couple of reasons were given for these results. One of them is that the area to be depleted was open, meaning that both oil and injection fluid could have moved out instead of towards the production wells, which seems reasonable. Another reason was given to the loss of polymer by adsorption and the salinity of the formation water. The latter also seems plausible as the rate of which the polymer solution was injected was very low, pore volume wise considering it took 4 years to inject 1.1 pore volume. Considering this and the fact that saturation of formation water was high at the start of the injection, means that it is less probable for the polymer to retain its viscosity properties than its viscosity decreasing by the formation water. Aside from this, the polymer solution to oil ratio was not very high, ≈ 4. With a high initial water saturation it is unrealistic to expect much from a polymer to oil viscosity ratio of 4 for low injection velocities. The other two fields returned good results. The author of de Melo et al. (2005) also states that enhanced oil recovery (EOR) methods, polymer flooding being one of them, should be employed as soon as possible.

5 Polymer Gel - Lab studies

The polymer properties were described in Section 4. However, an increase in polymer viscosity does not reflect a good and sturdy gel structure. It is the relation between the polymer and the crosslinker, experienced at different reservoir conditions. As such, gel strength has been proved to increase with polymer molecular weight (Sydansk, 1988).

5.1 HPAM and Chromium

The polymer-crosslinker system studies consisted of HPAM and C_r^{+3}. Contrary to what is expected for polymer flooding, an increase in the concentration of the polymer while maintaining a constant and low crosslinker concentration reduces the gel strength. This happens because polymer molecules are less connected to each other as a whole structure and instead, loose strings of polymer gel connections are formed. Oppositely to this though, an increasing crosslinker to polymer ratio has been seen to increase the strength of the gel structure Sydansk (1988). However, this may only happen when there is vast amount of crosslinker suitable connecting extensions on the selected polymer. HPAM is known to be able to produce more of these extensions by increasing the hydrolization factor. However, this might not be the appropriate way to increase the gelation strength, Sydansk (1988) notes that an increase in the crosslinker to polymer rate might be damaging since it promotes syneresis, which is the expulsion of water and causes shrinking of the gel structure. As a result, less area is covered and the gel structure could start flowing, moving from its position, and even exit the formation instead of being stuck and stopping the flow at that particular position.

PH has been previously mentioned to affect the crosslinking process. This system too experiences some changes as pH varies Sydansk (1988), mainly because it consists of a HPAM polymer, where pH affects the hydrolysis of the solution. The gelation rate and gel structure strength for this system increases with increasing pH in the range $pH = 7 - 10$. Additionally, this system is highly susceptible to high temperatures. The gelation time at $60°C$ is in the range of an hour with a stabilized maximum gel strength as time passes. Lower temperatures increase the gelation rate gradually, starting at $20°C$ and closing in on the asymptote of maximum gel strength. Furthermore, the strength of the gel structure is considerably lower after 20+ hours for a temperature of $T < 30°C$, the highest reached at 1/4 the total strength the $60°C$ test reached in an hour. As to how to proceed, following the calculations for optimal gel placement, Sydansk (1988) states that increasing or decreasing the gelation rate by using accelerators/decelerators, chemicals which make the binding of the polymer and crosslinker a faster or slower process, is more attractive operationally and economically than adjusting parameters like pH or temperature. For this particular case, percentage-wise an increase in accelerator concentration displays bigger changes than an increase in crosslinker concentration in terms of gel strength and the time it takes to achieve it Fig. 6. This shows that accelerators have a bigger impact on gelation rate and gel structure strength. Concluding that temperature is more of a factor giving initial conditions, which are to be satisfied by a specific polymer crosslinker system and later adjusted than choosing a combination satisfying the time requirements and adapting it to the reservoir conditions so as to increase its strength. As for the pH value, most polymer solutions, as needed to be injected in large quantities, are produced in a big scale, making it harder for smaller fields or mature fields to ask for specific pH or salt concentration specifications. That would require developing and adapt a specific polymer solution for every field, where a post addition of accelerators is more reasonable.

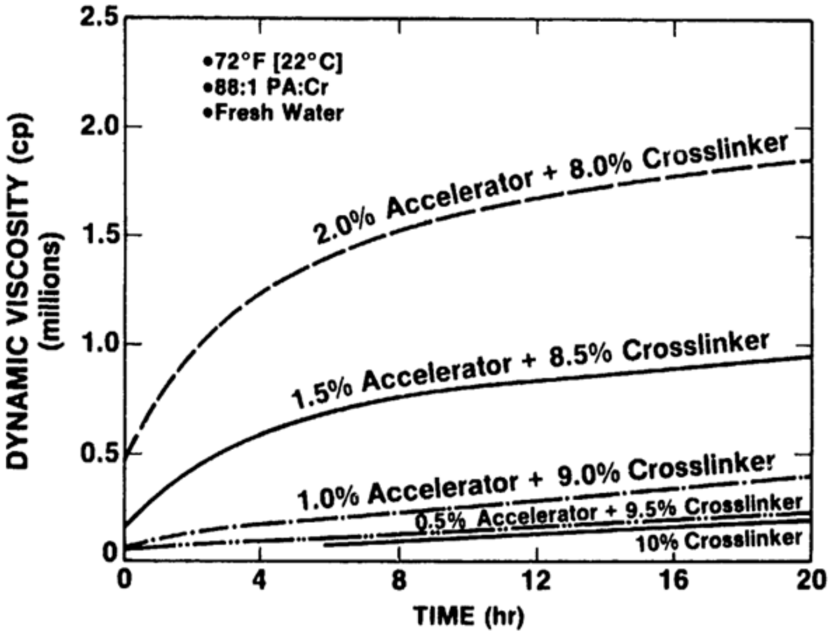


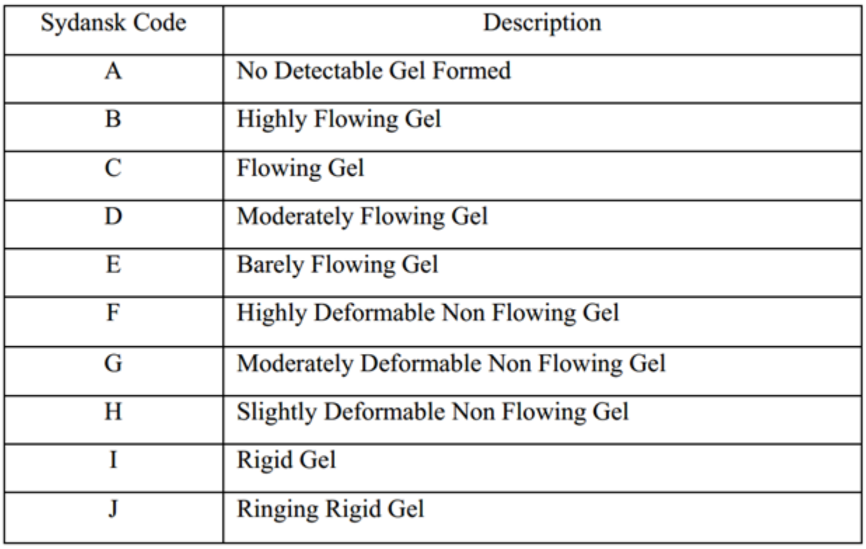
Figure 6: Gel strength for increasing accelerator and crosslinker concentration, Sydansk (1988).

5.2 System at low Temperature

Reddy et al. (2012) studied gelation in low temperature conditions. HPAM and C_r^{+3} is one of the most common systems for gel formation and as we already discussed, very bad at forming a fast or sturdy structure at low temperatures, $T \approx 20°C$. The system tested here, consisting of PHPA polymer, not that different from the fully or more hydrolyzed HPAM but with PEI as crosslinker instead. The test results show that gel at $7°C$ starts forming around $t = 12$ hours and the strength of the structure drastically increases, growing stronger with time. At 20 hours the apparent viscosity calculated equaled $20000cp$, viscosity for which the displacement of the gel is minimal. These results indicate that this system allows for gel placement even in cold regions like the sea in Norway. Furthermore, this gel is also good at elsewhere low temperatures of $20°C$ with a complete gelation time of less than 10 hours. Where depending on the salinity concentration of the field, these times can be even lower. A 7% active PHPA and 2% PEI in fresh water, forms a sturdy gel structure in under 3 hours in the range $7 < T < 25°C$, faster for higher temperatures. Reddy et al. (2012) also notes that it is important to consider the mixing time and the temperature over that interval, as the crosslinking process starts as soon as there is a contact between crosslinker and polymer and the temperature they are at the time. **??** mentions that surface temperatures may be significantly lower than reservoir temperature and it is easy to miscalculate the time it takes for gelation to occur if this is not taken into account. Consider this gelation time not starting until the injectant temperature is close to the reservoir temperature, which is a slow process. The gel formation would be far away from the indended position. Worth noting is the fact that an increase in polymer concentration, for this particular system, as opposed to the case presented by Sydansk (1988), reduces the gelation time, for which the strength is overall the same. Same response is obtained from increasing the polymer concentration instead. So as to show that a change a boost in parameters for a particular system does not return the same improvements for another system.

5.3 Classification of polymer gel structure

The Sydansk code was created in an intent to classify the strength and types of different polymer gel structures. It can be seen in Fig. 7 that there is a great variety of gel structures. This system has been employed by Sydansk (1988) and several other authors to date and provides an alternative way to to describe a gel structure aside from its resistance to flow or apparent viscosity.

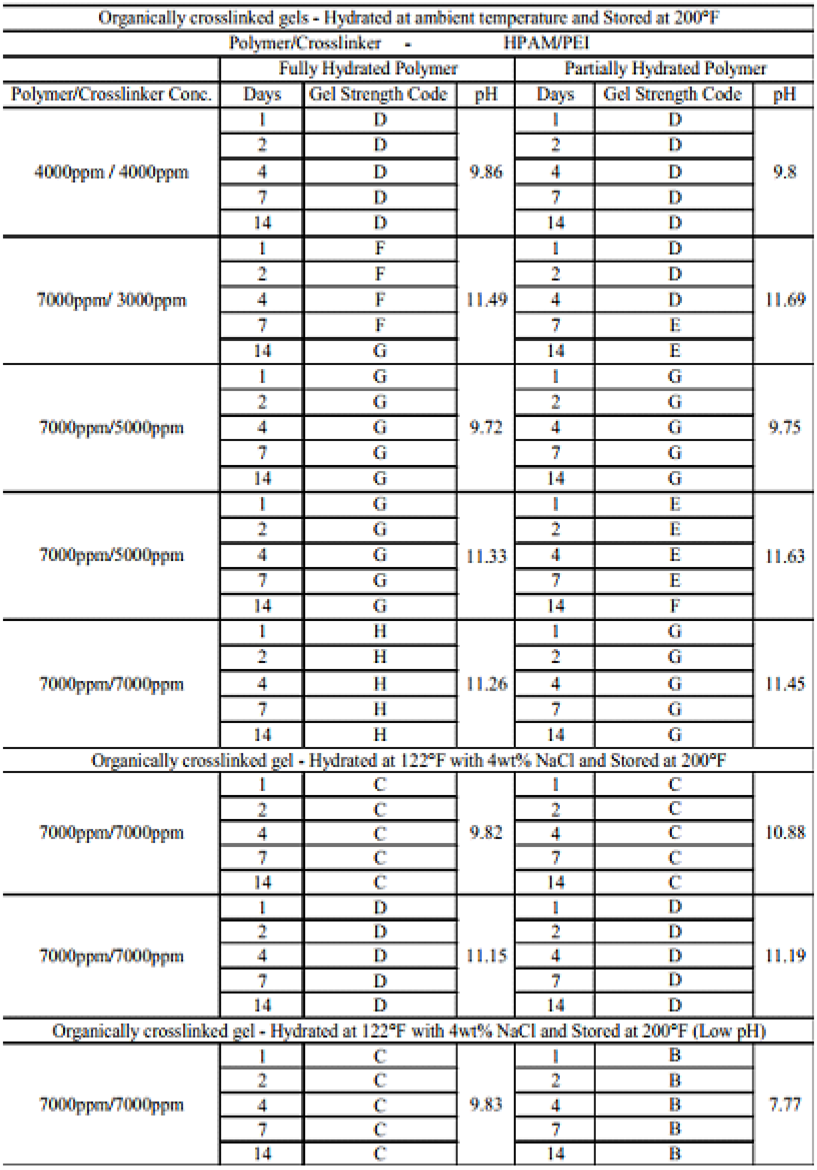


Sydansk Code	Description
A	No Detectable Gel Formed
B	Highly Flowing Gel
C	Flowing Gel
D	Moderately Flowing Gel
E	Barely Flowing Gel
F	Highly Deformable Non Flowing Gel
G	Moderately Deformable Non Flowing Gel
H	Slightly Deformable Non Flowing Gel
I	Rigid Gel
J	Ringing Rigid Gel

Figure 7: Sydansk code, (Sydansk, 2007).

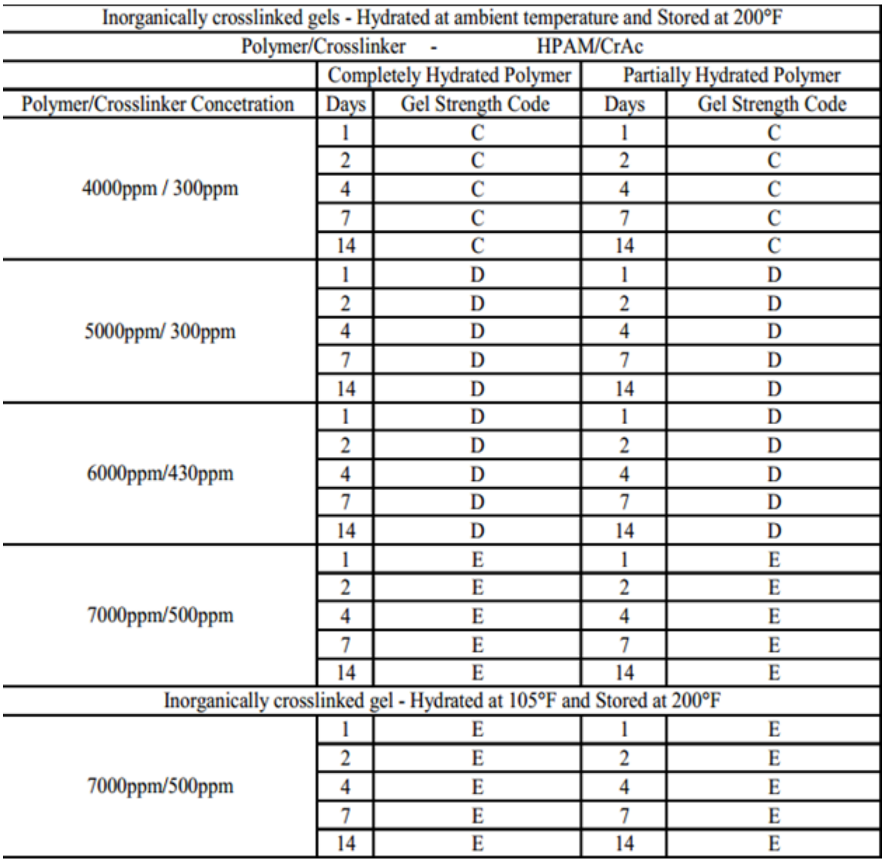
HPAM, PEI and Chromium connected to thre acetate, $C_r(CH_3CO_2)_3$ are all widely used polymer and crosslinker for polymer flooding. Specifically the combination of HPAM with $C_r(CH_3CO_2)_3$ and alternatively HPAM with the inorganically PEI. Shriwal & Lane (2012) conveniently constructed a list showing the gel strength of the system at different parameters. Fig. 8 shows the strength of the gel formed from an organically crosslinked system composed of HPAM polymer and PEI crosslinker formed at different pH values, temperature, polymer and crosslinker concentrations and hydrolization over the course of several days. While, Fig. 9 shows the strength of the gel formed from an inorganically crosslinked system composed HPAM polymer and $C_r(CH_3CO_2)_3$ formed at different temperatures, polymer and crosslinker concentrations and hydrolization of the polymer over the course of several days.

In addition, Fig. 10 is presented to give an impression on the polymer concentration and volume of gel that is required to solve a variety of possible scenarios that require the use of polymer gel, based on actual field cases (Lantz & Muniz, 2014).



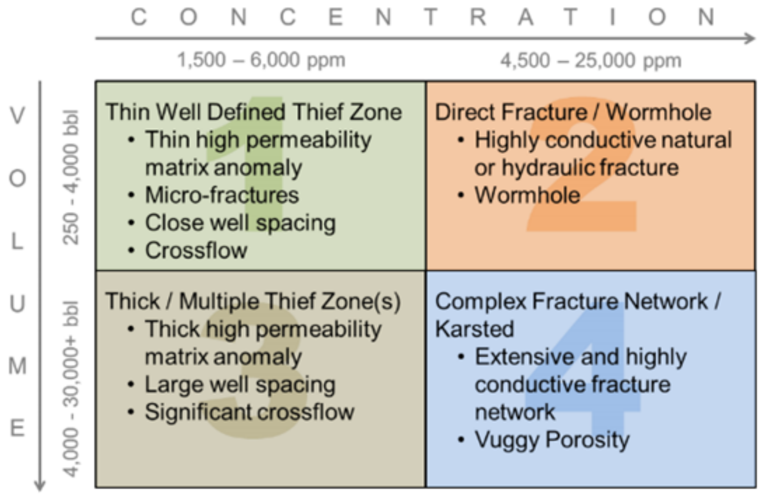
Organically crosslinked gels - Hydrated at ambient temperature and Stored at 200°F						
Polymer/Crosslinker - HPAM/PEI						
	Fully Hydrated Polymer			Partially Hydrated Polymer		
Polymer/Crosslinker Conc.	Days	Gel Strength Code	pH	Days	Gel Strength Code	pH
4000ppm / 4000ppm	1	D	9.86	1	D	9.8
	2	D		2	D	
	4	D		4	D	
	7	D		7	D	
	14	D		14	D	
7000ppm/ 3000ppm	1	F	11.49	1	D	11.69
	2	F		2	D	
	4	F		4	D	
	7	F		7	E	
	14	G		14	E	
7000ppm/5000ppm	1	G	9.72	1	G	9.75
	2	G		2	G	
	4	G		4	G	
	7	G		7	G	
	14	G		14	G	
7000ppm/5000ppm	1	G	11.33	1	E	11.63
	2	G		2	E	
	4	G		4	E	
	7	G		7	E	
	14	G		14	F	
7000ppm/7000ppm	1	H	11.26	1	G	11.45
	2	H		2	G	
	4	H		4	G	
	7	H		7	G	
	14	H		14	G	
Organically crosslinked gel - Hydrated at 122°F with 4wt% NaCl and Stored at 200°F						
7000ppm/7000ppm	1	C	9.82	1	C	10.88
	2	C		2	C	
	4	C		4	C	
	7	C		7	C	
	14	C		14	C	
7000ppm/7000ppm	1	D	11.15	1	D	11.19
	2	D		2	D	
	4	D		4	D	
	7	D		7	D	
	14	D		14	D	
Organically crosslinked gel - Hydrated at 122°F with 4wt% NaCl and Stored at 200°F (Low pH)						
7000ppm/7000ppm	1	C	9.83	1	B	7.77
	2	C		2	B	
	4	C		4	B	
	7	C		7	B	
	14	C		14	B	

Figure 8: Gel codes for organically crosslinked gels with fresh water, HPAM/PEI system, (Shriwal & Lane, 2012).



Inorganically crosslinked gels - Hydrated at ambient temperature and Stored at 200°F				
Polymer/Crosslinker -	HPAM/CrAc			
	Completely Hydrated Polymer		Partially Hydrated Polymer	
Polymer/Crosslinker Concetration	Days	Gel Strength Code	Days	Gel Strength Code
4000ppm / 300ppm	1	C	1	C
	2	C	2	C
	4	C	4	C
	7	C	7	C
	14	C	14	C
5000ppm/ 300ppm	1	D	1	D
	2	D	2	D
	4	D	4	D
	7	D	7	D
	14	D	14	D
6000ppm/430ppm	1	D	1	D
	2	D	2	D
	4	D	4	D
	7	D	7	D
	14	D	14	D
7000ppm/500ppm	1	E	1	E
	2	E	2	E
	4	E	4	E
	7	E	7	E
	14	E	14	E
Inorganically crosslinked gel - Hydrated at 105°F and Stored at 200°F				
7000ppm/500ppm	1	E	1	E
	2	E	2	E
	4	E	4	E
	7	E	7	E
	14	E	14	E

Figure 9: Gel codes for inorganically crosslinked gels with fresh water, HPAM/$C_r(CH_3CO_2)_3$ system (Shriwal & Lane, 2012).



Thin Well Defined Thief Zone
- Thin high permeability matrix anomaly
- Micro-fractures
- Close well spacing
- Crossflow

Direct Fracture / Wormhole
- Highly conductive natural or hydraulic fracture
- Wormhole

Thick / Multiple Thief Zone(s)
- Thick high permeability matrix anomaly
- Large well spacing
- Significant crossflow

Complex Fracture Network / Karsted
- Extensive and highly conductive fracture network
- Vuggy Porosity

Figure 10: Polymer Gel Injection Well Conformance Improvement Matrix (Lantz & Muniz, 2014).

5.4 Polymer Gel - Field studies

In this subsection a field case report will be examined to visualize the process of gel treatment problems that may be encountered. The field case was presented by Liu et al. (2004) and is located in the daqing oil field in China. They have employed a new type of gel formation process, consisting of the placement of the gel in the reservoir after the gel is formed. Once placed, the gel expands, forming a gel structure no different than other gel structures. This gel is stated to last for at least two years at room conditions. First of all simulations were carried out, these were carried out in a $20x20x5$ grid. These simulations calculated that the required injection volume would be $3000m^3$ for an increase in oil production of $2010m^3$. As for the results, they started with positive indications, the pressure increased from before to after the treatmeant, meaning that the gel was working as intended. Following the treatment, the blocking of the needed areas to increase sweep efficiency, the injection rate was kept constant. The wells expectedly started to increase production and in the interval of 7 months, the amount produced exceeded the amount indicated by the simulation results. Not much information is shared regarding the calculations of the placement of these granular, the needed volume of each, based on heterogeneity and permeability values other than that they vary from $1-5mm$. But it works, and the formation of the gel granular avoids the operation problems of having too many variables to adhere to, to place the gel fluid, satisfy the strength requirements and see that it does not grow weaker. The granule sized polymer-crosslinker gel is formed above ground, at favorable conditions, and are not affected by sanility concentration as they are screened beforehand, same reason for which pH does not influence it. A rough structure that is stable, small in size, easy to manage and able to cope with temperatures as high as $100°C$, (Coste et al., 2000).

6 Derivation of the Flow Equation

Kleppe (2015) covers the basic derivation of the flow equation in his handouts. It assumes no isothermal conditions to disregard the energy conservation equation and results in the derivation of the mass and momentum conservation equations.

6.1 Conservation of mass

Consider a volume given by length Δx as described by Fig. 11

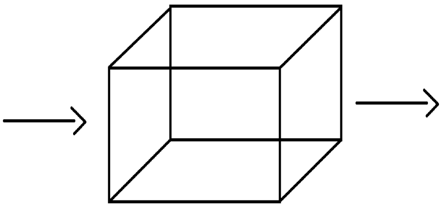


Figure 11: Box of length Δx

Conservation of mass tells us that the rate of mass on that volume is given by the mass that goes in minus the mass that goes out.

$$\{Mass_{in}\} - \{Mass_{out}\} = \{Mass_{Rate}\} \tag{5}$$

So that

$$\{u\rho A\}_x - \{\rho A\}_{x+\Delta x} = \frac{\partial}{\partial t}\{\phi\rho A\Delta x\}, \tag{6}$$

where ϕ is the porosity and ρ is the density. Which dividing by Δx and assuming a contant area A results in

$$-\frac{\partial}{\partial x}(u\rho) = \frac{\partial}{\partial t}(\phi\rho) \tag{7}$$

6.2 Conservation of momentum

Still following Kleppe's (2015) derivation, the conservation of momentum is reduced to the semi-empirical Darcy's equation for low velocity flow in porous media.

$$u = -\frac{k}{\mu}\frac{\partial P}{\partial x}. \tag{8}$$

Altervatively, the Forcheimer's equation can be used for higher velocities

$$-\frac{\partial P}{\partial x} = u\frac{\mu}{k} + \beta u^n. \tag{9}$$

To simplify the derivation, it will be assumed that Darcy's equation is valid for the continuation of the thesis. Usually the fluid is considered Newtonian for the same reason, which by further assuming constant temperature means that the viscosity μ is constant. That may be the case for the water and oil but some polymer solutions have non-Newtonian properties and the viscosity might be shear thinning or shear thickening. This behavior can be described by the Power law equation also referred to as the Ostwald-de Waele relationship

$$\mu = \mu_0 |u|^{n-1}, \tag{10}$$

where μ_0 represents the consistency of the power law fluid and is the Newtonian viscosity of the fluid, and n describes the fluid behavior

$$n \begin{cases} 0 < n < 1 & \text{Pseudoplastic or Shear thinning} \\ = 1 & \text{Newtonian} \\ > 1 & \text{Dilatant or Shear thinckening,} \end{cases}$$

and is in concordance with what was discussed about viscosity in Section 4.2.

To complete the derivations, a pressure dependency equation is defined for the density

$$c_f = (\frac{1}{V})(\frac{\partial V}{\partial P})_T \qquad \frac{d\rho}{dP} = c_f \rho, \tag{11}$$

and for the porosity

$$c_r = (\frac{1}{\phi})(\frac{\partial \phi}{\partial P})_T \qquad \frac{d\phi}{dP} = c_r \phi. \tag{12}$$

6.3 One-Dimensional Flow Equation

Following Kleppe's (2015) assumption of a newtonian fluid and combining Eq. (7) with Eq. (8)

$$-\frac{\partial}{\partial x}\left(-\rho\frac{k}{\mu}\frac{\partial P}{\partial x}\right) = \frac{\partial}{\partial t}(\phi\rho), \tag{13}$$

and solving for the left side of the equation, with k being a composite funcion of constants for distinct displacements along x for which $\frac{\partial(k(x))}{\partial x} = 0$, for μ being a constant and making use of Eq. (11) gives

$$\begin{aligned}
&= \frac{k}{\mu}\left(\rho\frac{\partial^2 P}{\partial x^2} + \frac{\partial P}{\partial x}\frac{\partial \rho}{\partial x}\right) && (14)\\
&= \frac{k}{\mu}\left(\rho\frac{\partial^2 P}{\partial x^2} + \frac{\partial P}{\partial x}\frac{d\rho}{dP}\frac{\partial P}{\partial x}\right)\\
&= \frac{k}{\mu}\left(\rho\frac{\partial^2 P}{\partial x^2} + c_f\rho\left(\frac{\partial P}{\partial x}\right)^2\right)\\
&= \rho\frac{k}{\mu}\left(\frac{\partial^2 P}{\partial x^2} + c_f(\frac{\partial P}{\partial x})^2\right), && (15)
\end{aligned}$$

which considering c_f is small for liquids and the pressure gradient is also small because of the low velocities previously assumed, results in

$$\frac{\partial^2 P}{\partial x^2} >> c_f\left(\frac{\partial P}{\partial x}\right)^2, \tag{16}$$

which reduces to

$$= \frac{\rho k}{\mu}\left(\frac{\partial^2 P}{\partial x^2}\right). \tag{17}$$

Similarly, the expansion of the right side of Eq. (13) is presented. It makes use of Eqs. (11) and (12)

$$\begin{aligned}
&= \rho\frac{\partial \phi}{\partial t} + \phi\frac{\partial \rho}{\partial t} && (18)\\
&= \rho\frac{d\phi}{dP}\frac{\partial P}{\partial t} + \phi\frac{d\rho}{dP}\frac{\partial P}{\partial t}\\
&= (c_f + c_r)\,\rho\phi\frac{\partial P}{\partial t}\\
&= c_t\rho\phi\frac{\partial P}{\partial t}. && (19)
\end{aligned}$$

Finally, the complete equation presented by Kleppe (2015) is reduced to

$$\begin{aligned}
\rho\frac{k}{\mu}\left(\frac{\partial^2 P}{\partial x^2}\right) &= c_t\rho\phi\frac{\partial P}{\partial t}\\
\left(\frac{\partial^2 P}{\partial x^2}\right) &= \frac{\mu c_t\phi}{k}\frac{\partial P}{\partial t}. && (20)
\end{aligned}$$

For the case of Non-Newtonian fluid

As mentioned in the above section, a Non-Newtonian fluid carries around the weight of a not so simple dependency of viscosity with shear stress which is directly proportional with the velocity of the fluid as described by Eq. (10). In this case the derivation of the 1-Dimensional equation complicates itself. Combining Eq. (10) with Eq. (8) and remembering the sign only indicates the orientation of the flow,

$$u = -\frac{k}{\mu_0 |u|^{n-1}} \frac{\partial P}{\partial x} \tag{21}$$

$$u^n = -\frac{k}{\mu_0} \frac{\partial P}{\partial x} \tag{22}$$

$$u = \begin{cases} -\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1/n} & \frac{\partial P}{\partial x} < 1 \\ \left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1/n} & \frac{\partial P}{\partial x} \geq 1. \end{cases} \tag{23}$$

Inserting Eq. (23) into Eq. (7)

$$-\frac{\partial}{\partial x}\left(-\rho\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1/n}\right) = \frac{\partial}{\partial t}(\phi\rho) \tag{24}$$

and solving as for Eq. (13) we get

$$= \frac{\partial}{\partial x}\left(\rho\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1/n}\right) \tag{25}$$

$$= \frac{\partial}{\partial x}\left(\rho\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right| \left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1/n-1}\right)$$

$$= \rho\frac{k}{\mu_0}\left(\frac{\partial\left|\frac{\partial P}{\partial x}\right|}{\partial x} \left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1/n-1}\right) + \rho\left(\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right| \frac{\partial\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1/n-1}}{\partial x}\right)$$

$$= \rho\left(\frac{k}{\mu_0}\right)^{1/n}\left(\frac{\partial\left|\frac{\partial P}{\partial x}\right|}{\partial x}\left(\frac{\partial P}{\partial x}\right)^{1/n-1} + \left|\frac{\partial P}{\partial x}\right|\left(\frac{1}{n}-1\right)\left|\frac{\partial P}{\partial x}\right|^{1/n-2}\frac{\partial\left|\frac{\partial P}{\partial x}\right|}{\partial x}\right)$$

$$= \frac{1}{n}\left(\frac{k}{\mu_0}\right)^{1/n}\left(\frac{\partial P}{\partial x}\right)^{1/n-1}\frac{\partial\left|\frac{\partial P}{\partial x}\right|}{\partial x}$$

$$= \frac{\rho\frac{k}{\mu_0}\left(\frac{\partial\left|\frac{\partial P}{\partial x}\right|}{\partial x}\right)}{n\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1-1/n}}. \tag{26}$$

Alternatively, it can be expressed as follows

$$= \frac{\frac{\partial}{\partial x}\left(\rho \left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|\right)}{n\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1-1/n}} \qquad \frac{\partial P}{\partial x} < 1 \tag{27}$$

$$= \frac{\frac{\partial}{\partial x}\left(\rho \left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|\right)}{n\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1-1/n}} \qquad \frac{\partial P}{\partial x} \geq 1. \tag{28}$$

From Eqs. (20) and (25), as the right side stays the same, the complete equation for a Non-Newtonian fluid, described by (10), by pressure differential $\frac{\partial P}{\partial x}$ is

$$\frac{\rho\frac{k}{\mu_0}\left(\frac{\partial^2 P}{\partial x^2}\right)}{n\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1-1/n}} = c_t \rho \phi \frac{\partial P}{\partial t} \tag{29}$$

$$\frac{\left(\frac{\partial^2 P}{\partial x^2}\right)}{n\left|\frac{k}{\mu_0}\frac{\partial P}{\partial x}\right|^{1-1/n}} = \frac{\mu_0 c_t \phi}{k}\frac{\partial P}{\partial t} \tag{30}$$

Wu et al. (1991); Fakcharoenphol & Wu (2010) proposed the use of the modified Blake-Kozeny which makes use of an effective viscosity μ_{eff} that replaces the original μ_0 from Eq. (10), which for one-dimensional flow is defined as

$$\mu_{eff} = \frac{\mu_0}{12}\ \left(9 + \frac{3}{n}\right)^n\ (150\ k\ \phi)^{\frac{1-n}{2}}\ , \tag{31}$$

6.4 Multidimensional flow

Taking the approach of rectangular coordinates equation (7) extends to

$$-\frac{\partial}{\partial x}(u_x\rho)-\frac{\partial}{\partial y}(u_y\rho)-\frac{\partial}{\partial z}(u_z\rho)=\frac{\partial}{\partial t}(\phi\rho) \tag{32}$$

which with a divergence operator can be expressed as

$$-\nabla\cdot(\boldsymbol{u}_i\rho)=\frac{\partial}{\partial t}(\phi\rho) \tag{33}$$

where $\boldsymbol{u}$ is the velocity vector over the X-Y-Z-space.
Including gravitational effects and assuming the simple case of no inclination of the $x\times y$ extension over the horizontal, the velocity over the X-Y-Z dimension is expressed as

$$u_h=\begin{cases}-\left(\frac{k_h}{\mu_{eff}}\frac{\partial P}{\partial h}\right)^{1/n} & \frac{\partial P}{\partial h}<1\\ \left(\frac{k_h}{\mu_{eff}}\frac{\partial P}{\partial h}\right)^{1/n} & \frac{\partial P}{\partial h}\geq 1\end{cases} \qquad \text{for } h=x,y \tag{34}$$

$$u_z=\begin{cases}-\left(\frac{k_z}{\mu_{eff}}\left(\frac{\partial P}{\partial z}-\rho g\right)\right)^{1/n} & \frac{\partial P}{\partial z}<1\\ \left(\frac{k_z}{\mu_{eff}}\left(\frac{\partial P}{\partial z}-\rho g\right)\right)^{1/n} & \frac{\partial P}{\partial z}\geq 1.\end{cases} \tag{35}$$

In vector form

$$\boldsymbol{u}=\begin{cases}-\left(\frac{\boldsymbol{k}}{\mu_{eff}}\left(\nabla P-\rho g\nabla z\right)\right)^{1/n} & \nabla P<1\\ \left(\frac{\boldsymbol{k}}{\mu_{eff}}\left(\nabla P-\rho g\nabla z\right)\right)^{1/n} & \nabla P\geq 1.\end{cases} \tag{36}$$

6.5 Multiphase flow

To account for multiphase flow, some coefficients have to be adjusted. For instance, (33) has to account for the amount of phase in question. This is done including saturation S for the involved phases. For a two-phase immisible flow of an aqueous phase that can be either water or a polymer solution and a non-aqueous phase that in this case is oil the saturation is defined as follow

$$S_w + S_o = 1, \qquad \begin{aligned} w &= \text{Water} \\ o &= \text{Oil}, \end{aligned} \tag{37}$$

and consequently equation (33)

$$-\nabla \cdot (\rho_\alpha \boldsymbol{u}_\alpha) = \frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) \qquad \text{for } \alpha = w, o. \tag{38}$$

Capilary pressure can be expressed as

$$P_c = P_o - P_w. \tag{39}$$

As for saturation, permeability k now has to be redefined as a function of the phases that comprises it. In vector form

$$\boldsymbol{k}_w = \boldsymbol{k} k_{rw} \qquad \boldsymbol{k}_o = \boldsymbol{k} k_{ro}, \tag{40}$$

where $\boldsymbol{k}$ is the vector permeability, $\boldsymbol{k}_w$ is the aqueous permeability vector and $\boldsymbol{k}$ is the non-aqueos permeability vector. The effective viscosity, extended to two phase by Wu et al. (1991) is

$$\mu_{eff} = \frac{\mu_0}{12} \left(9 + \frac{3}{n}\right)^n \left(150 \ k k_{r\alpha} \ (S_\alpha) \ \phi \ (S_\alpha - S_{\alpha_{irr}})\right)^{\frac{1-n}{2}} \qquad \text{for } \alpha = w, o. \tag{41}$$

Adjusting Darcy's equation accordingly

$$\boldsymbol{u}_\alpha = \begin{cases} -\left(\frac{\boldsymbol{k}_\alpha}{\mu_{\alpha_{eff}}} (\nabla P - \rho g \nabla z)\right)^{1/n_\alpha} & \nabla P < 1 \\ \left(\frac{\boldsymbol{k}_\alpha}{\mu_{\alpha_{eff}}} (\nabla P - \rho g \nabla z)\right)^{1/n_\alpha} & \nabla P \geq 1 \end{cases} \qquad \text{for } \alpha = w, o. \tag{42}$$

6.6 IMPES Solution

IMPES stands for *implicit pressure-explicit saturation* and is one of the two ways to solve for the differential equations previously derived presented here. The other being the fractional flow method presented in the next section 6.7. This method consists of a formulation in phase pressure and saturation and has been presented by Kleppe (2015); Chen et al. (2006); Binning & Celia (1999). This formulation is obtained from combining (38) and (42). Including source and sink terms and presented for each phase the formulation is expressed as

$$\begin{cases} \nabla\cdot\left(\frac{\rho_o}{\mu_{o_0}}\boldsymbol{k}_o(\nabla P_o-\rho_o g\nabla z)\right)^{1/n_o} = \frac{\partial(\phi\rho_o S_o)}{\partial t}-q_o \\ \nabla\cdot\left(\frac{\rho_w}{\mu_{w_0}}\boldsymbol{k}_w(\nabla P_w-\rho_w g\nabla z)\right)^{1/n_w} = \frac{\partial(\phi\rho_w S_w)}{\partial t}-q_w \end{cases} \qquad \text{for } \nabla P<1, \tag{43}$$

$$\begin{cases} -\nabla\cdot\left(\frac{\rho_o}{\mu_{o_0}}\boldsymbol{k}_o(\nabla P_o-\rho_o g\nabla z)\right)^{1/n_o} = \frac{\partial(\phi\rho_o S_o)}{\partial t}-q_o \\ -\nabla\cdot\left(\frac{\rho_w}{\mu_{w_0}}\boldsymbol{k}_w(\nabla P_w-\rho_w g\nabla z)\right)^{1/n_w} = \frac{\partial(\phi\rho_w S_w)}{\partial t}-q_w \end{cases} \qquad \text{for } \nabla P\geq 1. \tag{44}$$

Usually the oil behaves newtonian and Eqs. (43) and (44) simplify with $n_o=1$. Next is substituting S_o with (37) and P_w with (39) so that

$$P_w=P_o-P_c \qquad \nabla P_w=\nabla P_o-\nabla P_c \qquad \nabla P_c=\frac{dP_c}{dS_w}\nabla S_w. \tag{45}$$

Differentiating the right side of the equation and dividing by ρ_o and ρ_w respectively gives

$$\begin{cases} \frac{1}{\rho_o}\nabla\cdot\left(\frac{\rho_o}{\mu_{o_0}}\boldsymbol{k}_o(\nabla P_o-\rho_o g\nabla z)\right) = \frac{(1-S_w)}{\rho_o}\frac{\partial(\phi\rho_o)}{\partial t}-\frac{q_o}{\rho_o} \\ +\frac{1}{\rho_w}\nabla\cdot\left(\frac{\rho_w}{\mu_{w_0}}\boldsymbol{k}_w\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)^{1/n_w} = \frac{S_w}{\rho_w}\frac{\partial(\phi\rho_w)}{\partial t}-\frac{q_w}{\rho_w} \end{cases} \qquad \text{for} \nabla P<1, \tag{46}$$

$$\begin{cases} -\frac{1}{\rho_o}\nabla\cdot\left(\frac{\rho_o}{\mu_{o_0}}\boldsymbol{k}_o(\nabla P_o-\rho_o g\nabla z)\right) = \frac{(1-S_w)}{\rho_o}\frac{\partial(\phi\rho_o)}{\partial t}-\frac{q_o}{\rho_o} \\ -\frac{1}{\rho_w}\nabla\cdot\left(\frac{\rho_w}{\mu_{w_0}}\boldsymbol{k}_w\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)^{1/n_w} = \frac{S_w}{\rho_w}\frac{\partial(\phi\rho_w)}{\partial t}-\frac{q_w}{\rho_w} \end{cases} \qquad \text{for} \nabla P\geq 1. \tag{47}$$

and adding the equations of each phase

$$\left.\begin{aligned} \frac{1}{\rho_o}\nabla\cdot\left(\frac{\rho_o}{\mu_{o_0}}\boldsymbol{k}_o(\nabla P_o-\rho_o g\nabla z)\right) \\ +\frac{1}{\rho_w}\nabla\cdot\left(\frac{\rho_w}{\mu_{w_0}}\boldsymbol{k}_w\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)^{1/n_w} \\ =\frac{(1-S_w)}{\rho_o}\frac{\partial(\phi\rho_o)}{\partial t}-\frac{q_o}{\rho_o}+\frac{S_w}{\rho_w}\frac{\partial(\phi\rho_w)}{\partial t}-\frac{q_w}{\rho_w} \end{aligned}\right\} \qquad \text{for } \nabla P<1, \tag{48}$$

$$\left.\begin{aligned} -\frac{1}{\rho_o}\nabla\cdot\left(\frac{\rho_o}{\mu_{o_0}}\boldsymbol{k}_o(\nabla P_o-\rho_o g\nabla z)\right) \\ -\frac{1}{\rho_w}\nabla\cdot\left(\frac{\rho_w}{\mu_{w_0}}\boldsymbol{k}_w\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)^{1/n_w} \\ =\frac{(1-S_w)}{\rho_o}\frac{\partial(\phi\rho_o)}{\partial t}-\frac{q_o}{\rho_o}+\frac{S_w}{\rho_w}\frac{\partial(\phi\rho_w)}{\partial t}-\frac{q_w}{\rho_w} \end{aligned}\right\}\quad \text{for } \nabla P\geq 1, \tag{49}$$

and deriving, results in

$$\left.\begin{aligned} \nabla\cdot\left(\frac{\boldsymbol{k}_o}{\mu_{o_0}}(\nabla P_o-\rho_o g\nabla z)\right) \\ +\frac{\nabla\cdot\left(\frac{\boldsymbol{k}_w}{\mu_{w_0}}\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)}{n\left(\frac{\boldsymbol{k}_w}{\mu_{w_0}}\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)^{1-1/n}} \end{aligned}\right\}=\frac{(1-S_w)}{\rho_o}\frac{\partial(\phi\rho_o)}{\partial t}-\frac{q_o}{\rho_o}+\frac{S_w}{\rho_w}\frac{\partial(\phi\rho_w)}{\partial t}-\frac{q_w}{\rho_w} \quad \text{for } \nabla P<1, \tag{50}$$

$$\left.\begin{aligned} -\nabla\cdot\left(\frac{\boldsymbol{k}_o}{\mu_{o_0}}(\nabla P_o-\rho_o g\nabla z)\right) \\ -\frac{\nabla\cdot\left(\frac{\boldsymbol{k}_w}{\mu_{w_0}}\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)}{n\left(\frac{\boldsymbol{k}_w}{\mu_{w_0}}\left(\nabla P_o-\frac{dP_c}{dS_w}\nabla S_w-\rho_w g\nabla z\right)\right)^{1-1/n}} \end{aligned}\right\}=\frac{(1-S_w)}{\rho_o}\frac{\partial(\phi\rho_o)}{\partial t}-\frac{q_o}{\rho_o}+\frac{S_w}{\rho_w}\frac{\partial(\phi\rho_w)}{\partial t}-\frac{q_w}{\rho_w} \quad \text{for } \nabla P\geq 1. \tag{51}$$

Saturation is then explicitly evaluated and is used to solve for P_o, where assuming no capillary pressure simplifies this task considerably. The calculated P_o is used together with the oil phase of $(\alpha = o)$ of Eqs. (46) and (47) to calculate a new saturation. Chen et al. (2006) notes that this equation is strongly or fully coupled, meaning too dependent on each other. To reduce coupling, one has to introduce an alternate way to calculate some of the components, independent of the others and calculated separately. Ideally, the fully coupled system would give the right and precise answer but it would resort to high processing times. Alternatively, calculating some of its components separately, and obtaining similar results for those components in simpler way but still in accordance with the overall spreading, would give faster results while sacrificing a minimal and acceptable error.

6.7 Fractional Flow Solution

An alternative to the fully coupled IMPES Method is the fractional flow formulation. Much like IMPES it is composed by a pressure and a saturation equation but unlike it, the coupling is reduced by introducing a global pressure (Chen et al., 2006; Aarnes et al., 2007). For simplicity the densities are assumed constant.

We introduce the phase mobility λ as

$$\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha} \qquad\qquad \alpha = w, o \tag{52}$$

where the total mobility is defined as

$$\lambda = \lambda_w + \lambda_o. \tag{53}$$

The fractional flow is defined as

$$f_\alpha = \frac{\lambda_\alpha}{\lambda} \qquad\qquad \alpha = w, o, \tag{54}$$

where

$$f_w + f_o = 1. \tag{55}$$

The global pressure "(Antoncev, 1972; Chavent and Jaffre, 1986)" (Chen et al., 2006) is then defined as

$$P = P_o - \int^{P_c(S_w)} f_w\left(P_c^{-1}(\epsilon)\right) d\epsilon, \tag{56}$$

defining the total velocity $\boldsymbol{u}$ as

$$\boldsymbol{u} = \boldsymbol{u}_w + \boldsymbol{u}_o, \tag{57}$$

where for simplicity and the purpose of showing a simple derivation $n_w = 1$ (newtonian fluid), and combining it with Eqs. (42) and (52) to (54) expands to

$$\boldsymbol{u} = -\boldsymbol{k}\lambda(\nabla P - (\rho_w f_w + \rho_o f_o) g \nabla z). \tag{58}$$

Alternatively, for $n_w \neq 1$ and neglecting gravitational forces the vector for velocity is

$$\boldsymbol{u} = \boldsymbol{k}\nabla P \ \left(\lambda + \lambda_w \left((\boldsymbol{k}\lambda_w \nabla P)^{\frac{1}{n}-1} - 1\right)\right). \tag{59}$$

Having defined a total velocity $\boldsymbol{u}$, Eq. (38) becomes

$$\nabla \cdot \boldsymbol{u} = -\frac{\partial \phi}{\partial t} + q_w + q_o, \tag{60}$$

which by inserting Eq. (58) gives the following form of the *pressure equation*

$$-\nabla \cdot (\boldsymbol{k}\lambda(\nabla P - (\rho_w f_w + \rho_o f_o) g \nabla z)) = -\frac{\partial \phi}{\partial t} + q_w + q_o, \tag{61}$$

alternatively, one can make use of Eq. (59)

$$-\nabla \cdot \left(\boldsymbol{k}\nabla P \ \left(\lambda + \lambda_w \left((\boldsymbol{k}\lambda_w \nabla P)^{\frac{1}{n}-1} - 1\right)\right)\right) = -\frac{\partial \phi}{\partial t} + q_w + q_o, \tag{62}$$

which after deriving, to get rid of the exponent over ∇P looks like Eq. (50) but expressed with the mobility term λ

$$-\nabla \cdot (\boldsymbol{k}\lambda_o \nabla P) - \frac{\nabla \cdot (\boldsymbol{k}\lambda_w \nabla P)}{n \left|\boldsymbol{k}\lambda_w \nabla P\right|^{1-\frac{1}{n}}} = -\frac{\partial \phi}{\partial t} + q_w + q_o. \tag{63}$$

The *saturation equation* can be derived taking the aqueous phase (λ_w) of the velocity expressed for each phase

$$\boldsymbol{u}_w = f_w \boldsymbol{u} + \boldsymbol{k}\lambda_o f_w \nabla P_c + \boldsymbol{k}\lambda_o f_w (\rho_w - \rho_o) g \nabla z \tag{64}$$

$$\boldsymbol{u}_o = f_o \boldsymbol{u} - \boldsymbol{k}\lambda_o f_o \nabla P_c - \boldsymbol{k}\lambda_w f_o (\rho_w - \rho_o) g \nabla z, \tag{65}$$

and combining it with the aqueous phase (λ_w) of Eq. (38)

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot \left(\boldsymbol{k}\lambda_o f_w \left(\frac{dP_c}{dS_w} \nabla S_w + (\rho_w - \rho_o)\, g \nabla z\right) + f_w \boldsymbol{u}\right) = -S \frac{\partial \phi}{\partial t} + \frac{q_w}{\rho_w}. \tag{66}$$

Assuming no compressibility and no capillary pressure $P_c = 0$, (66) is reduced to

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (f_w \boldsymbol{u} + \boldsymbol{k}\lambda_o f_w (\rho_w - \rho_o) g \nabla z) = \frac{q_w}{\rho_w} \tag{67}$$

and combined with Eqs. (55) and (60) with $F(s_w) = f_w$ and doing this $\nabla \cdot f_w = \frac{df_w}{dS_w} \nabla \cdot S$ gives

$$\begin{aligned}
\phi \frac{\partial S_w}{\partial t} + \boldsymbol{u}\ (\nabla \cdot f_w) + f_w\ (\nabla \cdot \boldsymbol{u}) + \boldsymbol{k} g \nabla z\ (\nabla \cdot (\nabla_o f_w)) &= \frac{q_w}{\rho_w} \\
\phi \frac{\partial S_w}{\partial t} + \left(\frac{df_w}{dS} \boldsymbol{u} + \frac{d(\lambda_o f_w)}{dS} (\rho_w - \rho_o) g \boldsymbol{k} \nabla z\right) \nabla \cdot S &= \frac{q_w}{\rho_w} - \left(\frac{q_w}{\rho_w} + \frac{q_o}{\rho_o}\right) f_w \\
\phi \frac{\partial S_w}{\partial t} + \left(\frac{df_w}{dS} \boldsymbol{u} + \frac{d(\lambda_o f_w)}{dS} (\rho_w - \rho_o) g \boldsymbol{k} \nabla z\right) \nabla \cdot S &= \frac{q_w f_o}{\rho_w} - \frac{q_o f_w}{\rho_o}
\end{aligned} \tag{68}$$

which by neglecting the gravitational effects is the more familiar form of the Buckley-Leverett equation, (Chen et al., 2006) as presented in the handouts by Kleppe (2015) with included sink and source terms. Here used as the *saturation equation* and discretized in Section A

$$\phi \frac{\partial S_w}{\partial t} + \frac{df_w}{dS} \boldsymbol{u} \nabla \cdot S = \frac{q_w f_o}{\rho_w} - \frac{q_o f_w}{\rho_o}. \tag{69}$$

7 Numerical simulation using Matlab

This code reads a file containing characteristic reservoir properties of the field to be analyzed, diversified over the x, y and z axis. These values include the dimension of the field, grid size, permeability, porosity, injection and production rate, Newtonian viscosity and density of the phases involved, water, oil and polymer, as well as a non-Newtonian factor for viscosity, initial values for the water saturation and a desired lapse of time between calculations. The code makes use of a Corey type relation to describe the relative permeability of different sections. This relation is a correlation of lab results and depends on the endpoint relative permeability of the phases involved, obtained by flooding core samples. Seright (2010) makes use of this type of relation for his *Base Case* and *North Slope Case*, the latter one representative of an actual field, and can be found in Table 1.

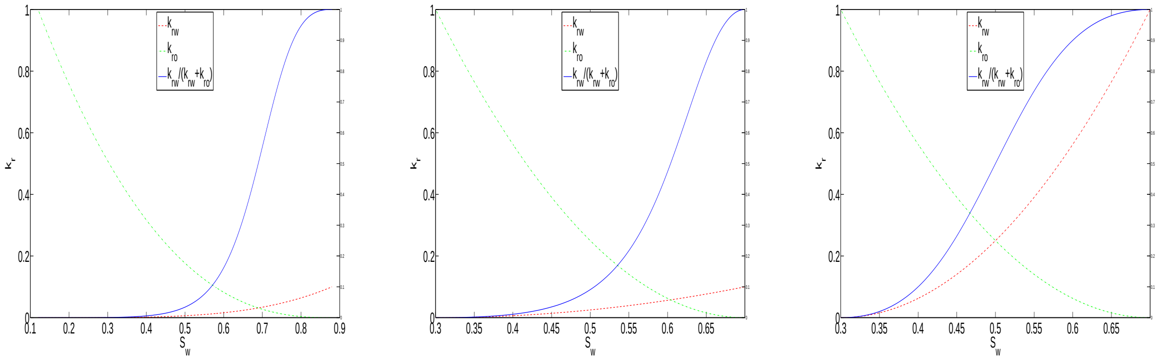
Table 1: Field Characteristics for depletion.

	North Slope Case	Base Case	Base Case-2
$k_{ro_{end}}$	1	1	1
$k_{rw_{end}}$	0.1	0.1	1
$S_{o_{irr}}$	0.12	0.3	0.3
$S_{w_{irr}}$	0.12	0.3	0.3
n_{k_w}	4	2	2
n_{k_o}	2.5	2	2

Where the relative permeability for oil and water are defined as

$$k_{ro} = k_{ro_{end}}[\frac{S_w - S_{w_{irr}}}{1 - S_{o_{irr}} - S_{w_{irr}}}]^{n_{k_o}}, \qquad k_{rw} = k_{rw_{end}}[\frac{1 - S_{o_{irr}} - S_w}{1 - S_{o_{irr}} - S_{w_{irr}}}]^{n_{k_w}}. \tag{70}$$

The behavior of these relative permeability's with varying saturation and the slope of relative permeability of the displacing fluid to the total fluid as saturation increases can be found plotted in Fig. 12.



(a) Base Case

(b) Base Case

(c) North Slope Case

Figure 12: Relative Permeability, a) and b) used by Seright (2010).

Furthermore, for simplicity purposes this code assumes no capillary pressure and no fluid and rock compressibility.

After reading the reservoir properties, the values are distributed over the rectangular coordinates, depending on how many zones were described, the length of each of one, and the constant grid length Δ of each zone. The reservoir will then be divided by a finite number of blocks with a volume $\Delta x_i \Delta y_j \Delta z_k$, each with the corresponding properties at that particular place. The first iteration makes use of the user input saturation values to calculate the relative permeability profiles for every block while the following iterations make use of the updated saturation profile obtained in the previous iteration.

Once the properties of each block is calculated, the transmissibility between blocks is calculated. The transmissibility is a value representing the flow between adjacent blocks and include the permeability, viscosity, non-Newtonian factor of the displacing fluid and grid length of the blocks involved. For the simulations carried out, the outside borders of the field are set to no flow, in other words transmissibility equal to zero. Transmissibility values vary in the range $0 < Tr < 1$ where $Tr = 1$ indicates complete contact between adjacent blocks, and can be modified manually to open up for more heterogeneities, as are the prevention of flow at different locations by what could be the presence of shale and facilitation of flow at other locations in the case of existing fractures. The transmissibility is calculated following the discretization in Section A and assuming no gravity forces, considering all the assumptions made to this point, the only determining factor for the pressure aside from the produced and injected fluid.

Matlab then solves a set of linear equations altered by the transmissibility values and returns the pressure distribution for that iteration. Having assumed no capillary pressure, this pressure is now the global pressure referred to in Section 6.7 from which the velocity distribution is calculated for each block in every coordinate. Which used together with the fractional flow profiles calculated from the saturation distribution of the previous run, updates the saturation of each block. The restrictions of this saturation update include: a) not exceeding the maximum saturation of the displacing fluid given by the irreducible saturation of the displaced fluid, as well as b) not being inferior to the corresponding lower limit given by the irreducible saturation of the displacing fluid, $S_{irr} < S_w < 1 - S_{or}$.

The next iteration makes use of the newly calculated saturation to produce a new saturation profile. The file includes plots that register the progress of said saturation making it possible to locate the fluid front graphically.

Aside from the two choices of relative permeability behavior previously presented, this file includes two distinct possible fluid viscosity behaviors. These are the analytical non-Newtonian viscosity dependence (Wu et al., 1991; Fakcharoenphol & Wu, 2010), with and without the velocity effects taken into account, the former being primarily used to produce results.

7.1 Weaknesses or shortcomings of using this particular Matlab code

As one might expect, the author encountered some challenges in the making of the code. These challenges were mostly solved for but may have resulted in shortcomings or extra processing times, mainly due to the inexperience of the author with Matlab and computation or the rare case that Matlab is not suitable for this task.

Firstly, to avoid long scripts of data, the code reads values divided into sections. The author believed that taking this approach would minimize the workload it takes to make the input files. Same reason for which a distribution of properties in the vertical axis was halted and instead input manually; however, it is possible to implement. The downside of implementing sections is that the heterogeneity distribution is affected. The benefits of this however, is that some sections could be of smaller size and have an increased grid distribution; of particular interest for fast changes as would be the presence of faults.

For the calculation of the effective polymer viscosity as presented in Section 6.5, one can see that the saturation cannot be equal to the irreducible water saturation $S_w \neq S_{irr}$ or the Darcy Eq. (8) becomes undefined. So in the case they were to be equal, a constant, bigger than zero and small enough to not affect the results was added to S_w. Similarly to the previous challenge, the saturation being the same for several neighboring blocks in an iteration, as is the case for fully aqueous saturated blocks close to the injection point, returned errors in Matlab. The difference between block properties was so small when the saturation was equal (relative permeability), that Matlab considered the matrix rows to be non-unique. This required making the rows more varied while maintaining the likeliness between them. Chances for the saturation to be equal between blocks is almost impossible unless we are considering full saturation of blocks with the same irreducible oil saturation. So for that case, the author introduced a random sequence of the size of 10^{-5} and subtracted it to the saturation. This solved the issue but altering those values means that the results are also changed, albeit slightly. No other reasonable solution came to mind and considering the low values altered and that at that stage the flow is mostly static, these changes would not affect the overall performance.

To solve the pressure equation with Matlab, it was a prerequisite to use a two dimensional matrix. This meant modifying the three dimensional system introduced to allocate the field properties that extended from $i = 1..GridX$, $j = 1..GridY$, $k = 1..GridZ$ for any (i, j, k) and reduce it to a two dimensional system (i, j). This was obtained introducing a new variable that translated the (i, j, k) coordinates the following way

$$(k+(j-1)GridZ+(i-1)GridYGridZ, k^*+(j^*-1)GridZ+(i^*-1)GridYGridZ) = (i, j, k), \tag{71}$$

where (i^*, j^*, k^*) are the coordinates of any of the seven grid blocks and (i, j, k) the coordinates of the block in the middle. Notice that the Z-Axis cycles faster followed by the Y-Axis.

7.2 Results

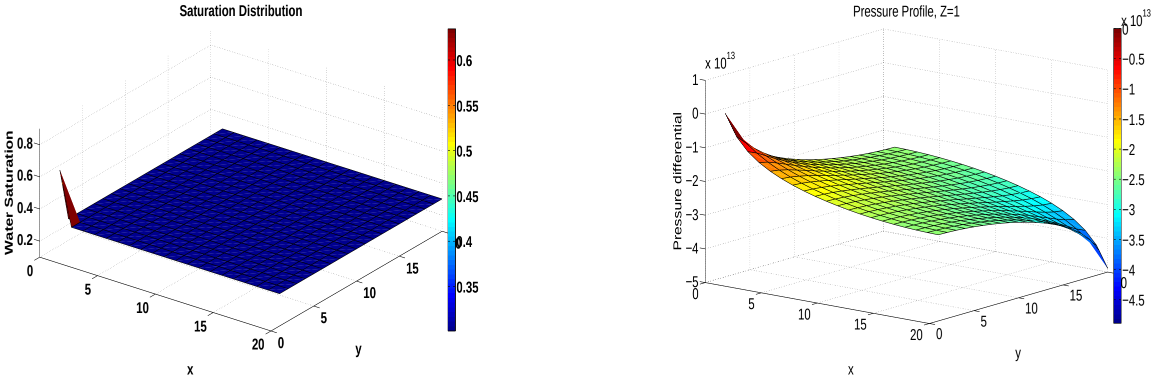
Gathered here are the results from the Matlab simulations. For comparison tool a homogeneous Base Case is created and can be found in table Table 2. This table lists zones for the different axis. This is to give certain zones a more exhaustive distribution by increasing the block amount over that particular area. Table 2 shows that the Z-Axis zone is equal to 1, this means that the block properties are shared along the Z-Axis. The extension "$\|_{(1,1,:)}$" refers to the blocks in the position $(x, y) = (1, 1)$ over the whole Z-Axis extension.

The Base Case saturation profile is plotted after 1, 50 and 150 repetitions, where t is a dimensionless variable for a dimensionless injection rate over the blocks "$\|_{(:,:,1)}$", presented in Fig. 16. This to show the saturation progress of the fluid front with increasing t, where $tq_w = 0.33$ block pore volume. Meaning that a field with the dimensions and properties of

Fig. 16 for an injection rate of $1250m^3/day$ per XY-plane, takes $453days$ to reach water breakthrough, Table 3. Furthermore, Figs. 13a, 14a and 15a show these results in three dimensions, to make it easier to see the inclination and the transition region before the front. Next to each of them is the pressure distribution, Figs. 13b, 14b and 15b.

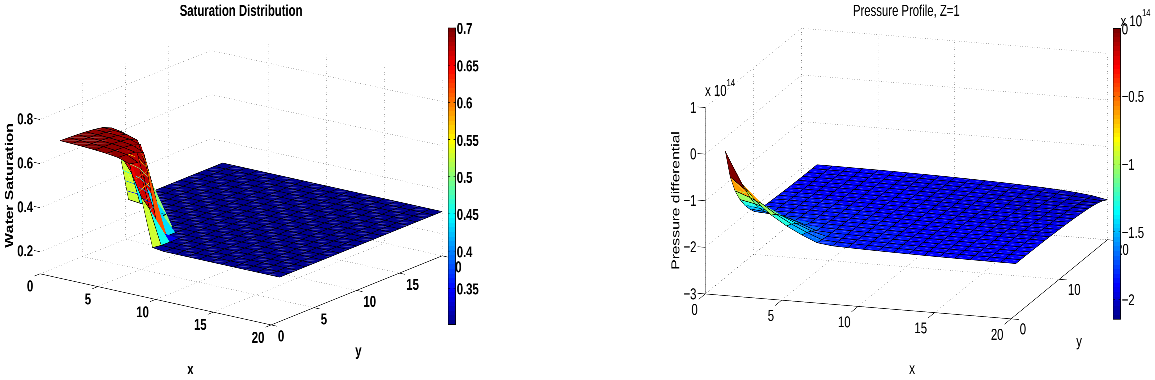
Table 2: Base case properties of the reservoir.

Property	Base case	Unit
$Initial\ Saturation$	0.301	
$Permeability\ X - Axis$	$3.0x10^-13$	m^2
$Permeability\ Y - Axis$	$3.0x10^-13$	m^2
$Permeability\ Z - Axis$	$3.0x10^-14$	m^2
$Porosity$	0.3	
$Extension\ X - Axis$	500	m
$Extension\ Y - Axis$	500	m
$Extension\ Z - Axis$	100	m
$Zones\ X - Axis$	3	m
$Zones\ Y - Axis$	3	m
$Zones\ Z - Axis$	1	m
$Blocks\ X - Axis$	25	m
$Blocks\ Y - Axis$	25	m
$Blocks\ Z - Axis$	20	m
ρ_{oil}	970	kg/m^3
$\rho_{polymer}$	970	kg/m^3
$Factor\ N\ for\ polymer$	1	
$Polymer\ Injection\ X - Axis\|\|_{(1,1,:)}$	0.001	
$Oil\ Injection\ X - Axis\|\|_{(1,1,:)}$	0	
μ_{oil}	6	cp
$\mu_{polymer}$	6	cp
$Time\ between\ calculations$	$2.5x10^3$	
$Type\ of\ viscosity$	$Newtonian$	
$Relative\ Permeability$	$Base\ case$	



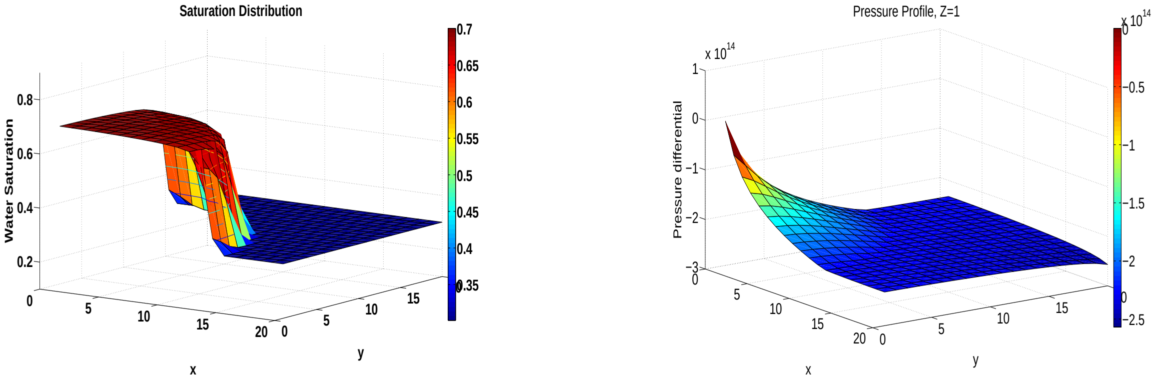
(a) Saturation (b) Pressure

Figure 13: Base case Saturation and Pressure Profiles, 1t.



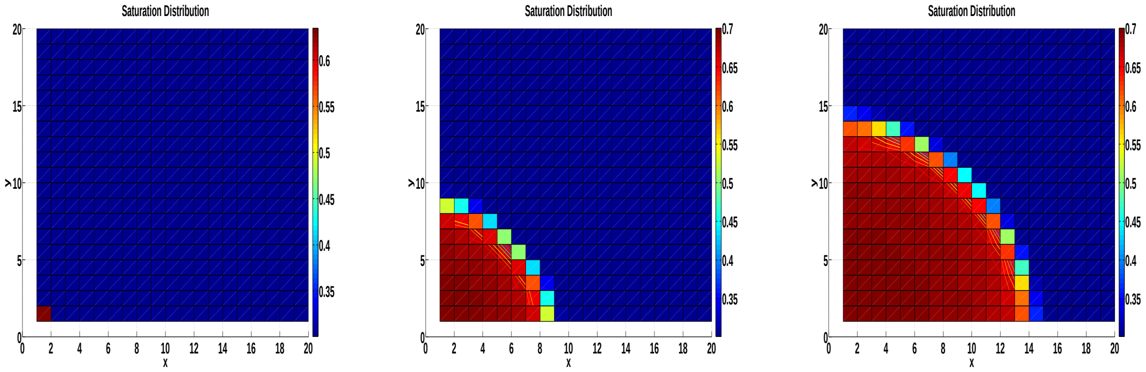
(a) Saturation (b) Pressure

Figure 14: Base case Saturation and Pressure Profiles, 50t.



(a) Saturation (b) Pressure

Figure 15: Base case Saturation and Pressure Profiles, 150t.



(a) Saturation 1t

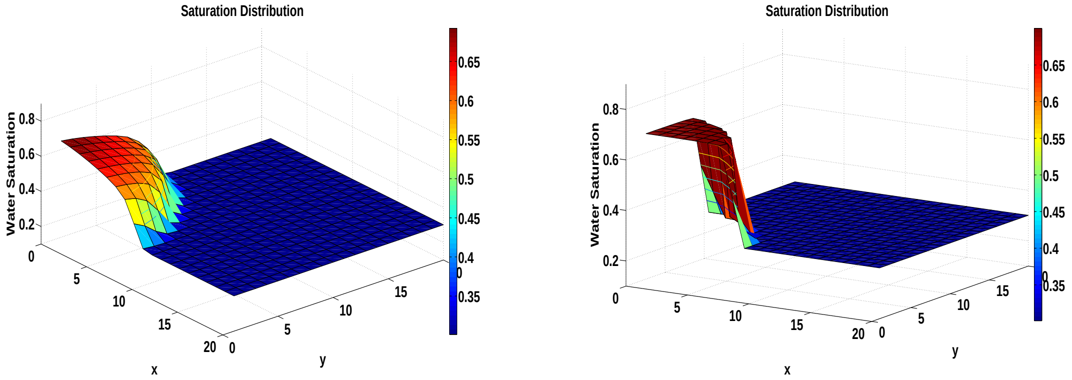
(b) Saturation 50t

(c) Saturation 150t

Figure 16: Base case Saturation, vertical perspective for 1t, 50t and 150t.

From Figs. 13 to 15 one can see that the Base Case has a very steep inclination, characteristic of a fluid front as it moves towards the exit. It starts at ≈ 0.65 and that the transition extension of this front to seems to affect $\approx 3-4$ parallel blocks Fig. 16.

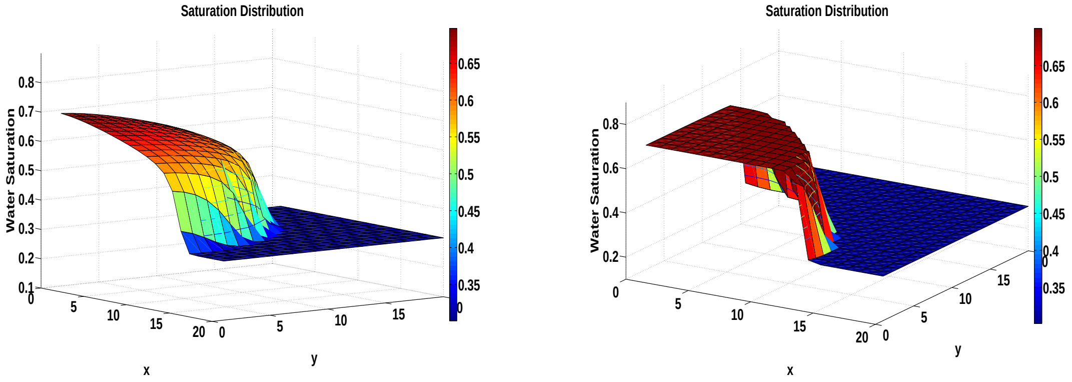
This transition however, is not so noticeable, therefore our next run will consist of variations of the polymer solution viscosity, Figs. 17 to 20. Therefore, the Low viscosity Base Case (LBC) is defined as $0.6cp$ and the High viscosity Base Case defined (HBC) as $60cp$, a ten factor viscosity change with the Base Case. The LBC displays a broad selection of colors that indicate a seemingly gradual increase in saturation towards the production well. However, the extent reached is larger than for the Base Case. In comparison, the HBC does not seem to differ much from the BC in terms of reach but shows a steeper inclination than that experienced from the BC, starting at a saturation very close to the maximum possible, ≈ 0.7, fully saturating the area before the front.



(a) Saturation, low viscosity 0.6cp

(b) Saturation, high viscosity 60cp

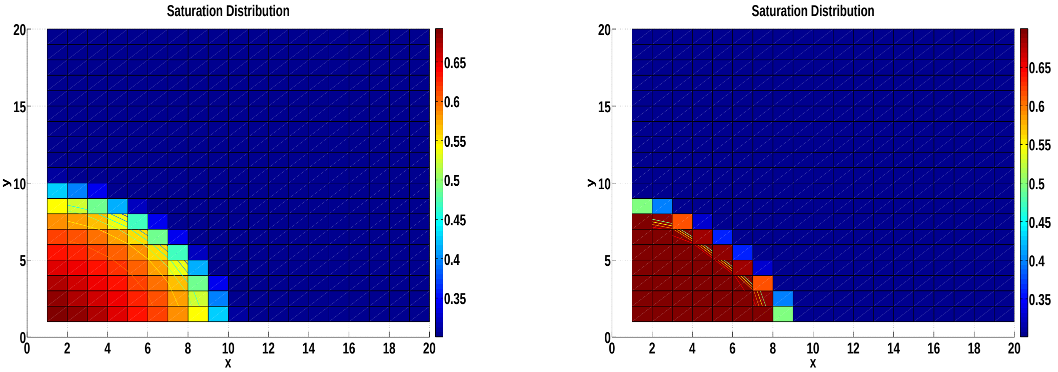
Figure 17: Variation of viscosity, at 50t.



(a) Saturation, low viscosity 0.6cp

(b) Saturation, high viscosity 60cp

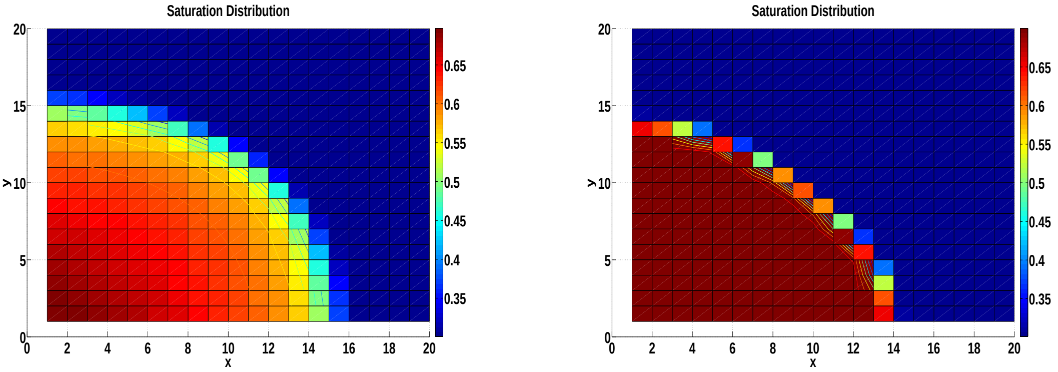
Figure 18: Variation of viscosity, at 150t.



(a) Saturation, low viscosity 0.6cp

(b) Saturation, high viscosity 60cp

Figure 19: Base case Saturation, vertical perspective at 50t.



(a) Saturation, low viscosity 0.6cp

(b) Saturation, high viscosity 60cp

Figure 20: Base case Saturation, vertical perspective at 150t.

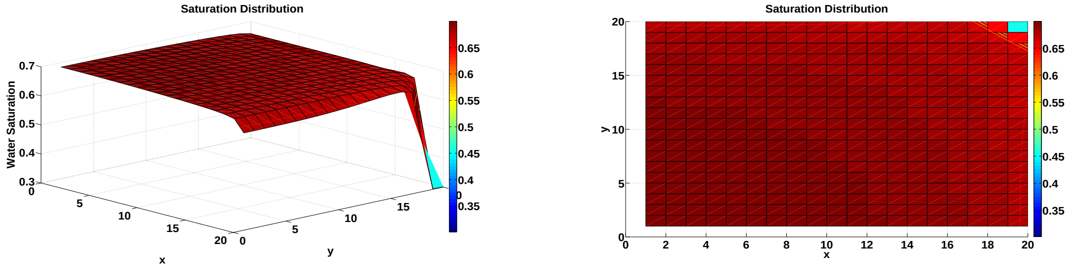
The reach of injection fluid was shown to increase with decreasing polymer solution viscosity, Figs. 17 to 20. This would result in an earlier breach of injection solution to the production well, commonly known as water breakthrough. The polymer solution viscosity is believed to vary with the non-Newtonian behavior and as such, several cases were simulated to show the change in water breackthrough. These results are listed in Table 3.

Table 3: Water breakthrough for different polymer viscosities, permeability profiles and no flow boundaries.

Permeability Profile Base Case

Water Breakthrough	Base case	Low viscosity, 0.6cp	High viscosity, 60 cp
$N = 1$	$453t$	$319t$	$471t$
$N = 0.9$	$364t$	$183t$	$462t$
$N = 0.8$	$215t$	$77t$	$407t$
$N = 0.7$	$86t$	$25t$	$254t$
$N = 0.6$	$19t$	$11t$	$96t$
$N = 0.5$	$11t$	$10t$	$18t$
Permeability Profile 2			
$N = 1$	$363t$	$271t$	
$N = 0.8$	$207t$	$77t$	
Permeability Profile 2, Boundary Profile 2			
$N = 1$	$434t$	$290t$	
$N = 0.8$	$124t$	$30t$	
Boundary Profile 3			
$N = 1$	$537t$	$318t$	
$N = 0.8$	$202t$	$71t$	
Permeability Profile 3 (vertical)			
$N = 1$			
$\boldsymbol{k}^*_{z_{G_{Shape}=0.025}} = \boldsymbol{k}_z$	$324t$	219	340
$\boldsymbol{k}^*_{z_{G_{Shape}=2.5}} = 10^3\boldsymbol{k}_z$	$253t$	$167t$	348
$\boldsymbol{k}^*_{z_{G_{Shape}=25}} = 10^4\boldsymbol{k}_z$	$253t$	$166t$	426
$\boldsymbol{k}^*_{z_{G_{Shape}=25000}} = 10^7\boldsymbol{k}_z$	$253t$	$166t$	471
$\boldsymbol{k}^*_z = 1$	$253t$	$166t$	517

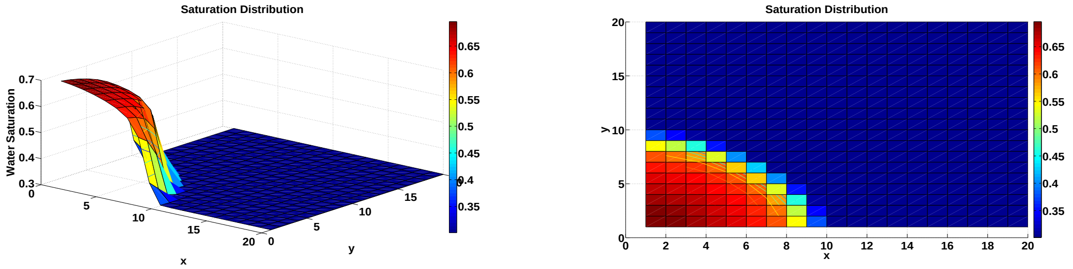
Fig. 21 shows the BC saturation distribution at the time of water breakthrough from two different perspectives. Figs. 22 and 23 show plots of saturation at the time of water breakthrough and at $50t$ for the Base Case Permeability Profile with a non-Newtonian viscosity given by $N = 0.9$.



(a) Three dimensional view

(b) Vertical view

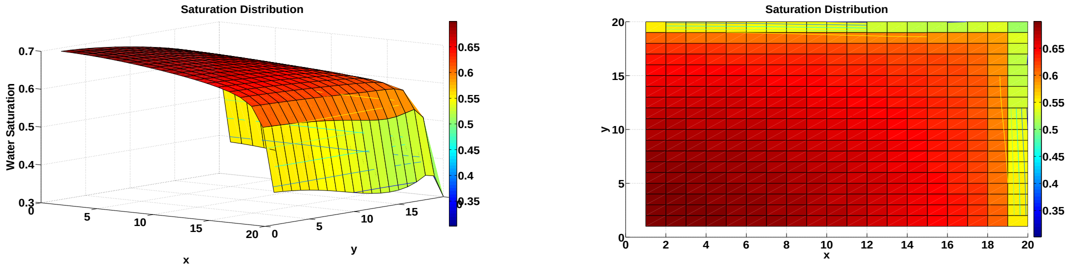
Figure 21: Base Case Saturation distribution at the time of water breakthrough.



(a) Three dimensional view

(b) Vertical view

Figure 22: Base Case Permeability Profile, N=0.9 at 50t.

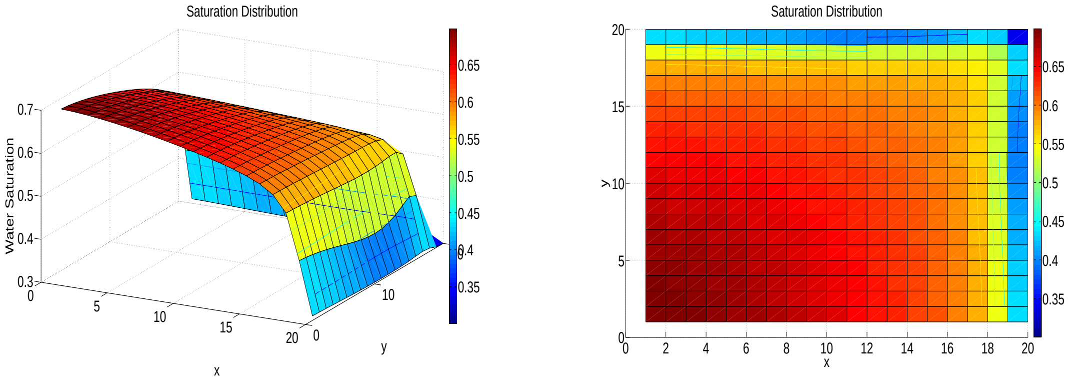


(a) Three dimensional view

(b) Vertical view

Figure 23: Base Case Permeability Profile, N=0.9 at the time of water breakthrough.

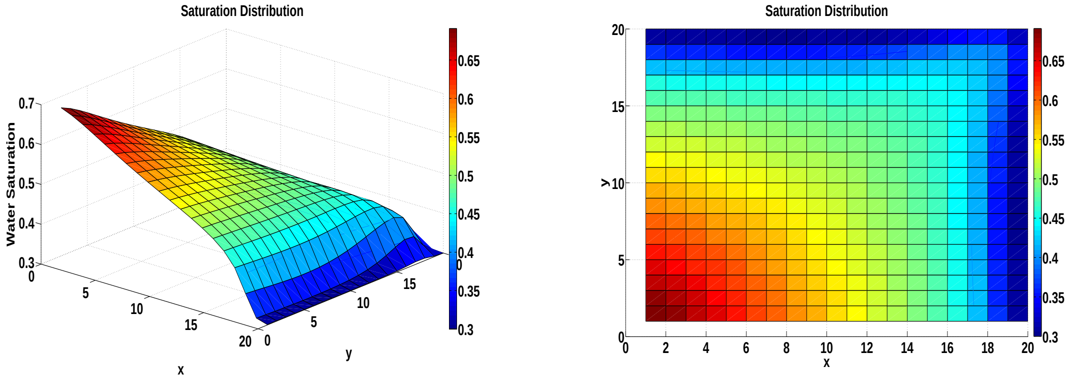
From Fig. 22 one can see that the inclination looks closer to the LBC than the original BC or HBC. Table 3 indicates that there is a change of $\approx 20\%$ in the time for water breakthrough. This effect is reflected by comparing Figs. 21 and 23, in that each block before the front is less saturated, meaning that the injected fluid reaches farther with the lower N value. However, geometrically, the middle section is far from under saturated, meaning that even at more inefficient sweep values like the implications of $N < 1$, recovery of oil is realized partially equally over the extent of the field, excluding the locality very near the borders of the X-Axis and Y-Axis. This locality is seen to increase if the relative viscosity of the system is represented by Base Case-2 from Table 1 instead.



(a) Three dimensional view

(b) Vertical view

Figure 24: Base Case 2 Relative Permeability, N=1 at the time of water breakthrough.



(a) Three dimensional view

(b) Vertical view

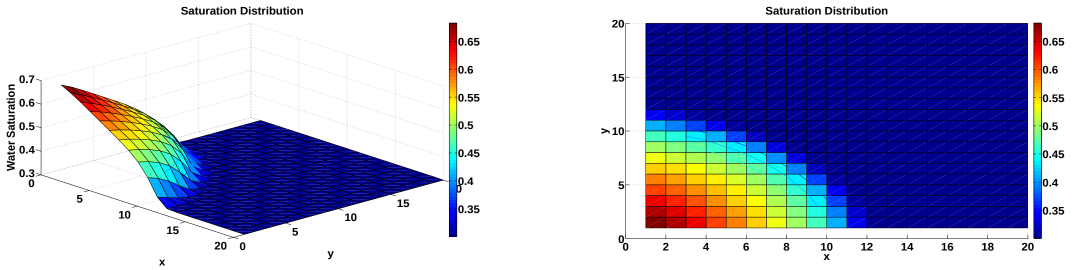
Figure 25: Base Case 2 Relative Permeability, N=0.9 at the time of water breakthrough.

Figs. 24 and 25 show the aforementioned and Table 4 includes the water breakthrough times.

Table 4: Water breakthrough for the Relative Permeability Profile Base Case-2.

Relative Permeability Profile Base Case-2	
Water Breakthrough	Base case
$N = 1$	$320t$
$N = 0.9$	$191t$

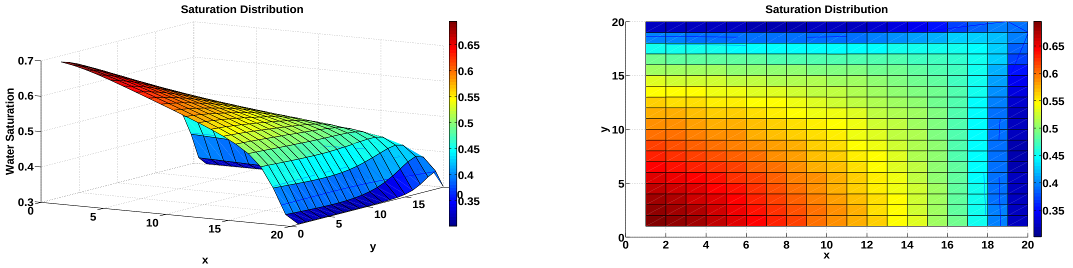
Next, follow the remaining plots for non-Newtonian viscosity factor, for the Base Case Permeability Profile.



(a) Three dimensional view

(b) Vertical view

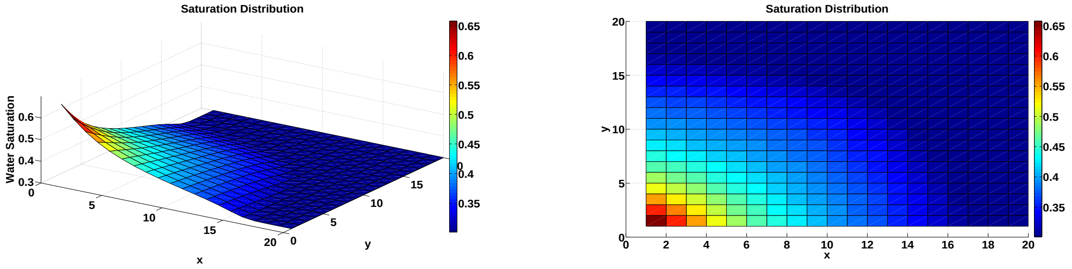
Figure 26: Base Case Permeability Profile, N=0.8 at 50t.



(a) Three dimensional view

(b) Vertical view

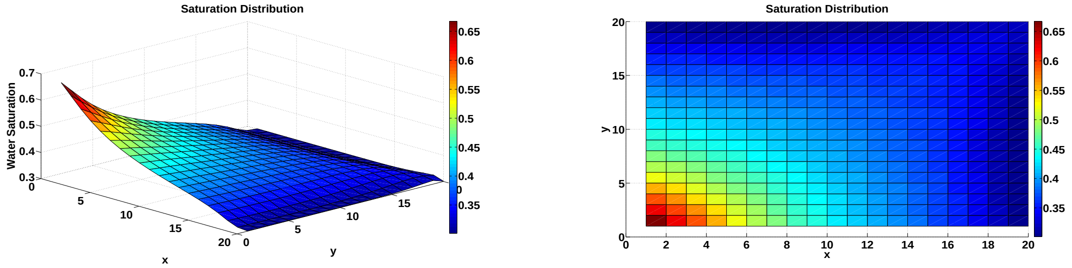
Figure 27: Base Case Permeability Profile, N=0.8 at the time of water breakthrough.



(a) Three dimensional view

(b) Vertical view

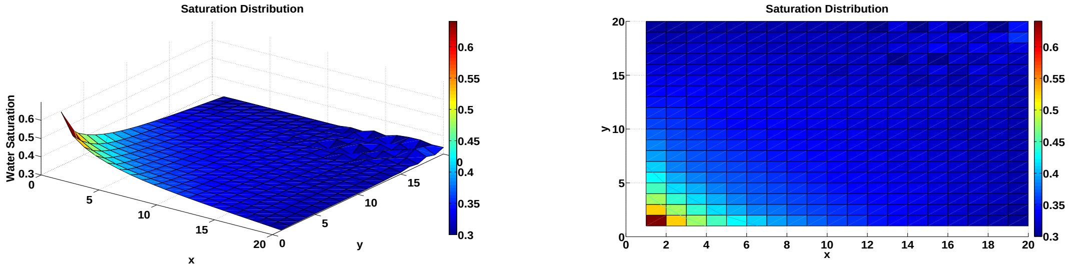
Figure 28: Base Case Permeability Profile, N=0.7 at 50t.



(a) Three dimensional view

(b) Vertical view

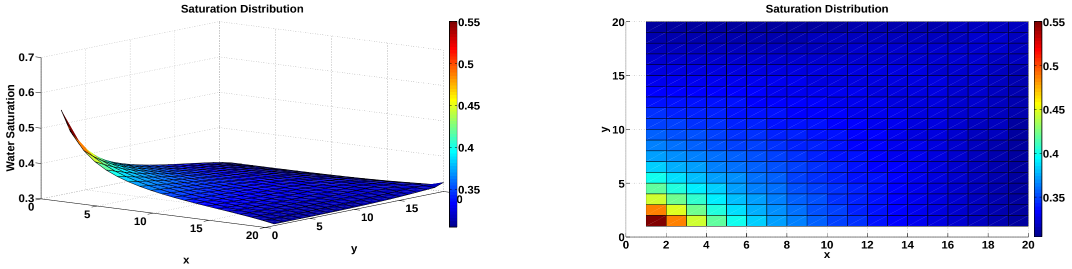
Figure 29: Base Case Permeability Profile, N=0.7 at the time of water breakthrough.



(a) Three dimensional view

(b) Vertical view

Figure 30: Base Case Permeability Profile, N=0.6 at 50t*, t*=$\frac{t}{2}$.

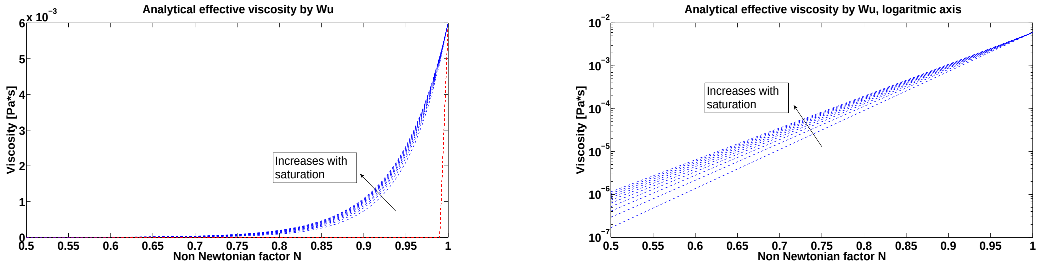


(a) Three dimensional view

(b) Vertical view

Figure 31: Base Case Permeability Profile, N=0.6 at 50t*, t*=$\frac{t}{10}$.

Figs. 26 to 29 show that this trend is consistent and it is clear that the impact of the non-Newtonian factor is considerable. These plots and Table 3 indicate that the polymer solution flows so freely that it seems to disregard the precense of oil. Another way to view the impact of the non-Newtonian characteristics of the displacing fluid is to make plots of the effective viscosity suggested by Wu et al. (1991) in Eq. (31). Plots with varying saturation displaying this effect are gathered in Fig. 32.



(a) (b) Logaritmic Y-Axis

Figure 32: Effective viscosity as a function of saturation.

Figs. 30 and 31 are plotted with a different time between calculations. The first one of these two shows the effect of halving the time between calculations, $t = 1.25x10^3$ all the way to 50t. The second one makes use of $t = 2.5x10^2$ up to the time of water breakthrough. This time is equal to $93t$ which is in concordance with Table 3 for water breakthrough with $N = 0.6$. The advantage is obvious, the result is a finer and more correct saturation distribution, obtained with a smaller time difference between calculations, a reasonable approach considering the low processing times and intended purpose (water breakthrough time).

As for the progression of the fractional flow, these can be found in Fig. 33 for the BC, LBC and HBC over the length of the X-Axis. Naturally, they all converge into a single fractional flow curve as the whole field adapts the same relative permeability behavior. Furthermore, Fig. 33 indicates that at higher viscosity values, compared to the BC, situates the fractional flow curve more to the right as opposed to what lower viscosity values like for LBC do.

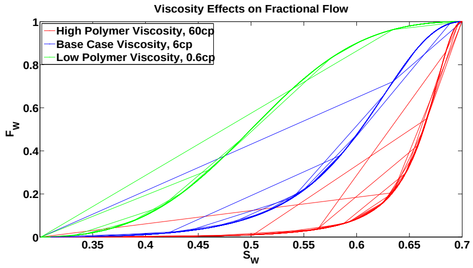


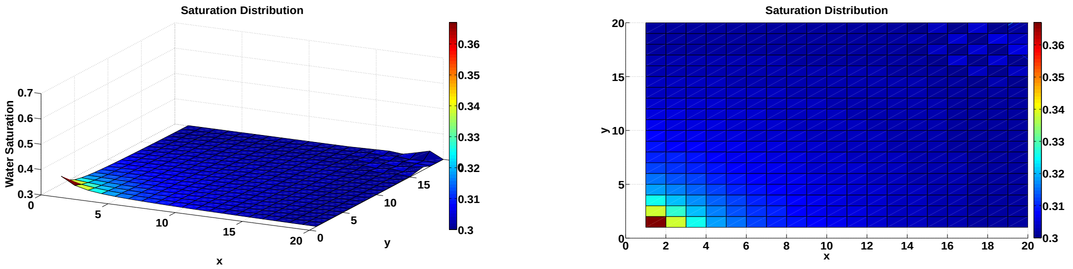
Figure 33: Fractional flow variation with viscosity along the X-Axis.

Fig. 30 shows a discontinuity in the progressive saturation increase along the X-Y plane, the same be seen in Fig. 34. This behavior was also present in the first few timesteps, before it reached 50t and implies that the equation that gradually updates saturation, is faulty, at least at lower processing times, where the block sizes or timesteps are too big to return good calculations. Aside from adjusting the time between calculations, like in Fig. 31, the equation for saturation update was changed, to see if it fixed the discontinuity in the saturation update. The new equation for saturation update was the one that Rossen et al. (2011) employed in his calculations, further extended to three dimensions for the use with this code. Eq. (72) shows the difference between the two employed equations in one dimension.

$$S_i^{t+\Delta t} = S_i^t + \left(\frac{\Delta t}{\Delta x}\right)_i \left({f^t}_{w_i} - {f^t}_{w_{i+1}}\right) \frac{1}{\phi_i \Delta x_i A_i} \qquad \text{Option 2, as opposed to} \tag{72}$$

$$S_i^{t+\Delta t} = S_i^t + \left(\left(\frac{\Delta t}{\Delta x}\right)_{i-\frac{1}{2}} {f^t}_{w_i} - \left(\frac{\Delta t}{\Delta x}\right)_{i+\frac{1}{2}} {f^t}_{w_{i+1}}\right) \frac{1}{\phi_i \Delta x_i A_i} \qquad \text{Option 1} \tag{73}$$

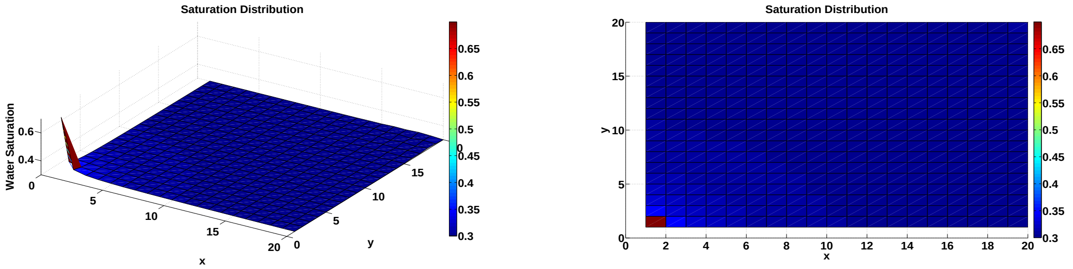
Figs. 34 and 35 contains the saturation distribution plots for the two possible equation options for the saturation update.



(a) Three dimensional view

(b) Vertical view

Figure 34: Base Case Permeability Profile, N=0.6 at 50t*, t*=$\frac{t}{10}$, Option 1.

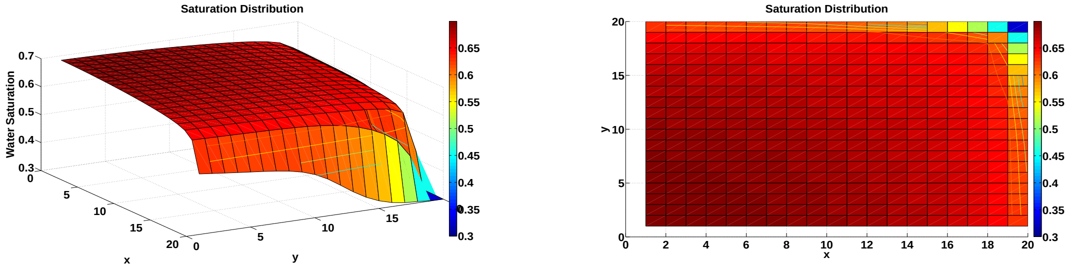


(a) Three dimensional view

(b) Vertical view

Figure 35: Base Case Permeability Profile, N=0.6 at 50t*, t*=$\frac{t}{10}$, Option 2.

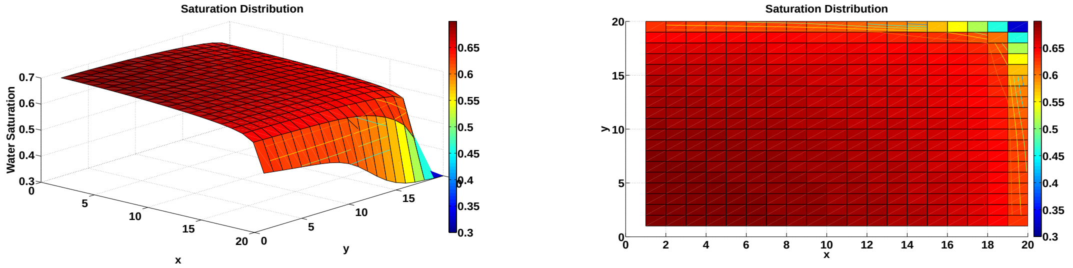
Unfortunately this change bore no fruits, mainly because of the spike in the injection block, making the result worthless but even ignoring that detail, it did not solve the issue of the wave scenario experienced with $N = 0.5$ and $N = 0.6$. However, at higher viscosity values and time between calculations, this second option returns the same distribution as the first option, Figs. 36 and 37.



(a) Three dimensional view

(b) Vertical view

Figure 36: Base Case Permeability Profile, N=0.8 at the time of water breakthrough, Option 1.

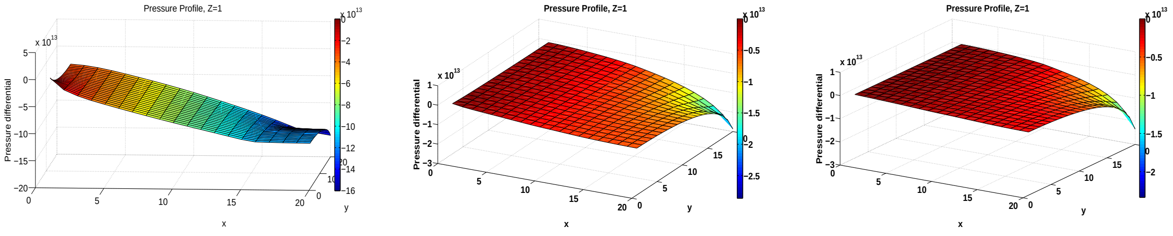


(a) Three dimensional view

(b) Vertical view

Figure 37: Base Case Permeability Profile, N=0.8 at the time of water breakthrough, Option 2.

Next, is a saturation distribution for the permeability profile 2. The field in this case, aside from the permeability over the X-Axis, maintains the characteristcs of the Base Case. The changes are found in Table 5, where the permeability is expressed in Darcy's. Figs. 39 to 41 are plots that show how the permeability profile 2 affects the displacement pattern of oil and Fig. 38, the pressure distribution at the time of water breakthrough in the field.



(a) N=1, 6cp

(b) N=0.8, 6cp

(c) N=0.8, 0.6cp

Figure 38: Pressure distribution, Permeability Profile 2, breakthrough.

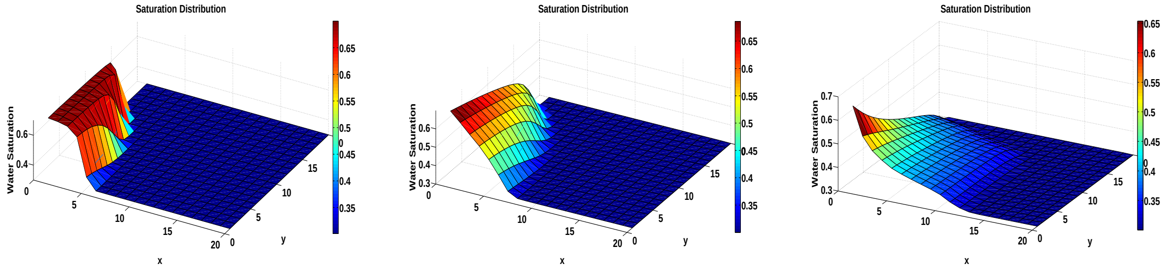
Table 5: Water breakthrough for the Relative Permeability Profile Base Case-2.

Permeability X-Axis, Base Case

	$\boldsymbol{k}_x$ [D]	$\boldsymbol{k}_x$ [D]	$\boldsymbol{k}_x$ [D]
Zone 1	0.3	0.3	0.3
Zone 2	0.3	0.3	0.3
Zone 3	0.3	0.3	0.3

Permeability X-Axis, Profile 2

	$\boldsymbol{k}_x$ [D]	$\boldsymbol{k}_x$ [D]	$\boldsymbol{k}_x$ [D]
Zone 1	6	3	0.3
Zone 2	6	3	0.3
Zone 3	6	3	0.3

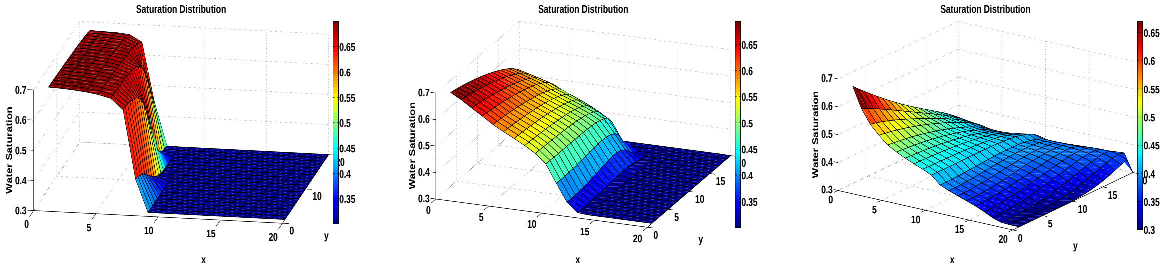


(a) N=1, 6cp

(b) N=0.8, 6cp

(c) N=0.8, 0.6cp

Figure 39: Saturation distribution, Permeability Profile 2, 50t.

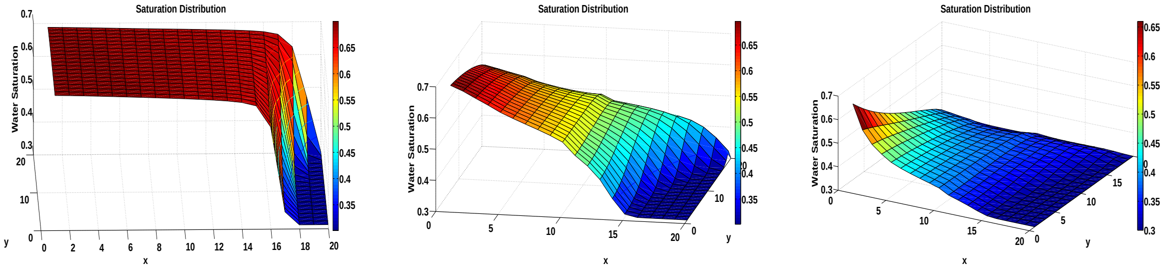


(a) N=1, 6cp

(b) N=0.8, 6cp

(c) N=0.8, 0.6cp

Figure 40: Saturation distribution, Permeability Profile 2, 150t.



(a) N=1, 6cp

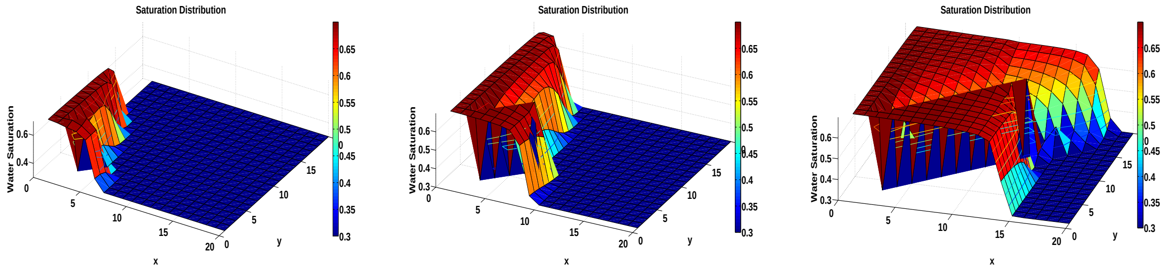
(b) N=0.8, 6cp

(c) N=0.8, 0.6cp

Figure 41: Saturation distribution, Permeability Profile 2, breakthrough.

Figs. 39 to 41 indicate that if a pattern such as the one present with the permeability profile 2 is present, the injection fluid will adhere to it in a higher degree for more viscous fluids. The pressure however, Figs. 38a to 38c, makes a clear distinction between a value of $N = 1$ to $N = 0.8$ compared to a variation from $\mu = 6cp$ to $\mu = 0.6cp$, for this particular case.

The permeability profile 2 was simulated in conjunction with a field affected by what would be a rock formation impeding flow. This forms a diagonal stretch from the locality of the injection block to the locality of the production block, on every Z-plane.

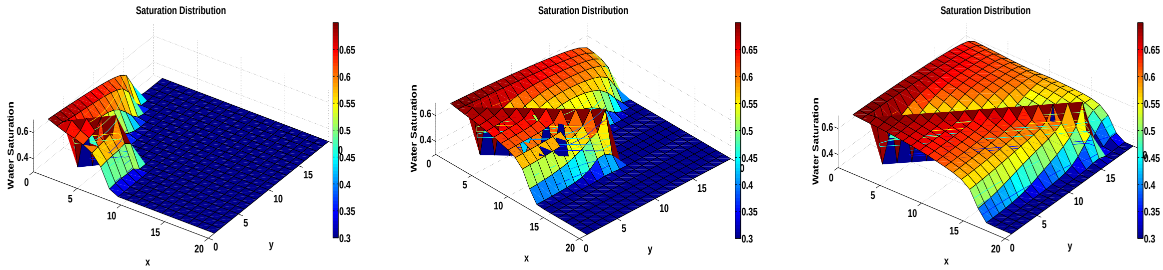


(a) 50t

(b) 150t

(c) At the time of water breakthrough

Figure 42: Saturation distribution, Permeability Profile 2, Boundary Profile 2, N=1, 6cp.



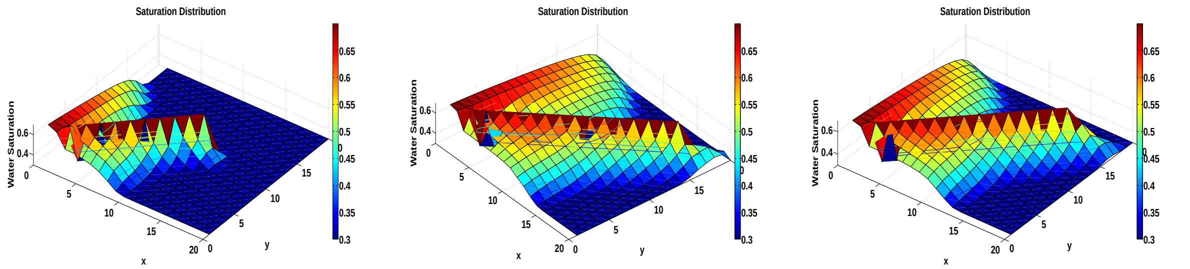
(a) 50t

(b) 150t

(c) At the time of water breakthrough

Figure 43: Saturation distribution, Permeability Profile 2, Boundary Profile 2, N=1, 0.6cp.

Fig. 42 indicates that the upper section of this system reaches the water breakthrough time first. In the region close to the wall, the saturation in the upper section is lower than the lower section, which appears to be completely saturated. Lowering the viscosity tenfold makes it hard to distinguish which section reaches the exit first and it can be seen that both the upper and lower sections are less saturated, Fig. 43. Changing to a non-Newtonian behavior with $N = 0.8$, Fig. 44 interchanges the section that reaches the production well first. Decreasing the viscosity tenfold for this case too, Fig. 45 makes it so that the distribution of the injection fluid seemingly ignores the fault and instead flows diagonally by the sides of the impenetrable diagonal middle section to the production well. These figures also show some ugly holes and scratches but that is only due to the resolution of the tex system to convert the images properly. The diagonal middle section also appears to have continuous spikes, showing where the boundary between the sections exist. These come with the code, approaching sections with minimal transmissibility between adjacent blocks with flow on each side. They are believed to decrease in number or disappear with a higher grid number or time steps.

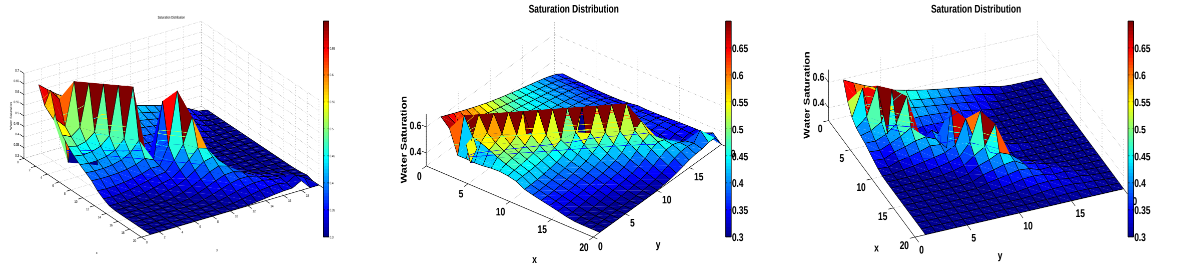


(a) 50t

(b) 150t

(c) At the time of water breakthrough

Figure 44: Saturation distribution, Permeability Profile 2, Boundary Profile 2, N=0.8, 6cp.



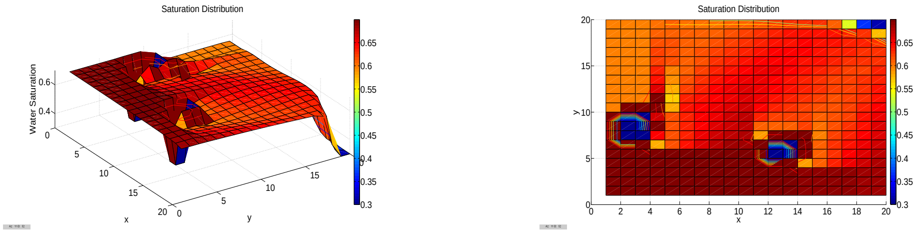
(a) 50t

(b) 150t

(c) At the time of water breakthrough

Figure 45: Saturation distribution, Permeability Profile 2, Boundary Profile 2, N=0.8, 0.6cp.

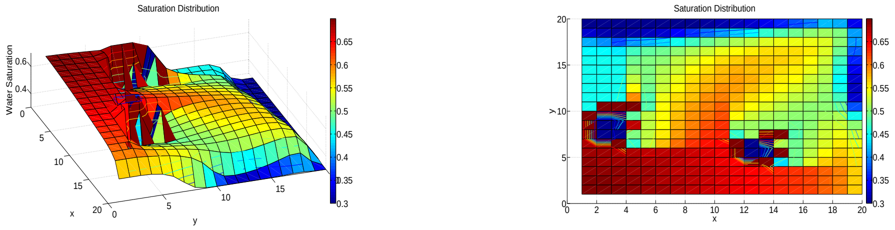
Figs. 46 to 49 consists of plots with the “Boundary 3 Profile”, to show heterogeneity in the simulations with certain boundaries in the field and to see the implications blocked off areas as opposed to just a path, would have in the recovery of oil in terms of water breakthrough time and saturation distribution. These plots show a typical representation of the effects of viscosity, being that the height of the saturation decreases with decreasing viscosity. However, due to the blocked off areas, the saturation near the injection well looks more concentrated than normal, compared to the clean Base Case and the overall saturation distribution does not systematically decrease as it gets closer to the production well. There are parts of the field that are simply less saturated than others, as behind the zones that are blocked off, from the perspective of the injection well, and parts more concentrated as are the blocks before those zones are encountered. The breakthrough times are listed in Table 3. Worth mentioning is also the fact that the flow is divided in three different paths where the path in the middle is the one depleted first as the path in the extensions contribute to it when they connect, more clear in Figs. 46b, 47b and 48b.



(a) Three dimensional View

(b) Vertical View

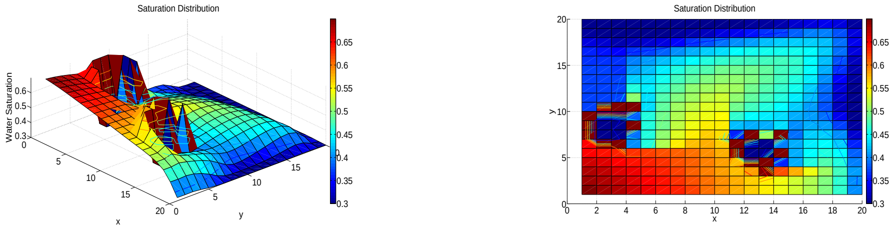
Figure 46: Saturation distribution, Boundary Profile 3, N=1, 6cp at the time of water breakthrough.



(a) Three dimensional View

(b) Vertical View

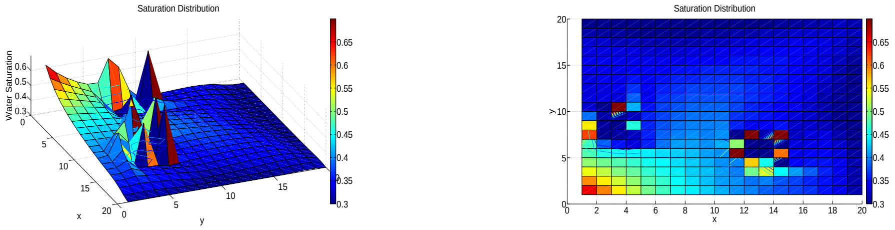
Figure 47: Saturation distribution, Boundary Profile 3, N=1, 0.6cp at the time of water breakthrough.



(a) Three dimensional View

(b) Vertical View

Figure 48: Saturation distribution, Boundary Profile 3, N=0.8, 6cp at the time of water breakthrough.

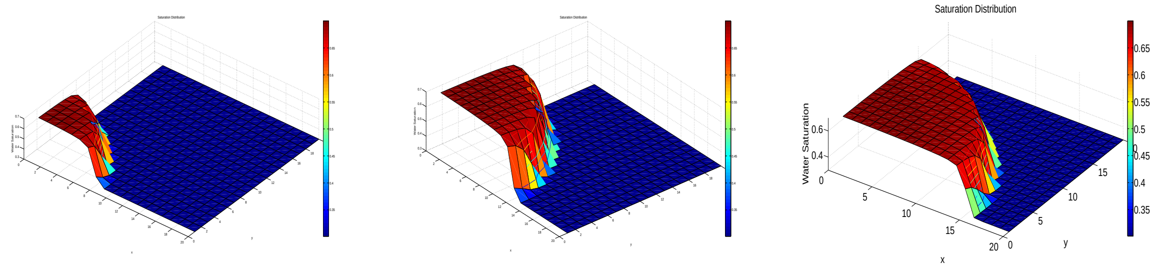


(a) Three dimensional View

(b) Vertical View

Figure 49: Saturation distribution, Boundary Profile 3, N=0.8, 0.6cp at the time of water breakthrough.

The simulations carried out for the “Permeability Profile 3” are all done for two XY-planes, the field having the same dimensions and properties as the Base Case, Table 2 with the exception of the permeability’s. The permeability in the X-Axis and Y-Axis for $Z = 2$ was increased a hundredfold and the Z-Axis permeability, for the whole field, according to table specifications. Plots for the first stretch are presented in Figs. 50 to 54, for both XY-planes. Common in all these figures is the fact that the front in the lower section reaches the destination first, and as the previous simulations, the effect of the viscosity is clear too. Less viscosity in this case pierces deeper in the lower section at the cost of the upper section. While this effect almost seems to be negated at short Δt’s where the viscosity is high, Figs. 54a and 55a. Finally, comparing the position of the front in the upper section over the X-Axis, one can see that they all agree.

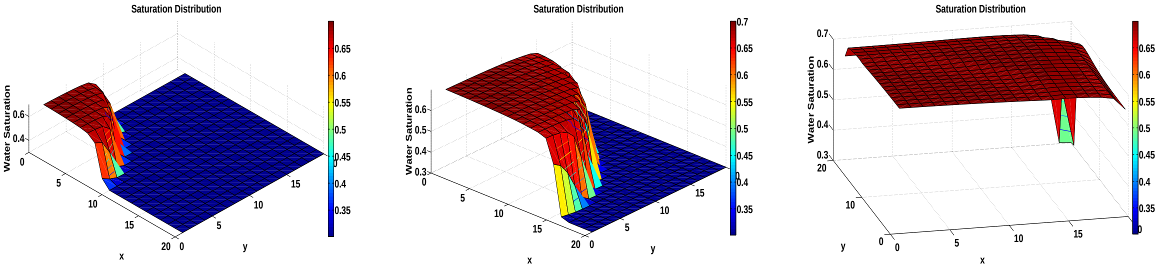


(a) 50t

(b) 150t

(c) At the time of water breakthrough

Figure 50: Saturation distribution, Permeability Profile 3 (Vertical), 6cp, Z=1.



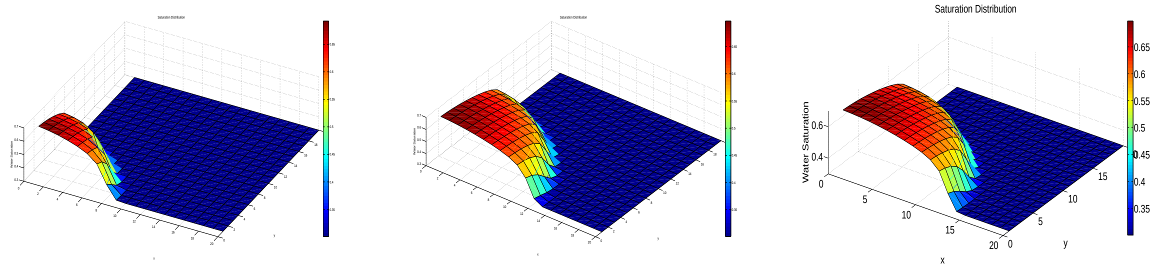
(a) 50t

(b) 150t

(c) At the time of water breakthrough

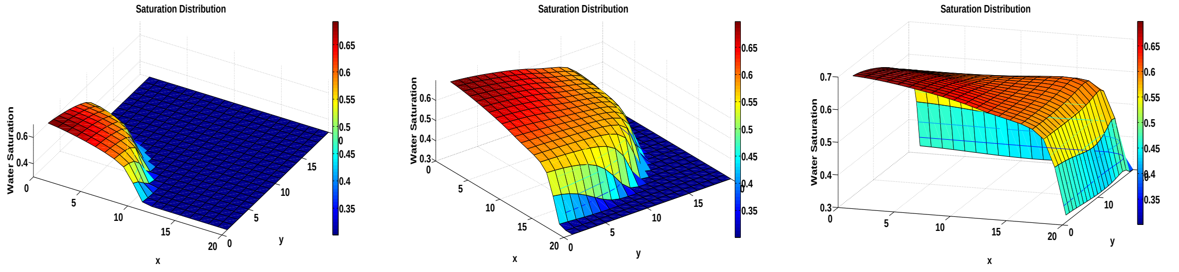
Figure 51: Saturation distribution, Permeability Profile 3 (Vertical), 6cp, Z=2.

The distribution for $6cp$ tells us that the lower XY-plane reaches water breakthrough first, the plane with a higher permeability. At the time of water breakthrough, one can see that volumetrically, the lower plane is about doubly saturated. Both planes also seem to share the saturation and hence depletion pattern, where the front in the lower section moves faster.



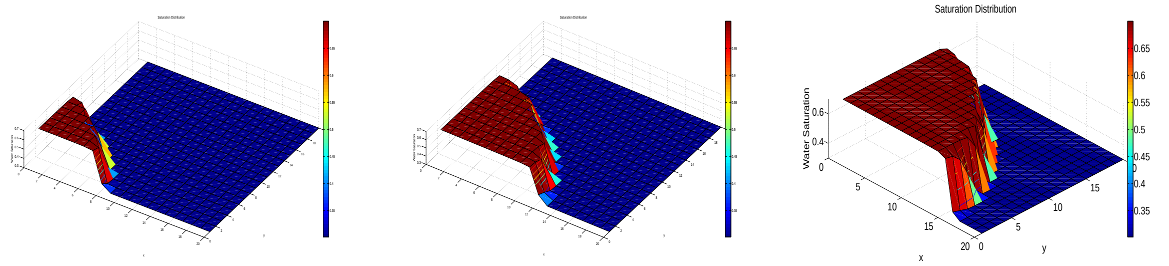
(a) 50t (b) 150t (c) At the time of water breakthrough

Figure 52: Saturation distribution, Permeability Profile 3 (Vertical), 0.6cp, Z=1.



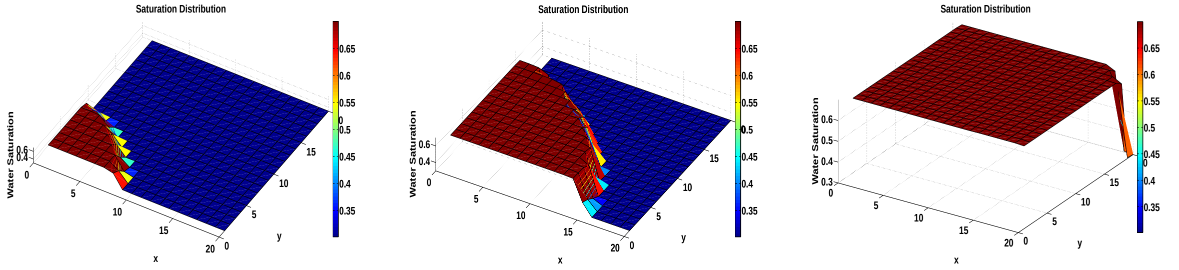
(a) 50t (b) 150t (c) At the time of water breakthrough

Figure 53: Saturation distribution, Permeability Profile 3 (Vertical), 0.6cp, Z=2.



(a) 50t (b) 150t (c) At the time of water breakthrough

Figure 54: Saturation distribution, Permeability Profile 3 (Vertical), 60cp, Z=1.



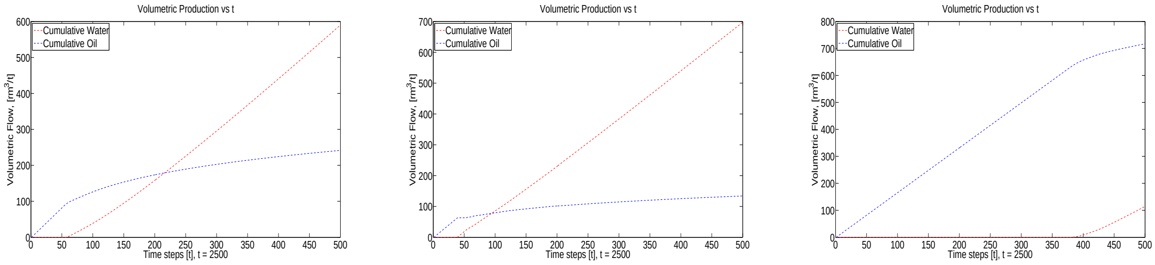
(a) 50t (b) 150t (c) At the time of water breakthrough

Figure 55: Saturation distribution, Permeability Profile 3 (Vertical), 60cp, Z=2.

To get a broader understanding of the effects of a diverse permeability profile, with contact between the layers, the XY-plane's horizontal permeability was multiplied 5^z, with the same dimensions as the BC. To show the effects of an increase in viscosity, the simulation was first ran with a low viscosity displacing fluid, to the time of water breakthrough and then up to $200t$ and $500t$, collecting the water and oil recovered in that period and plotted in Figs. 56 and 57, while Table 6 lists times for water breakthrough and values for water cut at $\Delta t = 200t, 500t$. Figs. 58 to 61 shows saturation distribution at $500t$ for different Z, when the injection fluid is being produced and the front slowly increases in height.

Table 6: Water breakthrough and Water cut, Diverse Case.

	Permeability Profile 1	Relative Permeability Profile 2	
	$0.06cp$	$0.06cp$	20
$Water\ Breakthrough$	$56t$	$40t$	$369t$
$Water\ Cut, 200t$	0.48	0.70	0.0
$Water\ Cut, 500t$	0.71	0.84	0.14

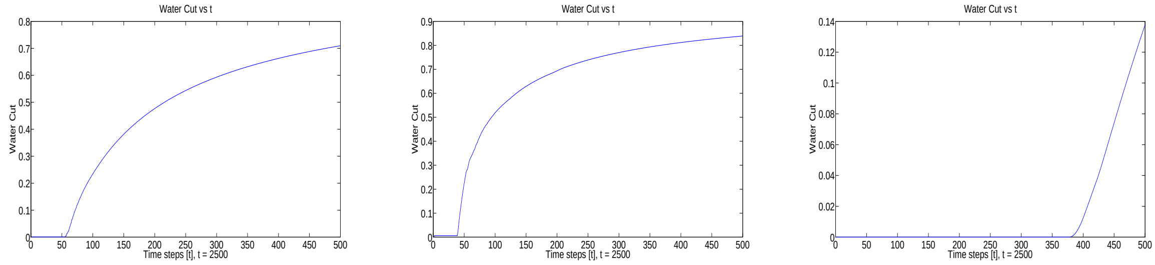


(a) 0.06cp, Profile 1

(b) 0.06cp, Profile 2

(c) 20cp, Profile 2

Figure 56: Volumetric Production.

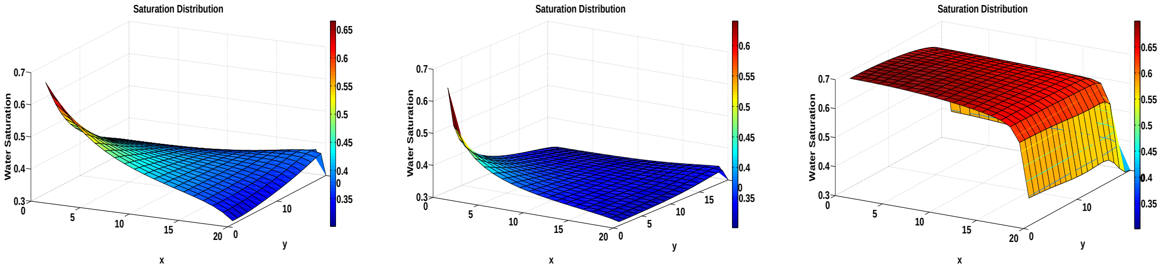


(a) 0.06cp, Profile 1

(b) 0.06cp, Profile 2

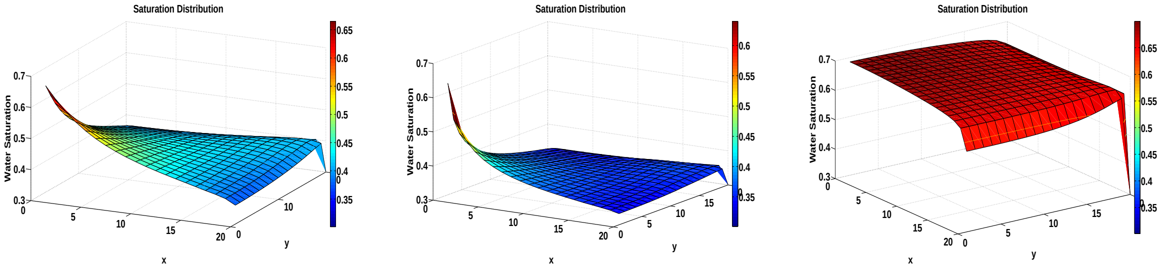
(c) 20cp, Profile 2

Figure 57: Volumetric Production.



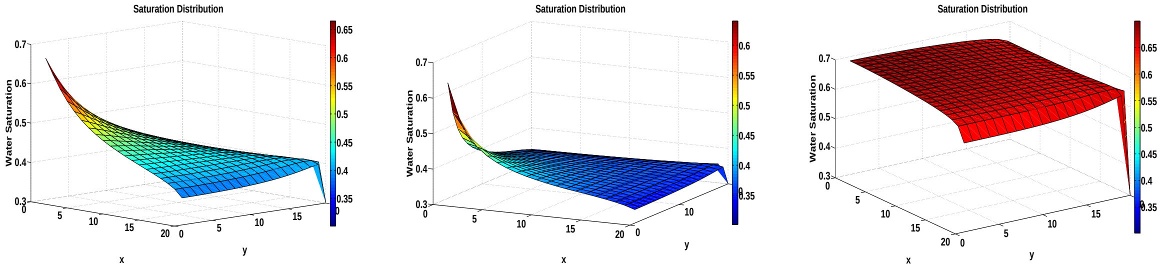
(a) 0.06cp, Profile 1 (b) 0.06cp, Profile 2 (c) 20cp, Profile 2

Figure 58: Saturation distribution, 500t.



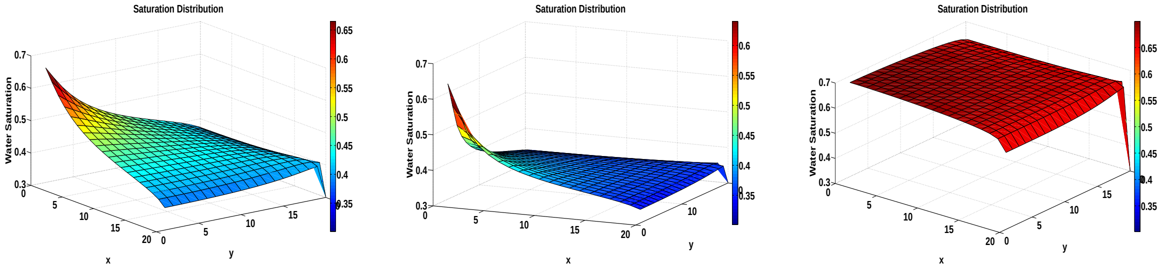
(a) 0.06cp, Profile 1 (b) 0.06cp, Profile 2 (c) 20cp, Profile 2

Figure 59: Saturation distribution, 500t.



(a) 0.06cp, Profile 1 (b) 0.06cp, Profile 2 (c) 20cp, Profile 2

Figure 60: Saturation distribution, 500t.

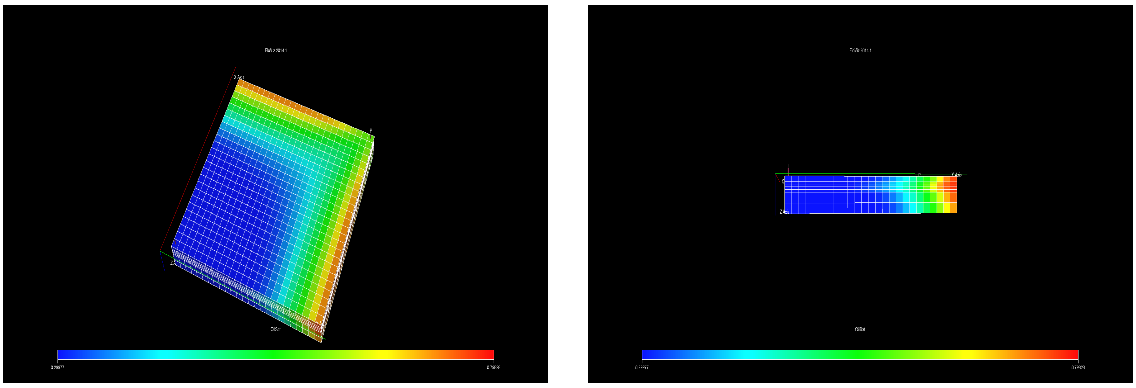


(a) 0.06cp, Profile 1 (b) 0.06cp, Profile 2 (c) 20cp, Profile 2

Figure 61: Saturation distribution, 500t.

8 Numerical simulation using Eclipse

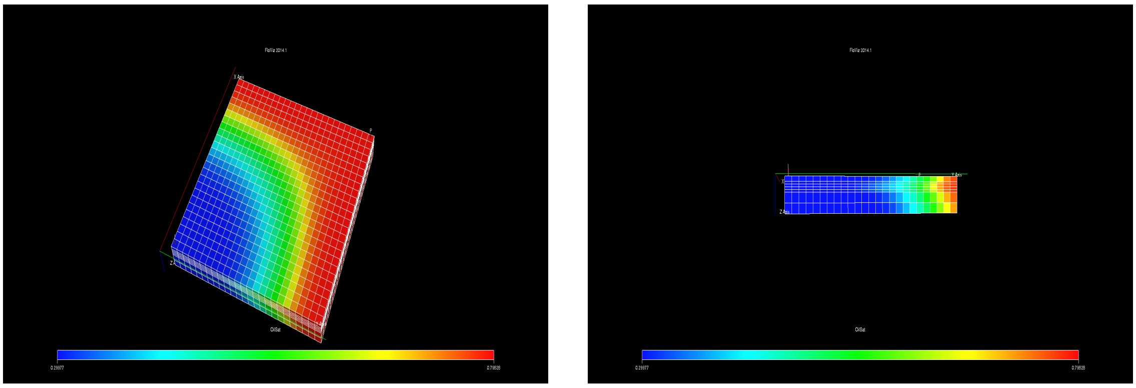
The purpose of using Eclipse, a reservoir simulation designed program, was to have something reliable to compare the performance of the Matlab code, but also to investigate the impact of adsorption and salt concentration on the displacing of oil. Three different, simple text files were created for this purpose.



(a) Top XY-plane view

(b) Side XZ-plane view

Figure 62: Saturation at the end of the simulation, File 1.

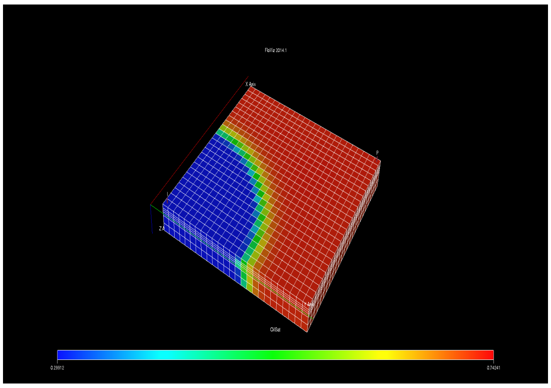


(a) Top XY-plane view

(b) Side XZ-plane view

Figure 63: Saturation, 2t before the end of the simulation, File 1.

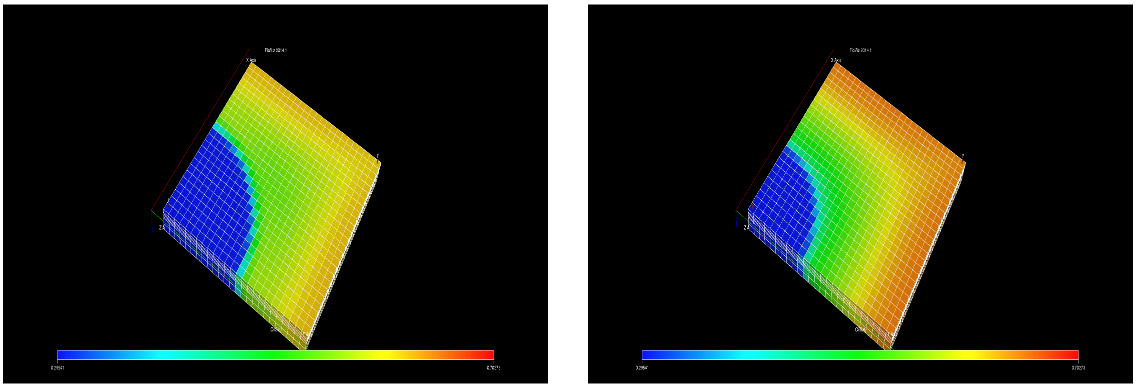
The dimensional size of the field and grid system used, of all the three files, is the same as the Base Case for the Matlab code but with the a more varied horizontal permeability distribution and 10 grid blocks in the vertical direction instead of 5 of the Base Case. The viscosity of the displacing fluid and the displaced fluid was set the same in file 1 while it increased in file 2 and overall decreased in file 3. The files can be found in **??**. Figs. 62b and 63b show how the permeability varies considerably in the vertical direction where the last 2 blocks correspond to the last zones and show that the lower one displaces oil faster. Fig. 64a on the other hand shows no visible distinction for both those zones and less difference in the distance displaced over the vertical direction of the grid system. Also note that water breakthrough has not been reached, for file 2, with the same time interval.



(a) Top XY-plane view

Figure 64: Saturation at the end of the simulation, File 2.

File 3 consists of two flushes, one like the file 2 for a short time interval followed by a salt concentrated displacing fluid that reduces its viscosity properties. This can be seen in the high color contrast of the saturation Fig. 65. The blue color is mostly the effect of the high viscous fluid, depleting the zone to the left, followed by the same fluid, filled with salts, reducing the viscosity and slowly depleting the rest. Comparing Figs. 65a and 65b, one can see that the green contrast in Fig. 65b was distributed along the remaining undepleted area, and slowly increase it more evenly Fig. 65a.

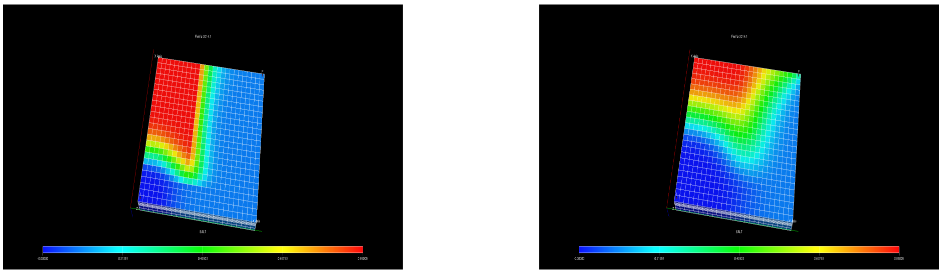


(a) Saturation final t

(b) Saturation final - 2t

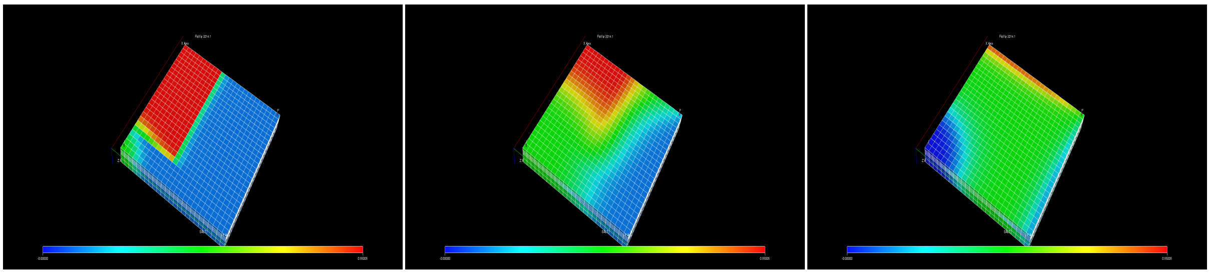
Figure 65: Top XY-plane view distribution, File 3.

In terms of salt concentratio, originally in place in the field, this was flushed out with the injection of the polymer solution. It was expected to be configured as part of the rock properties but it was defined as being part of the aqueous fluid in the reservoir, meaning that once that fluid got displaced, so did the salt concentration. A comparison between Figs. 66 and 67, shows how the salt moves with the injection of fluid, even after being part of the injectant fluid Fig. 67c.



(a) 4t (b) 6t

Figure 66: Salt concentration distribution, File 2.



(a) 4t (b) 6t (c) 6t

Figure 67: Salt concentration distribution, File 2.

Finally, a quantifiable effect, in terms of flow production and salt production are plotted in Figs. 68 to 73. The injection flow values were a bit exaggerated but these plots show clearly how much salt was produced, and would potentially discern depletion from a zone, rich in salt concentration but low oil saturation in a mature field by the use of a more viscous polymer. Production of water and oil, show clearly that more oil is produced, sooner, with the use of a higher viscosity displacement fluid compared to lower one. And how in file 1 field simulation, oil production rate gradually increases and finally settles at 800 *days*.

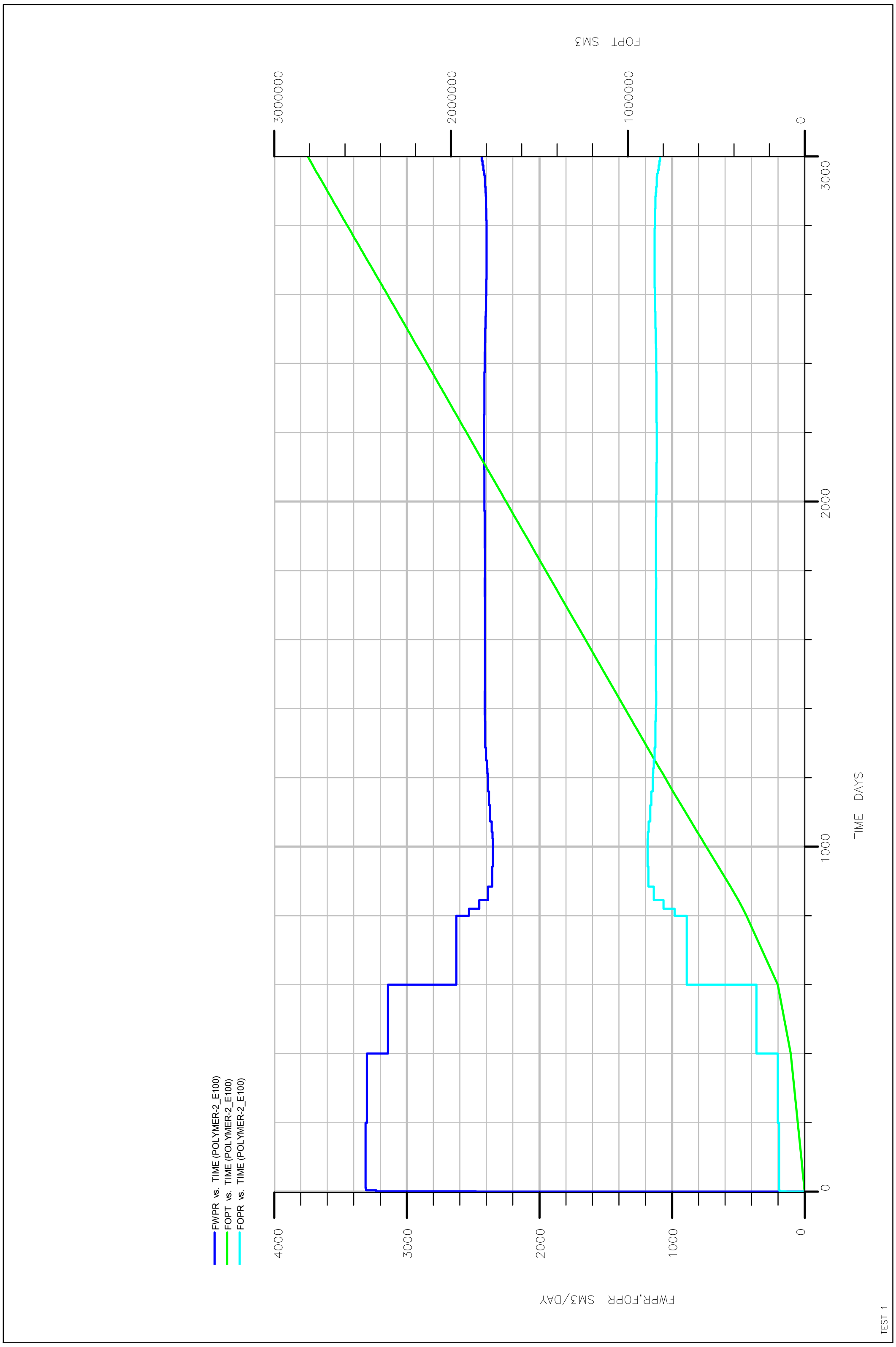


Figure 68: FOPR vs Time, File 1.

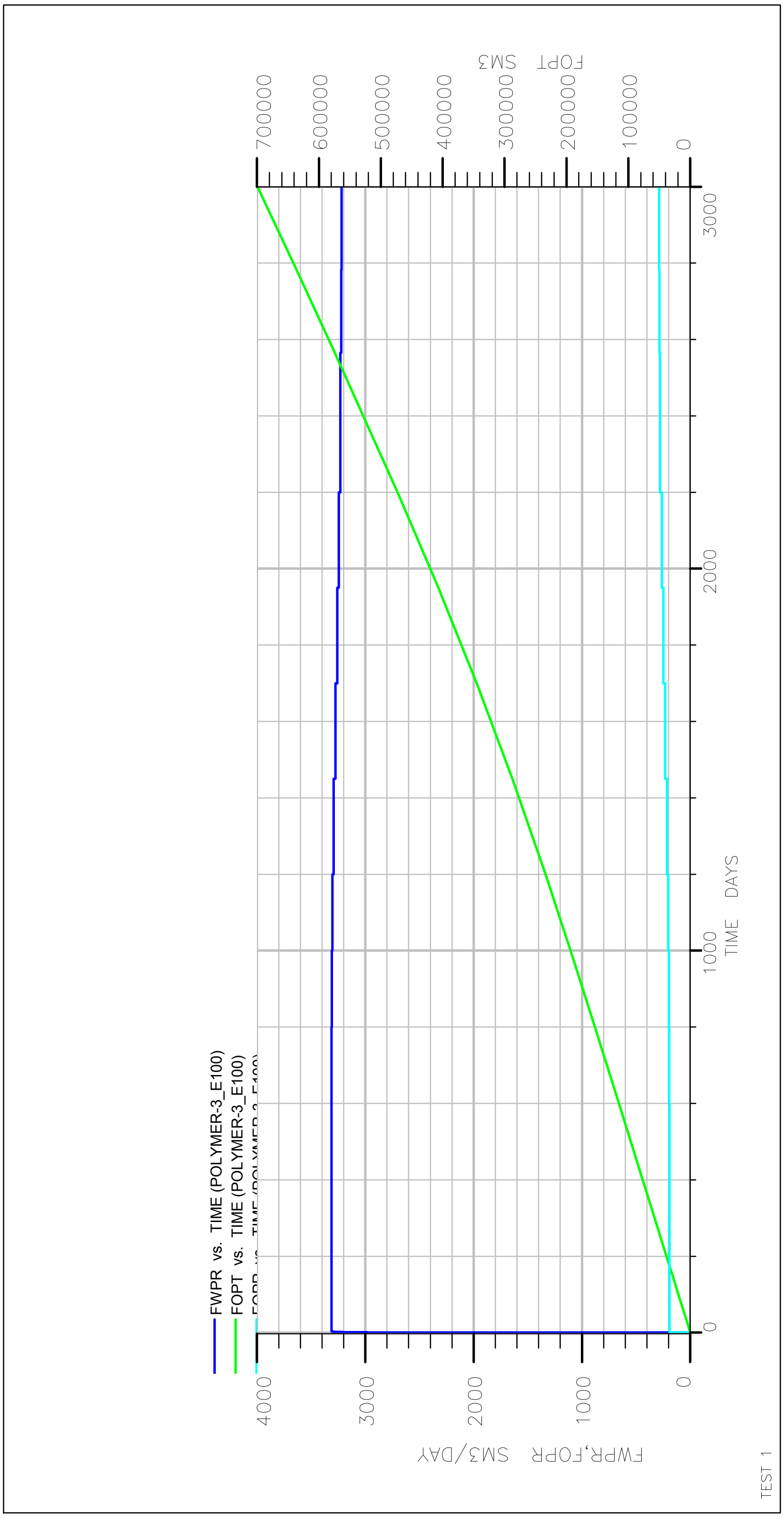


Figure 69: FOPR vs Time, File 2.

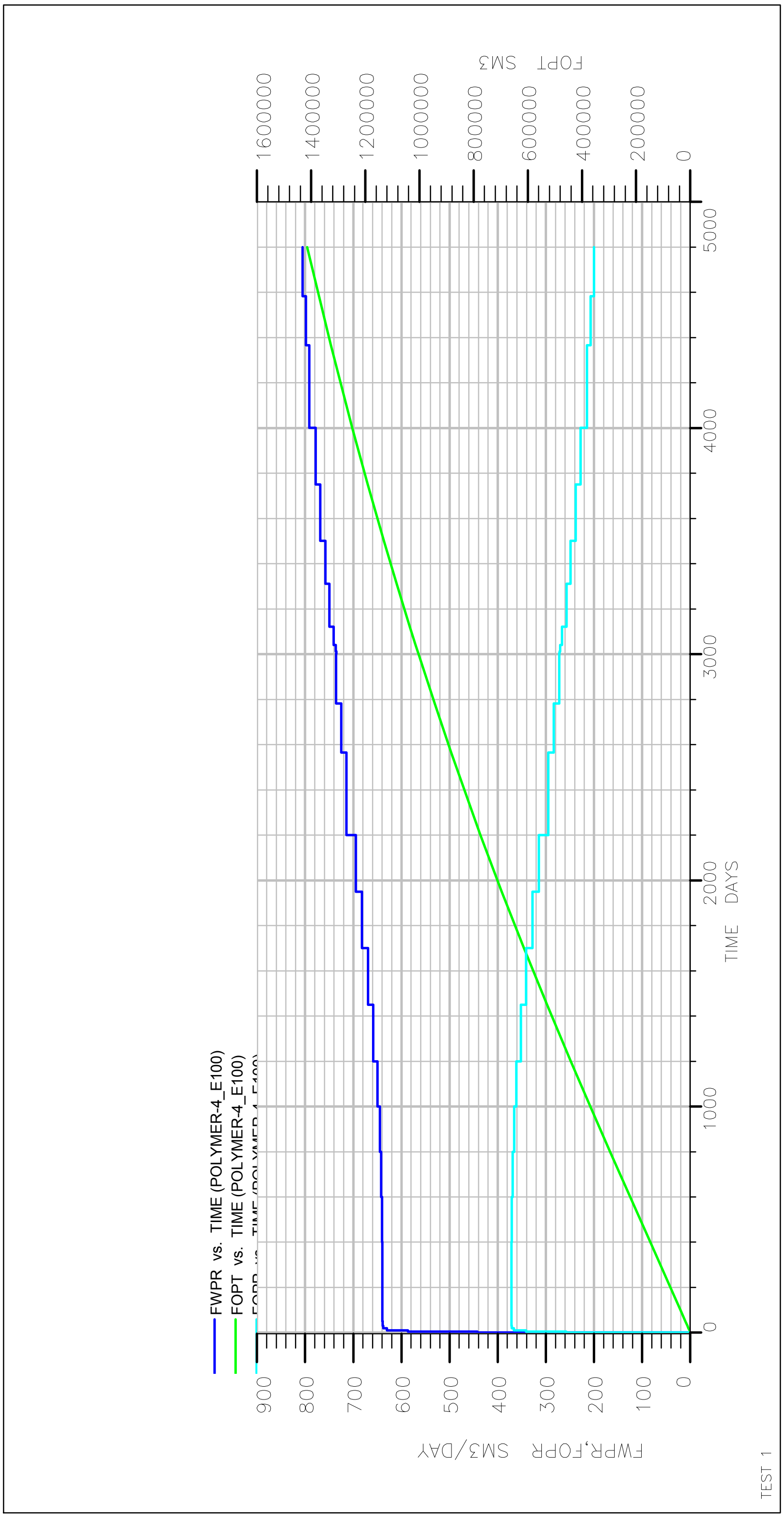


Figure 70: FOPR vs Time, File 3.

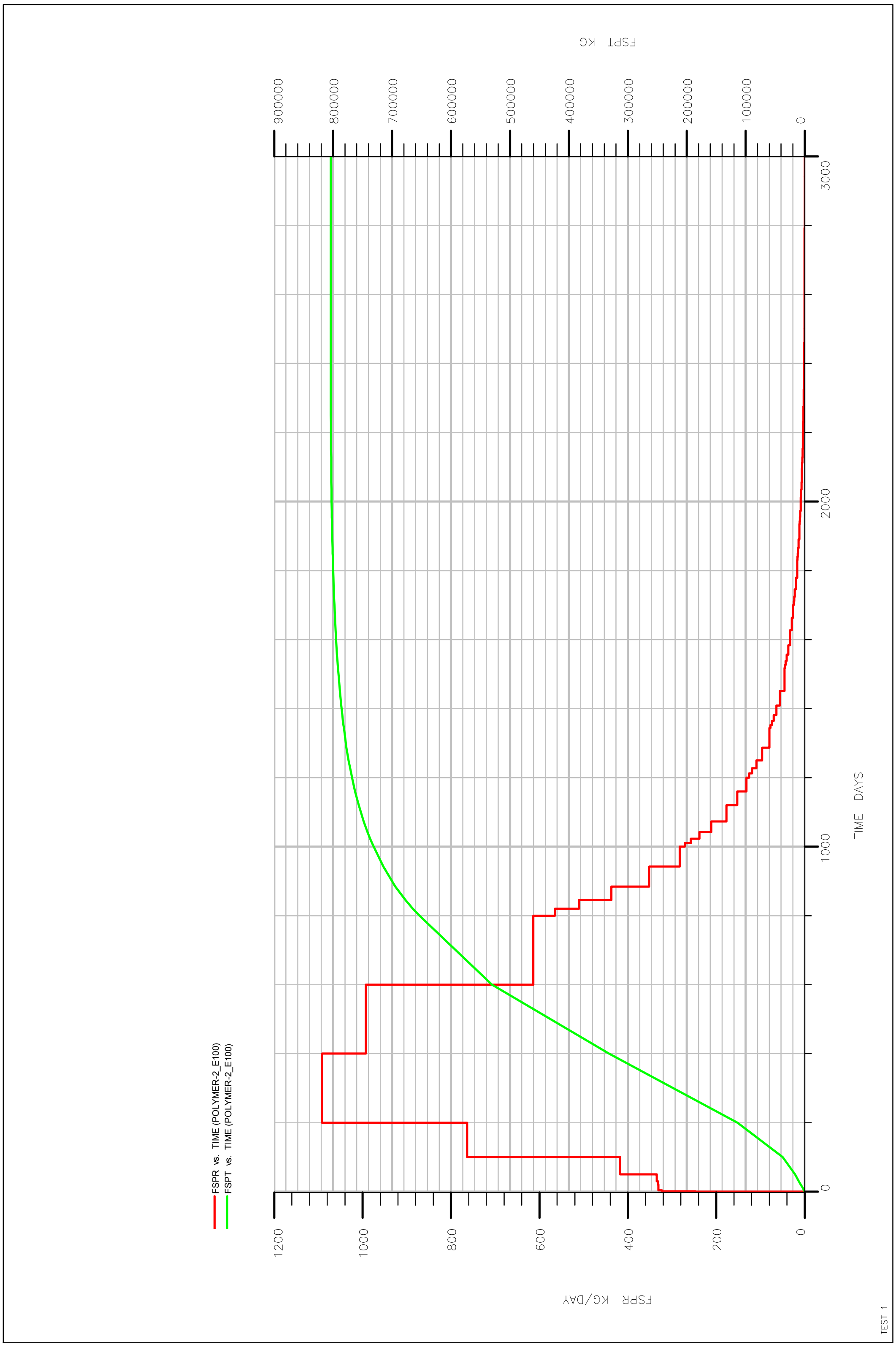


Figure 71: FSPR vs Time, File 1.

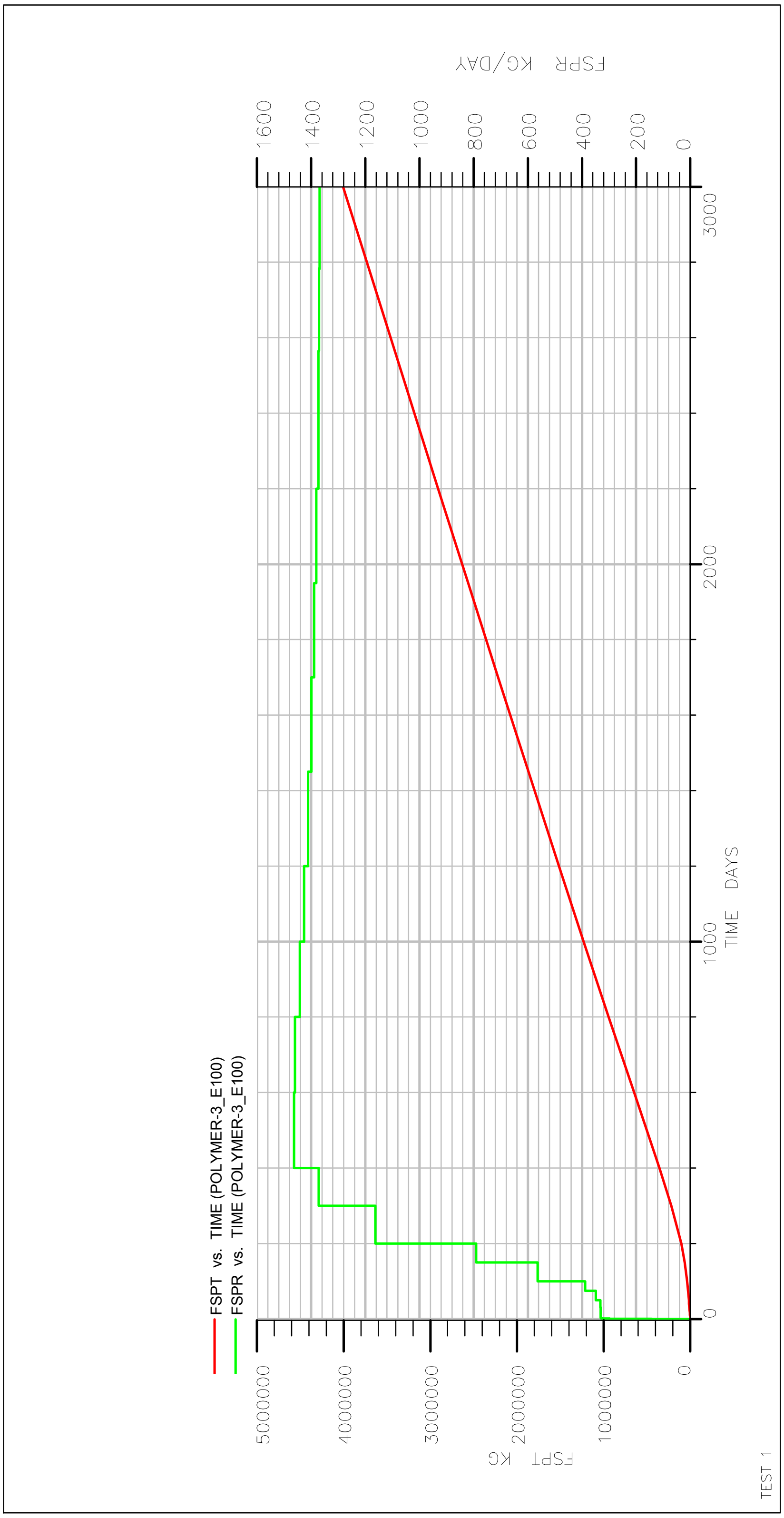


Figure 72: FSPR vs Time, File 2.

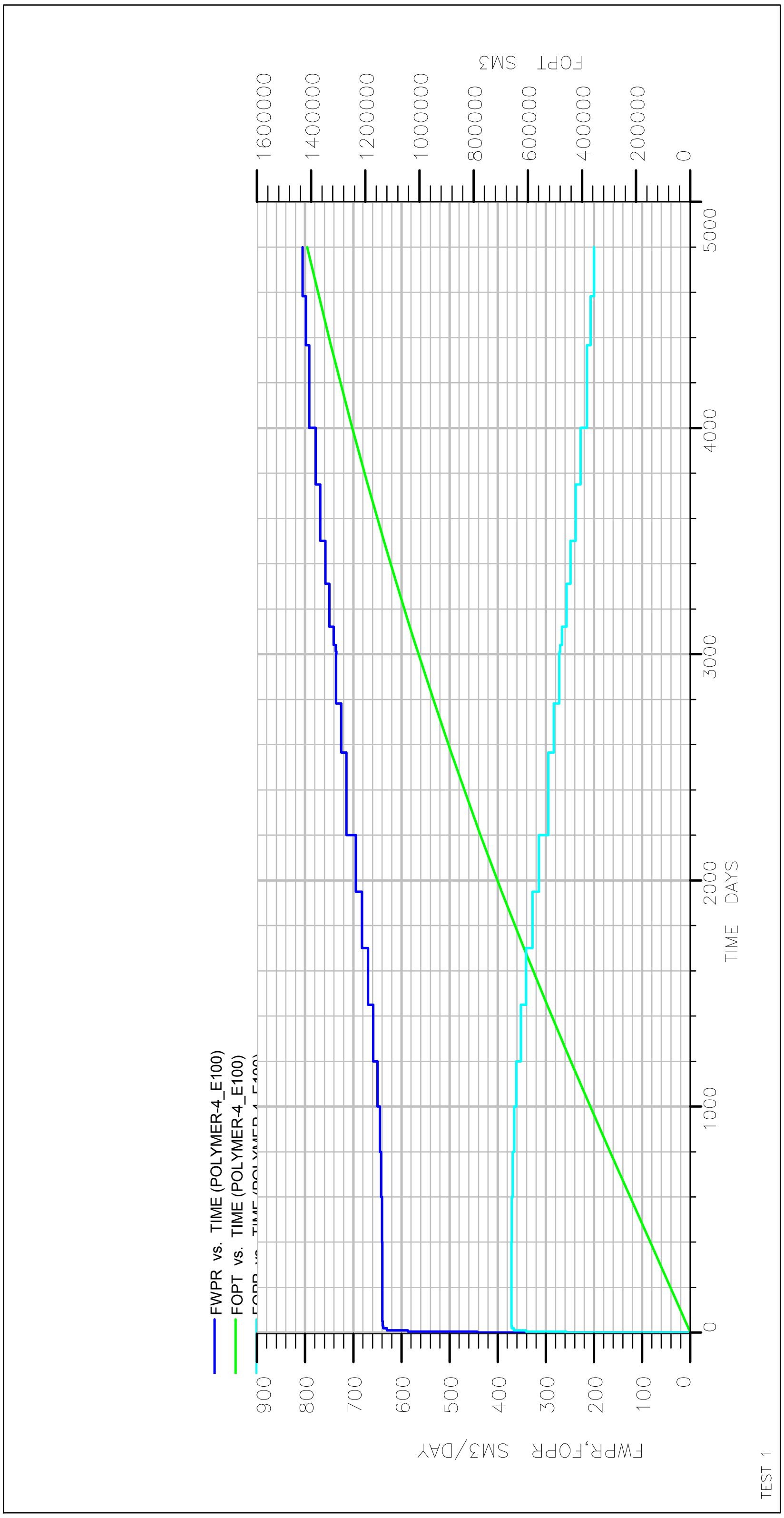


Figure 73: FSPR vs Time, File 3.

9 Discussion

The plots in Fig. 12 show that a variation in the relative permeability curves can greatly affect the mobility of the displacing fluid with unit viscosity. This is corroborated in Table 6. Though at lower viscosity and high time interval, this change does not seem so great, as can be seen in Figs. 58 to 61. The most notable effect is a change in viscosity. These can be found over the whole Section 7.2. Common in all of those is that a higher viscosity has a better coverage of the volume behind the front, its more saturated, and that the top of the front is higher. And the fractional flow curves clearly shows this Fig. 33. For the case of non-Newtonian polymer solution, we can see that the the viscosity gradually decreases as N approaches 1. And at $N = 0.7$ the viscosity gas been considerably reduced. For the Base Case and Low viscosity Base Case, the viscosity is reduced to 20% and $< 10\%$ of the original viscosity. Fig. 24 and Fig. 25 one can clearly see how the front's high is reduced, notice that N only decreased to $N = 0.9$. Lesser N values follow the same pattern. Different patters introduced by the permeability distribution are followed by the injection fluid, where the more viscous fluid is more rigorous with the act Figs. 39 to 41. A look at the pressure distribution shows that the pressure is distributed more evenly for a higher viscous displacing fluid. The only reason to explain this is that a stability in saturation reflects a gradual differential in pressure while a more diverse saturation distribution experiences similar pressure everywhere. This could be confirmed if pressure distribution at different times where provided. The second pattern, given by a wall of sorts, possible shale formation extended over the middle section show the same general results. The more viscous fluid to a higher degree follows the long way out Figs. 42 to 44. Worth noting however is the fact that the right side of the grid system, facing diagonally from the injection to the production well, is filled more than the left side. This is due to the position of the injection well and that flow is easier to the left, from the descirption of Permeability Profile 2 and that the effects is shared for a varied range of viscosity. The third patter, shows that there may be cases in which certain zones may be ignored by the polymer flow solution, not by having different properties but simply because of its position relative to the direction of the injection and blocked off zones Figs. 46 to 49. At lower polymer viscosities this might not reflect the structure of the reservoir but at higher viscosities it makes it clear as the saturation of water produced is considerably higher, sooner in the production stage if that were not the case. As with the purpose of creating a three dimensional system, the plots show that the discrepancy between zones in the vertical direction increases with lower viscosity values, flow being redirected.

Overall, the eclipse results represent the same distribution, and the vertical figures of the grid in Section 8 are in concordance with the matlab results. Additionally, the purpose of including salt was to see the effect of a reduced polymer viscosity but as can be seen from the figures, this salt was flushed out. This was an error by the author as the injection rate, pore volume wise was too high and the only connection between the salt concentrated formation water and injected polymer solution was at the front.

In terms of literature, different polymers are susceptible to different parameters. What affects a polymer solution may not reflect what affects a polymer gel system. There is a new, improved and proved system for conformance gel. It consists of the use of granules that are produced above ground at favorable conditions, bypassing the problems normally encountered, like flow restrictions, concentration of salts or pH changes, etc. The type of bonding in a polymer and gel system has a great impact on gelation times and strength, and there is a reasonable initial guess for how much volume of gel is needed and concentration polymer, based on experiences with different fields presented in according to the problem to be solved, Section 5.

HPAM is one of the most common polymers and is based on acrylamide. The degree of hydrolization has a big impact in its behavior and this type of polymer is used for both polymer flooding operation and gel conformance problems. Where, an organically crosslinked polymer is a suitable choice for low temperature systems, organically crosslinked polymer for higher temperature systems, a good choice will often be HPAM and that to have reliable results in high shear flows, PEI should be used.

Finally, the Matlab code works fine, it has several options and to reduce processing times, the use of the fractional flow equation is preffered to the IMPES solution.

10 Conclusion

The author concludes: 1) That the code created to simulate polymer flow in porous media in Matlab is fully capable of being used for simulations predicting behavior in real fields.
2) A reasonable choice for a polymer for both polymer flooding and gel conformance problems is HPAM as there is a vast network of studies available describing how it varies with different factors.
3) A higher viscosity polymer flows slower and has a bigger impact in the pressure distribution in the reservoir, where a pressure change indicates that the polymer solution is displacing fluid towards the production well. 4) Low viscosity flow aside from reaching the production well sooner, increases the saturation slightly as time passes distributed over the path it has taken.
5) An decrease in the non-Newtonian factor decreases viscosity.
6) Polymer viscosity depends on a variety of parameters and those same responses are not shared by polymer gel systems
7) The Matlab code has a few weaknesses but is nothing that cannot be fixed with a better larger grid distribution or less injection or time intervals.

Nomenclature

Greek Symbols

Symbol	Description	Units
μ	Viscosity	$\frac{kg}{m \cdot s}$
α	Oil, Water	–
β	Constant	–
λ	Mobility	–
μ_0	Newtonian Viscosity of the Fluid	$\frac{kg}{m \cdot s}$
μ_{eff}	Effective Viscosity	$\frac{kg}{m \cdot s}$
∂	Partial Derivative	–
ϕ	Porosity	L
ρ	Density	$\frac{kg}{m^3}$

Roman Symbols

Symbol	Description	Units
$\boldsymbol{k}$	Permeability Vector	m^2
$\boldsymbol{k}_o$	Oil Permeability Vector	m^2
$\boldsymbol{k}_w$	Water/Polymer Solution Permeability Vector	m^2
$\boldsymbol{u_{shear}}$	Shear Velocity	$\frac{m}{s}$
$\boldsymbol{u}$	Velocity Vector	$\frac{m}{s}$
Δ	Variation	m
∇	Gradient, Sum of Partial Derivatives over every Axis	m^2
A	Area	m^2
a	Acceleration	$\frac{m}{s^2}$
F	Force	N
g	Acceleration due to gravity	$\frac{m}{s^2}$

k	Permeability	m^2
k_h	Horizontal Permeability	m^2
k_v	Vertical Permeability	m^2
k_r'	End Point Relative Permeability	–
M	Mobility Ratio	–
m	Mass	kg
n	Non-Newtonian Factor	–
n_o	Oil Non-Newtonian Factor	s/–
n_w	Water/Polymer Solution Non-Newtonian Factor	s/–
P	Pressure	$\frac{N}{m^2}$
P_c	Capillary Pressure	$\frac{N}{m^2}$
P_o	Oil Pressure	$\frac{N}{m^2}$
P_w	Water/Polymer Solution Pressure	$\frac{N}{m^2}$
R_f	Resistance Factor	–
R_{rf}	Residual Resistance Factor	–
S_o	Oil Saturation	–
S_w	Water/Polymer Solution Saturation	–
T	Temperature	$^{\circ}C$
t	Time	s/–
u	Velocity	$\frac{m}{s}$
V	Volume	m^3
X	Dimension X	m
x	Length in the X-Axis	m
Y	Dimension Y	m
Z	Dimension Z	m

Bibliography

Aarnes, J. E., T. Gimse, & K.-A. Lie (2007). An introduction to the numerics of flow in porous media using matlab. *Geometric Modelling, Numerical Simulation, and Optimization*, pp 265–306. http://dx.doi.org/10.1007/978-3-540-68783-2_9.

Al-Muntasheri, G. A., H. A. Nasr-El-Din, & P. L. J. Zitha (2007, February). A study of polyacrylamide based gels crosslinked with polyethyleneimine. SPE-105925-MS. http://dx.doi.org/10.2118/105925-MS.

Argillier, J.-F., A. Dupas, R. Tabary, I. Henaut, P. Poulain, D. Rousseau, & T. Aubry (2013, April). Impact of polymer mechanical degradation on shear and extensional viscosities: Toward better injectivity forecasts in polymer flooding operations. SPE-164083-MS. http://dx.doi.org/10.2118/164083-MS.

Baijal, S. (1975). Interaction during polymer flooding. *Society of Petroleum Engineers*. SPE-5849-MS. `https://www.onepetro.org/general/SPE-5849-MS`.

Baker, R. (1997, April). Reservoir management for waterfloods—part ii. `http://www1.uis.no/Fag/Learningspace_kurs/PetBachelor/webpage/tech%5CReservoir%5Cmobility.pdf`.

Binning, P. & M. A. Celia (1999). Practical implementation of the fractional flow approach to multi-phase flow simulation. http://dx.doi.org/10.1016/S0309-1708(98)00022-0.

Borling, D., K. Chan, T. Hughes, & R. Sydansk (1994, April). Pushing out the oil with conformance control. pp. 44–58. `http://www.slb.com/~/media/Files/resources/oilfield_review/ors94/0494/p44_58.pdf`.

Chauveteau, G., K. Denys, & A. Zaitoun (2002, April). New insight on polymer adsorption under high flow rates. SPE-75183-MS. http://dx.doi.org/10.2118/75183-MS.

Chen, Z., G. Huan, & Y. Ma (2006). *Computational Methods for Multiphase Flows in Porous Media*. Society for Industrial and Applied Mathematics. ISBN=9780898716061.

Christmann, K. (2012). Thermodynamics and kinetics of adsorption. `http://w0.rz-berlin.mpg.de/imprs-cs/download/Vortrag_IMPRS_Schmoeckwitz_Mi_9-11_KChrist.pdf`.

Clark, R. K. (1984). Applications of water-soluble polymers as shale stabilizers in drilling fluids. *American Chemistry Society Volume 213*(Chapter 10), pp 171–181. http://dx.doi.org/10.1021/ba-1986-0213.ch010.

Coste, J.-P., Y. Liu, B. Bai, Y. LI, P. Shen, Z. Wang, & G. Zhu (2000, April). In-depth fluid diversion by pre-gelled particles. laboratory study and pilot testing. SPE-59362-MS. http://dx.doi.org/10.2118/59362-MS.

Dake, L. (1983). Fundamentals of reservoir engineering. *Volume 8*, pp 122. ISBN-9780080568980.

Dang, C. T. Q., Z. J. Chen, N. T. B. Nguyen, W. Bae, & T. H. Phung (2011, September). Development of isotherm polymer/surfactant adsorption models in chemical flooding. SPE-147872-MS. http://dx.doi.org/10.2118/147872-MS.

de Melo, M. A., C. R. Holleben, I. G. Silva, A. de Barros Correia, G. A. Silva, A. J. Rosa, A. G. L. Jr., & J. C. de Lima (2005, June). Evaluation of polymer-injection projects in brazil. SPE-94898-MS. http://dx.doi.org/10.2118/94898-MS.

Fakcharoenphol, P. & Y.-S. Wu (2010, June). Displacement of non-newtonian fluids in linear and radial composite porous media. http://petroleum.mines.edu/research/emg/pon_papers/2010-icpm3.pdf.

Hardy, M., W. Botermans, A. Hamouda, J. Valdal, & J. Warren (1999, February). The first carbonate field application of a new organically crosslinked water shutoff polymer system. SPE-50738-MS. http://dx.doi.org/10.2118/50738-MS.

Holmen, A. (2011). Heterogen katalyse. Kompendium, Course TKP4155, NTNU.

Huh, C., E. Lange, & W. Cannella (1990, April). Polymer retention in porous media. SPE-20235-MS. http://dx.doi.org/10.2118/20235-MS.

Kleppe, J. (2014–2015). Handouts. http://www.ipt.ntnu.no/~kleppe/TPG4150/andhttp://www.ipt.ntnu.no/~kleppe/TPG4160.

Lantz, M. & G. Muniz (2014, April). Conformance improvement using polymer gels: A case study approach. SPE-169072-MS. http://dx.doi.org/10.2118/169072-MS.

Liu, H., H. Han, Z. Li, & B. Wang (2004, March). Granular gel-polymer treatment successful in daqing oilfield. SPE-87071-PA,SPE-87071-MS. http://dx.doi.org/10.2118/87071-MS.

Lockhart, T. (1994, April). Chemical properties of chromium/polyacrylamide gels. SPE-20998-PA. http://dx.doi.org/10.2118/20998-PA.

Moorhouse, R., D. N. Harry, L. Matthews, & U. Merchant (1998, February). Inter-relationships between polymer/crosslinker chemistry and performance in. SPE-39531-MS. http://dx.doi.org/10.2118/39531-MS.

Moradi-Araghi, A. (1999, December). A review of thermally stable gels for fluid diversion in petroleum production. *Journal of Petroleum Science & Engineering*. http://dx.doi.org/10.1016/S0920-4105(00)00015-2.

Needham, R. B. & P. H. Doe (1987, december). Polymer flooding review. SPE-17140-PA. http://dx.doi.org/10.2118/17140-PA.

Norman, C. & J. Smith (1999, May). Economics of in-depth polymer gel processes. SPE-55632-MS. http://dx.doi.org/10.2118/55632-MS.

Ogunberu, A. & K. Asghari (2004, June). Water permeability reduction under flow-induced polymer adsorption. Petroleum Society of Canada. PETSOC-2004-236. http://dx.doi.org/10.2118/2004-236.

Reddy, B., F. Crespo, & L. S. Eoff (2012, April). Water shutoff at ultralow temperatures using organically crosslinked polymer gels. SPE-153155-MS. http://dx.doi.org/10.2118/153155-MS.

Reddy, B., L. Eoff, E. D. Dalrymple, K. Black, D. Brown, & M. Rietjens (2003, June). A natural polymer-based cross-linker system for conformance gel systems. SPE-84937-PA. http://dx.doi.org/10.2118/84937-PA.

Romero-Zeron, L., F. Manalo, & A. Kantzas (2004, February). Characterization of crosslinked gel kinetics and gel strength using nmr. SPE-86548-MS. http://dx.doi.org/10.2118/86548-MS.

Rossen, W. R., A. Venkatraman, R. T. Johns, K. R. Kibodeaux, H. Lai, & N. M. Tehrani (2011). Fractional flow theory applicable to non-newtonian behavior in eor processes. *Transport in Porous Media Volume 89*(Issue 2), pp 213–236. http://dx.doi.org/10.1007/s11242-011-9765-2.

Schneider, F. & W. Owens (1982, February). Steady-state measurements of relative permeability for polymer/oil systems. *Society of Petroleum Engineers Journal 22*. SPE-9408-PA. http://dx.doi.org/10.2118/9408-PA.

Seright, R. (2010, August). Potential for polymer flooding reservoirs with viscous oils. *SPE Reservoir Evaluation & Engineering 13*(04). SPE-129899-PA. http://dx.doi.org/10.2118/129899-PA.

Seright, R. S. & J. M. Seheult (2008, September). Injectivity characteristics of eor polymers. SPE-115142-MS. http://dx.doi.org/10.2118/115142-MS.

Shriwal, P. & R. H. Lane (2012, April). Impacts of timing of crosslinker addition on water shutoff polymer gel properties. SPE-153241-MS. http://dx.doi.org/10.2118/153241-MS.

SI (2006). *The International System of Units (SI).* Bureau International des Poids et Mesures. `http://www.bipm.org/utils/common/pdf/si_brochure_8_en.pdf`.

Sorbie, K. & R. Seright (1992, April). Gel placement in heterogeneous systems with crossflow. SPE-24192-MS. http://dx.doi.org/10.2118/24192-MS.

Sydansk, R. (1988, April). A new conformance-improvement-treatment chromium(iii) gel technology. SPE-17329-MS. http://dx.doi.org/10.2118/17329-MS.

Sydansk, R. D. (2007). Petroleum engineering handbook. ISBN 978-1-55563-120-8.

Wu, Y.-S., K. Pruess, & P. A. Witherspoon (1991, April). Displacement of a newtonian fluid by a non-newtonian fluid in a porous medium. *Transport in Porous Media Volume 6*(Issue 2), pp 115–142. http://dx.doi.org/10.1007/BF00179276.

Yerramilli, S. S., P. L. J. Zitha, & R. C. Yerramilli (2013, June). Novel insight into polymer injectivity for polymer flooding. SPE-165195-MS. http://dx.doi.org/10.2118/165195-MS.

Yu, J. F. S., J. L. Zakin, & G. K. Patterson (2003, March). Mechanical degradation of high molecular weight polymers in dilute solution. *Journal of Applied Polymer Science Volume 23*(Issue 8), pp 2493–2512. http://dx.doi.org/10.1002/app.1979.070230826.

Zaitoun, A. & H. Bertin (1998, April). Two-phase flow property modifications by polymer adsorption. SPE-39631-MS. http://dx.doi.org/10.2118/39631-MS.

Zhang, G. & R. Seright (2014, June). Effect of concentration on hpam retention in porous media. SPE-166265-PA. http://dx.doi.org/10.2118/166265-PA.

Zitha, P. L. (2001, May). In-depth filtration of macromolecules induced by bridging adsorption in porous media. SPE-68980-MS. http://dx.doi.org/10.2118/68980-MS.

A Discretization

The discretization of the equations derived in Section 6 are presented here. A discretization that redefines the equations in term of time step and grid blocks for use in numerical simulations.

We start with the pressure, for the case of variable grid lengths. Variable grid length enables the option to opt for finer grid sizes at specific points where a better accuracy is required, resulting in a varied amount of grid sizes. Usually one would opt for a finer grid alignment over the whole grid extension but that would result in very long time-consuming simulation runs. Opting instead for local sections with finer grid alignment, will however, slightly increase the simulation time and can even be adjusted to neutralize such increase by giving a rougher distribution elsewhere. The following discretization was presented by Kleppe (2015) for one-dimensional flow.

The Taylor expansion of the pressure for the variable grid size at a given time is

$$P_{i+1} = P_i + \frac{\frac{(\Delta x_i + \Delta x_{i+1})}{2}}{1!} P_i' + \frac{(\frac{(\Delta x_i + \Delta x_{i+1})}{2})^2}{2!} P_i'' + \frac{(\frac{(\Delta x_i + \Delta x_{i+1})}{2})^3}{3!} P_i''' \ldots. \tag{74}$$

$$P_{i-1} = P_i + \frac{\frac{-(\Delta x_i + \Delta x_{i+1})}{2}}{1!} P_i' + \frac{(\frac{-(\Delta x_i + \Delta x_{i+1})}{2})^2}{2!} P_i'' + \frac{(\frac{-(\Delta x_i + \Delta x_{i+1})}{2})^3}{3!} P_i''' \ldots. \tag{75}$$

Recall that the pressure equation Eq. (61), assuming no gravitational forces depends on $\boldsymbol{k}$, λ as well as ∇P. Kleppe (2015) suggests representing the factors aside from the pressure as $f(x, y, z)$ and applying a taylor expansion to the whole group, similar to the way as previously done for the pressure Eq. (74) but for a central approximation instead and extended to rectangular coordinates.

$$f(x, y, z) = \boldsymbol{k}\lambda, \qquad -\nabla \cdot (f(x, y, z)\nabla P) = -\frac{\partial \phi}{\partial t} + q_w + q_o. \tag{76}$$

$$[f(x, y, z)\nabla P]_{i+\frac{1}{2}} = [f(x, y, z)\nabla P]_i + \frac{(\frac{\Delta x_i}{2})}{1!} \nabla [f(x, y, z)\nabla P]_i + \frac{(\frac{\Delta x_i}{2})^2}{2!} \nabla^2 [f(x, y, z)\nabla P]_i \ldots. \tag{77}$$

$$[f(x, y, z)\nabla P]_{i-\frac{1}{2}} = [f(x, y, z)\nabla P]_i + \frac{(\frac{-\Delta x_i}{2})}{1!} \nabla [f(x, y, z)\nabla P]_i + \frac{(\frac{-\Delta x_i}{2})^2}{2!} \nabla^2 [f(x, y, z)\nabla P]_i \ldots. \tag{78}$$

which solved for the divergence of $[f(x, y, z)\nabla P]_i$ is

$$\nabla [f(x, y, z)\nabla P]_i = \frac{[f(x, y, z)\nabla P]_{i+\frac{1}{2}} - [f(x, y, z)\nabla P]_{i-\frac{1}{2}}}{\Delta x_i} + 0(\Delta x^2) \tag{79}$$

Similarly one can express the divergence of P

$$\nabla P_{i+\frac{1}{2}} = \frac{P_{i+1} - P_i}{(\Delta x_i + \Delta x_{i+1})/2} + 0(\Delta x) \tag{80}$$

$$\nabla P_{i-\frac{1}{2}} = \frac{P_i - P_{i-1}}{(\Delta x_i + \Delta x_{i-1})/2} + 0(\Delta x) \tag{81}$$

and inserting into Eq. (79) gives

$$\nabla[f(x,y,z)\nabla P]_i = \frac{2f(x,y,z)_{i+\frac{1}{2}}\frac{(P_{i+1}-P_i)}{(\Delta x_i+\Delta x_{i+1})} - 2f(x,y,z)_{i-\frac{1}{2}}\frac{(P_i-P_{i-1})}{(\Delta x_i+\Delta x_{i-1})}}{\Delta x_i} + 0(\Delta x) \tag{82}$$

with a first order approximation error.

But this covers only the Newtonian characteristics of the displacing fluid. As presented in Eq. (63), a similar discretization has to be done for the divisor on the left side. As this value alters the flow potential between adjacent blocks, we use the pressure of the blocks in question, making use of Eq. (80) as we define the first flow potential at the edge $i = 1$ with pressure values starting at $i = 2$ inside the reservoir. Thus, Eq. (82) is expanded to

$$\nabla[f(x,y,z)\nabla P]_i = \frac{\frac{2f(x)_{i+\frac{1}{2}}\frac{(P_{i+1}-P_i)}{(\Delta x_i+\Delta x_{i+1})} - 2f(x)_{i-\frac{1}{2}}\frac{(P_i-P_{i-1})}{(\Delta x_i+\Delta x_{i-1})}}{\Delta x_i}}{n\left|2f(x,y,z)_i\frac{P_{i+1}-P_i}{(\Delta x_i+\Delta x_{i+1})}\right|^{1-1/n}} \tag{83}$$

The above equation applies to all 3 dimensions of the rectangular coordinates, just by replacing x and i with y and j or z and k.

Next is the discretization of $f(x,y,z)_{i\pm1}$. Kleppe (2015) suggest using the average total permeability of the blocks involved, which in x-direction would be i and $i+1$ with the an upwind mobility term. The reasoning behind the latter is that the average mobility between blocks has to account for both blocks. If the block with exiting flow has reached it's residual saturation, flow out of that block will not stop unless the next block also has reached residual saturation. This produces inconsistencies and in this particular matlab program, it will even create oscilation problems which renders it useless. Much of this cause is because of the total amount of gridblocks selected, but as time is of utmost importance, these undesired effects would increase with decreasing processing times. Though it may be possible to bypass this by introducing an "if" statement taking into account that scenario and for that particular case use the upwind mobility term instead, however this was not programmed.

The permeability average between blocks is derived from Darcy's equation (8). Placing all the position terms to the left gives us

$$\boldsymbol{u}\frac{\partial x}{\boldsymbol{k}} = \frac{\partial P}{\mu}, \tag{84}$$

which by integrating

$$\boldsymbol{u}\int_i^{i+1}\frac{\partial x}{\boldsymbol{k}} = \int\frac{\partial P}{\mu} \tag{85}$$

then solving for the left side

$$\boldsymbol{u}\int_i^{i+1}\frac{\partial x}{\boldsymbol{k}} = \boldsymbol{u}\frac{\left(\frac{\Delta x_i}{\boldsymbol{k}_i}+\frac{\Delta x_{i+1}}{\boldsymbol{k}_{i+1}}\right)}{2} \tag{86}$$

and defining an average permeability $\bar{\boldsymbol{k}}$ gives

$$\boldsymbol{u}\frac{\left(\frac{\Delta x_i}{\boldsymbol{k}_i}+\frac{\Delta x_{i+1}}{\boldsymbol{k}_{i+1}}\right)}{2} = \boldsymbol{u}\frac{\frac{(\Delta x_i+\Delta x_{i+1})}{2}}{\bar{\boldsymbol{k}}} \tag{87}$$

$$\bar{\boldsymbol{k}} = \boldsymbol{k}_{\boldsymbol{i}\pm\frac{\mathbf{1}}{\mathbf{2}}} = \frac{(\Delta x_i + \Delta x_{i\pm1})}{\left(\frac{\Delta x_i}{\boldsymbol{k}_i}+\frac{\Delta x_{i\pm1}}{\boldsymbol{k}_{i\pm1}}\right)}. \tag{88}$$

The upwind mobility term is defined as follow

$$\lambda_{i\pm\frac{1}{2}} = \begin{cases} \lambda_i & \text{for} \quad P_i > P_{i\pm1} \\ \lambda_{i\pm1} & \text{for} \quad P_i \leq P_{i\pm1} \end{cases} \tag{89}$$

Having discretizised the permeability and mobility, we define the factors accompanying the pressure in equation (82) as transmissibility $T_{r\beta}$ for $\beta = x, y, z$. We include all three coordinates and express it as follows

$$\pm 2f(x,y,z)_{\left(i\pm\frac{1}{2},j,k\right)}\frac{\left(P_{(i\pm1,j,k)}-P_{(i,j,k)}\right)}{(\Delta x_i+\Delta x_{i\pm1})} = \pm 2\left(\boldsymbol{k}\lambda\right)_{\left(i\pm\frac{1}{2},j,k\right)}\frac{\left(P_{(i\pm1,j,k)}-P_{i,j,k}\right)}{(\Delta x_i+\Delta x_{i\pm1})} \tag{90}$$

$$= \pm 2\frac{(\Delta x_i+\Delta x_{i\pm1})}{\left(\frac{\Delta x_i}{\boldsymbol{k}_{(i,j,k)}}+\frac{\Delta x_{i\pm1}}{\boldsymbol{k}_{(i\pm1,j,k)}}\right)}\lambda_{\left(i\pm\frac{1}{2},j,k\right)}\frac{\left(P_{(i\pm1,j,k)}-P_{(i,j,k)}\right)}{(\Delta x_i+\Delta x_{i\pm1})} \tag{91}$$

$$= \pm T_{rx_{\left(i+\frac{1}{2},j,k\right)}}\left(P_{(i\pm1,j,k)}-P_{(i,j,k)}\right) \tag{92}$$

where the dividend of Eq. (83) that accounts for non-Newtonian behavior is taken from the previous timestep, that makes use of the absolute value of the velocity between two adjacent blocks in a given axis, expressed as follows

$$n\left|2f(x,y,z)^{t-\Delta t}_{\left(i+\frac{1}{2},j,k\right)}\frac{P^{t-\Delta t}_{(i+1,j,k)}-P^{t-\Delta t}_{(i,j,k)}}{(\Delta x_i+\Delta x_{i+1})}\right|^{1-1/n} = n\left|T^{t-\Delta t}_{rx_{\left(i+\frac{1}{2},j,k\right)}}\left(P^{t-\Delta t}_{(i+1,j,k)}-P^{t-\Delta t}_{(i,j,k)}\right)\right|^{1-1/n}. \tag{93}$$

This completes the discretization of the left side, of the *pressure equation.*

The discretization of the *saturation equation* covers Eq. (69) in terms of saturation progress with time and variation of fractional flow with saturation as saturation progresses with position along the different coordinates. This is expressed with timesteps, with $f_{w_{(i,j,k)}}$ taken upstream, as with the viscosity, and with the velocity taken in the intersection between the blocks, which in the X-Axis is expressed as $\boldsymbol{u}_{\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}} = \boldsymbol{u}_{\boldsymbol{i}\pm\frac{\mathbf{1}}{\mathbf{2}},\boldsymbol{j},\boldsymbol{k}}$

$$f_{w_{(i,j,k)}} = \begin{cases} f_{w_{\left(i-\frac{1}{2},j,k\right)}} & \text{for} \quad \boldsymbol{u}_{\left(i-\frac{1}{2},j,k\right)}+\boldsymbol{u}_{\left(i+\frac{1}{2},j,k\right)} >= 0 \\ f_{w_{\left(i+\frac{1}{2},j,k\right)}} & \text{for} \quad \boldsymbol{u}_{\left(i-\frac{1}{2},j,k\right)}+\boldsymbol{u}_{\left(i+\frac{1}{2},j,k\right)} < 0. \end{cases} \tag{94}$$

Eq. (94) is the Standard way the program calculates the saturation update, Option 1. Alternatively, the direct way, from Eq. (69), Option 2 consists of Eq. (72).

The *saturation equation* then takes the form of

$$\begin{aligned}
\frac{\phi_{(i,j,k)}}{\Delta x_i \Delta y_j \Delta z_k} \frac{\left(S^{t+\Delta t}_{w_{(i,j,k)}} - S^{t}_{w_{(i,j,k)}}\right)}{\Delta t} &- \left(f_{w_{x_{(i,j,k)}}} \boldsymbol{u}_{\left(i-\frac{1}{2},j,k\right)} - f_{w_{x_{(i+1,j,k)}}} \boldsymbol{u}_{\left(i+\frac{1}{2},j,k\right)}\right) && (95)\\
&- \left(f_{w_{y_{(i,j,k)}}} \boldsymbol{u}_{\left(i,j-\frac{1}{2},k\right)} - f_{w_{y_{(i,j+1,k)}}} \boldsymbol{u}_{\left(i,j+\frac{1}{2},k\right)}\right) && (96)\\
&- \left(f_{w_{z_{(i,j,k)}}} \boldsymbol{u}_{\left(i,j,k-\frac{1}{2}\right)} - f_{w_{z_{(i,j,k+1)}}} \boldsymbol{u}_{\left(i,j,k+\frac{1}{2}\right)}\right) && (97)\\
&= \frac{q_{w_{(i,j,k)}} f_{o_{(i,j,k)}}}{\rho_w} + \frac{q_{o_{(i,j,k)}} f_{w_{(i,j,k)}}}{\rho_o}, && (98)
\end{aligned}$$

and expressed as in the matlab code gives

$$\begin{aligned}
S^{t+\Delta t}_{w_{(i,j,k)}} =& S^{t}_{w_{(i,j,k)}} && (99)\\
&+ \left(f_{w_{x_{(i,j,k)}}} \boldsymbol{u}_{\left(i-\frac{1}{2},j,k\right)} - f_{w_{x_{(i+1,j,k)}}} \boldsymbol{u}_{\left(i+\frac{1}{2},j,k\right)}\right) \frac{\Delta t}{\phi_{(i,j,k)} \Delta x_i \Delta y_j \Delta z_k} && (100)\\
&+ \left(f_{w_{y_{(i,j,k)}}} \boldsymbol{u}_{\left(i,j-\frac{1}{2},k\right)} - f_{w_{y_{(i,j+1,k)}}} \boldsymbol{u}_{\left(i,j+\frac{1}{2},k\right)}\right) \frac{\Delta t}{\phi_{(i,j,k)} \Delta x_i \Delta y_j \Delta z_k} && (101)\\
&+ \left(f_{w_{z_{(i,j,k)}}} \boldsymbol{u}_{\left(i,j,k-\frac{1}{2}\right)} - f_{w_{z_{(i,j,k+1)}}} \boldsymbol{u}_{\left(i,j,k+\frac{1}{2}\right)}\right) \frac{\Delta t}{\phi_{(i,j,k)} \Delta x_i \Delta y_j \Delta z_k} && (102)\\
&+ \left(\frac{q_{w_{(i,j,k)}}}{\rho_w} + \frac{q_{o_{(i,j,k)}}}{\rho_o} - \left[f_w \frac{q_{w_{(i,j,k)}}}{\rho_w}\right]\right) \frac{\Delta t}{\phi_{(i,j,k)} \Delta x_i \Delta y_j \Delta z_k}, && (103)
\end{aligned}$$

with what is inside the square brackets included in the first three fractional flow differentials, for Option 1, and as it is only relevant in the injection block, ignored completely.

B Files Matlab

File name: inputxyz.txt

```
% INPUT Values

%                  m             m        m             m           m             m
%                  ZoneLength-X  DeltaX-i ZoneLength-Y  DeltaY-i    ZoneLength-Z  DeltaZ-i
% 1-Length X [m]
500                100           25       100           25          100           20
% 2-Zones X
3                  150           25       150           25
% 3-Length Y [m]
500                250           25       250           25
% 4-Zones Y
3
% 5-Length Z [m]
100
% 6-Zones Z
1
% 7-Density OIL in [kg/m^3]
970
% 8-Density WATER in [kg/m^3]
970
% 9-Density POLYMER in [kg/m^3]
970
% 10-Factor NO for OIL
1
% 11-Factor N for POLYMER
%0.5
%0.6
%0.7
0.8
%0.9
%1
% 12-OIL injection in [m/s]
0
% 13-WATER injection in [m/s] x-direction
%7.1E-7
%25
%1
%0.5
%0.1
%0.05
%0.01
%3.75
%7.5
0.001
%0.0001
%0.00005
%0.000025
%0.00001
% 14-POLYMER injection in [m/s]
0
```

```
% 15-Viscosity OIL [cp]
%60
6
%100
% 16-Viscosity WATER [cp]
%0.75
6
%1
%5.6
%6
%15
%25
%7.5
%50
%75
%100
%1000
% 17-Viscosity POLYMER [cp]
%750
%60
6
%1
%0.6
% 18-Delta Time in seconds [s]
%1
%2.5E4
2500
%1250
%250
%Type of viscosity Wu[with velocity effects] = 1,
%                  Wu[without velocity effects] = 0,
%                  Injection of water instead = Other number.
%0
1
```

File name: testingxyz.m

```
%M = dlmread('input.txt');
FID=fopen('inputxyz.txt','rt');
a=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('porosity.txt','rt');
b=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('permeabilityx.txt','rt');
c=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('permeabilityy.txt','rt');
d=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);
```

```
FID=fopen('permeabilityz.txt','rt');
e=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('saturationw.txt','rt');
f=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('relativew.txt','rt');
g=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('saturationw2.txt','rt');
SATZ2=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('porosity2.txt','rt');
PORZ2=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('permeabilityx2.txt','rt');
PERMXZ2=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('permeabilityy2.txt','rt');
PERMYZ2=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

FID=fopen('permeabilityz2.txt','rt');
PERMZZ2=textscan(FID,'%f%f%f%f%f%f%f%f%f%f%f','CommentStyle','%');
fclose(FID);

M1=a{1};
M2=a{2};
M3=a{3};
M4=a{4};
M5=a{5};
M6=a{6};
M7=a{7};
%M8=a{8};
%M9=a{9};
%M10=a{10};
%M11=a{11};
%M12=a{12};
%M13=a{13};
%M14=a{14};
%M15=a{15};
%M16=a{16};
%M17=a{17};
%M18=a{18};

N1=b{1};
N2=b{2};
N3=b{3};
N4=b{4};
```

```
N5=b{5};
N6=b{6};
N7=b{7};
N8=b{8};
N9=b{9};
N10=b{10};

O1=c{1};
O2=c{2};
O3=c{3};
O4=c{4};
O5=c{5};
O6=c{6};
O7=c{7};
O8=c{8};
O9=c{9};
O10=c{10};

P1=d{1};
P2=d{2};
P3=d{3};
P4=d{4};
P5=d{5};
P6=d{6};
P7=d{7};
P8=d{8};
P9=d{9};
P10=d{10};

Q1=e{1};
Q2=e{2};
Q3=e{3};
Q4=e{4};
Q5=e{5};
Q6=e{6};
Q7=e{7};
Q8=e{8};
Q9=e{9};
Q10=e{10};

R1=f{1};
R2=f{2};
R3=f{3};
R4=f{4};
R5=f{5};
R6=f{6};
R7=f{7};
R8=f{8};
R9=f{9};
R10=f{10};

S1=g{1};
S2=g{2};
S3=g{3};
S4=g{4};
S5=g{5};
```

```
S6=g{6};
S7=g{7};
S8=g{8};
S9=g{9};
S10=g{10};

LengthX = M1(1); % In meters [m]

ZonesX  = M1(2); % How many zones the reservoir is divided on.

LengthY = M1(3); % In meters [m]

ZonesY  = M1(4); % How many zones the reservoir is divided on.

LengthZ = M1(5); % In meters [m]

ZonesZ  = M1(6); % How many zones the reservoir is divided on.

DensityOIL = M1(7); % [kg/m^3]
DensityWATER = M1(8); % [kg/m^3]
DensityPOLYMER = M1(9); % [kg/m^3]

FactorNO = M1(10);    % "no" in case the oil is non newtonian
FactorN = M1(11);    % "n" indicating how non-newtonian the viscosity
                    % of the polymer is

qOIL = M1(12);
qWATER = M1(13);
qPOLYMER = M1(14);

ViscosityOIL = M1(15);
ViscosityWATER = M1(16);
ViscosityPOLYMER = M1(17);

DeltaT = M1(18);

Type = M1(19);

for i=1:ZonesX
    LengthzoneX(i)=M2(i); % [m]
    LengthzoneX(i+1)=0;   % Means nothing, its just something i needed
                          % further down.
end

for i=1:ZonesX
    DeltaX(i)=M3(i); % [m]
end

for i=1:ZonesY
    LengthzoneY(i)=M4(i); % [m]
    LengthzoneY(i+1)=0;   % Means nothing, its just something i needed
                          % further down.
end

for i=1:ZonesY
    DeltaY(i)=M5(i); % [m]
```

```
end

for i=1:ZonesZ
    LengthzoneZ(i)=M6(i); % [m]
    LengthzoneZ(i+1)=0;   % Means nothing, its just something i needed
                          % further down.
end

for i=1:ZonesZ
    DeltaZ(i)=M7(i); % [m]
end

%FOR ZONE Z = 1

for i=1:ZonesY
    N(1,i) = N1(i);
    N(2,i) = N2(i);
    N(3,i) = N3(i);
    N(4,i) = N4(i);
    N(5,i) = N5(i);
    N(6,i) = N6(i);
    N(7,i) = N7(i);
    N(8,i) = N8(i);
    N(9,i) = N9(i);
    N(10,i) = N10(i);

    O(1,i) = O1(i);
    O(2,i) = O2(i);
    O(3,i) = O3(i);
    O(4,i) = O4(i);
    O(5,i) = O5(i);
    O(6,i) = O6(i);
    O(7,i) = O7(i);
    O(8,i) = O8(i);
    O(9,i) = O9(i);
    O(10,i) = O10(i);

    P(1,i) = P1(i);
    P(2,i) = P2(i);
    P(3,i) = P3(i);
    P(4,i) = P4(i);
    P(5,i) = P5(i);
    P(6,i) = P6(i);
    P(7,i) = P7(i);
    P(8,i) = P8(i);
    P(9,i) = P9(i);
    P(10,i) =P10(i);

    Q(1,i) = Q1(i);
    Q(2,i) = Q2(i);
    Q(3,i) = Q3(i);
    Q(4,i) = Q4(i);
    Q(5,i) = Q5(i);
    Q(6,i) = Q6(i);
    Q(7,i) = Q7(i);
    Q(8,i) = Q8(i);
    Q(9,i) = Q9(i);
    Q(10,i) = Q10(i);

    R(1,i) = R1(i);
    R(2,i) = R2(i);
    R(3,i) = R3(i);
    R(4,i) = R4(i);
    R(5,i) = R5(i);
    R(6,i) = R6(i);
    R(7,i) = R7(i);
    R(8,i) = R8(i);
    R(9,i) = R9(i);
    R(10,i) = R10(i);

    S(1,i) = S1(i);
    S(2,i) = S2(i);
    S(3,i) = S3(i);
    S(4,i) = S4(i);
    S(5,i) = S5(i);
    S(6,i) = S6(i);
    S(7,i) = S7(i);
    S(8,i) = S8(i);
    S(9,i) = S9(i);
    S(10,i) = S10(i);
end

for i=1:ZonesX
    for j=1:ZonesY
        Porosity(i,j,1) = N(i,j);
        PermeabilityX(i,j,1) = O(i,j)*1E-12;
        PermeabilityY(i,j,1) = P(i,j)*1E-12;
        PermeabilityZ(i,j,1) = Q(i,j)*1E-12;
        SaturationW(i,j,1) = R(i,j);
        SaturationO(i,j,1) = 1-R(i,j);
        RelativeKW(i,j,1) = S(i,j);
        RelativeKO(i,j,1) = 1-S(i,j);
    end
end
```

File name: transmissibilityxyz1.m

```
%calculations of coefficients for transmissibility

% Shorting down the variables
DO = DensityOIL;
DW = DensityWATER;
N  = FactorN;

% Here we assign DX(i,j,k) for the whole length of the reservoir
% Here we assign KX(i,j,k) for the whole length of the reservoir
% We define KXO(i,j,k) as the relative permeability of OIL
% We define KXW(i,j,k) as the relative permeability of WATER
% DX(j) stands for "delta x" at position "j"
% KXYZ(j) stands for "permeability x" at position "j"
% "i" indicates which reservoir extension to extract from
% SumLength = Sum of the Length of the Reservoir by small X intervals DX(i)
SumLengthX = 0;
SumLengthY = 0;
SumLengthZ = 0;
j = 1;
k = 1;
l = 1;
asd=0;
% TotalLength will be used as the sum of the partial lengths of the
% reservoir covered starting from the first interval
TotalLengthX = LengthzoneX(1);
TotalLengthY = LengthzoneY(1);
TotalLengthZ = LengthzoneZ(1);
for i=1:ZonesX

    while SumLengthX < TotalLengthX

        for i2=1:ZonesY

            while SumLengthY < TotalLengthY

                for i3=1:ZonesZ

                    while SumLengthZ < TotalLengthZ
                         Por(j,k,l) = Porosity(i,i2,i3);

                         DX(j,k,l) = DeltaX(i);
                         DY(j,k,l) = DeltaY(i2);
                         DZ(j,k,l) = DeltaZ(i3);

                         KX(j,k,l) = PermeabilityX(i,i2,i3);
                         KY(j,k,l) = PermeabilityY(i,i2,i3);
                         KZ(j,k,l) = PermeabilityZ(i,i2,i3);

                         KXO(j,k,l) = KX(j,k,l);
                         KXW(j,k,l) = KX(j,k,l);
                         KYO(j,k,l) = KY(j,k,l);
                         KYW(j,k,l) = KY(j,k,l);
                         KZO(j,k,l) = KZ(j,k,l);
                         KZW(j,k,l) = KZ(j,k,l);

                         SO(j,k,l) = SaturationO(i,i2,i3);
                         SW(j,k,l) = SaturationW(i,i2,i3);


                         % Calculating Mobility Term for OIL
                         ViscTermOIL(j,k,l) = 1. / ViscosityOIL *1000;

                         % Calculating Mobility Term for WATER
                         ViscTermWATER(j,k,l) = 1. / ViscosityWATER*1000;

                         % Calculating Mobility Term for POLYMER
                         %ViscTermPOLYMER(j,k,l) = 1 / ViscosityPOLYMER2(j,k,l)*1000;
```

```
          % I will use this part to initialize the injection
          % of oil, water and polymer for j>1
          if j>1 || k>1
            qO(j,k,l) = 0;
            qW(j,k,l) = 0;
            qP(j,k,l) = 0;
          end

          % Progressing the loop
          SumLengthZ = SumLengthZ + DZ(j,k,l);
          l = l + 1;
          suml=l;
          end

          % Adds the next interval
          TotalLengthZ = TotalLengthZ + LengthzoneZ(i3+1);
          % Calculates the total amount of DZ at the end of each zone
          TotalDZ(i3) = l;
        end
        l=1;
        SumLengthZ=0;
        TotalLengthZ=LengthzoneZ(1);


        % Progressing the loop
        SumLengthY = SumLengthY + DY(j,k,l);
        k = k + 1;
        sumk=k;
      end

        % Adds the next interval
        TotalLengthY = TotalLengthY + LengthzoneY(i2+1);
        % Calculates the total amount of DY at the end of each zone
        TotalDY(i2) = k;
    end
    k=1;
    SumLengthY=0;
    TotalLengthY=LengthzoneY(1);

    % Progressing the loop
    SumLengthX = SumLengthX + DX(j,k,l);
    j = j + 1;
    sumj=j;

    end

    % Adds the next interval
    TotalLengthX = TotalLengthX + LengthzoneX(i+1);
    % Calculates the total amount of DX at the end of each zone
    TotalDX(i) = j;

end

GridX=size(DX,1);
```

```
GridY=size(DY,2);
GridZ=size(DZ,3);

%Diversity in permeability in the vertical direction for crossflow/no
%crossflow
%increasing the permeability tenfold

% Z_Counter = floor(GridZ/2)+1; %round(GridZ/2);
% %  crossflow make 2 runs
%
% %Z_Counter=2;
% Z_Counter1=1;
% AMOUNT = 100;
% AMOUNT2 = 1E13/3;%400;%1000;%1;%10000000000000/3;
% for i=1:GridX
%     for j=1:GridY
%         for k=Z_Counter:GridZ
%             KXO(i,j,k) = AMOUNT * KXO(i,j,k); % X-Direction
%             KXW(i,j,k) = AMOUNT * KXW(i,j,k); % X-Direction
%             KYO(i,j,k) = AMOUNT * KYO(i,j,k); % Y-Direction
%             KYW(i,j,k) = AMOUNT * KYW(i,j,k); % Y-Direction
%             %KZO(i,j,k) = AMOUNT * KZO(i,j,k); % Z-Direction
%             %KZW(i,j,k) = AMOUNT * KZW(i,j,k); % Z-Direction
%         end
%     end
% end
% for i=1:GridX
%     for j=1:GridY
%         for k=Z_Counter1:GridZ
%             KZO(i,j,k) = AMOUNT2 * KZO(i,j,k); % Z-Direction
%             KZW(i,j,k) = AMOUNT2 * KZW(i,j,k); % Z-Direction
%         end
%     end
% end

%%%%NEW SYSTEM%%%

% Z_Counter = floor(GridZ/2)+1; %round(GridZ/2);
% %  crossflow make 2 runs
%
% %Z_Counter=2;
% Z_Counter1=1;
% AMOUNT = 100;
% AMOUNT2 = 1E13/3;%400;%1000;%1;%10000000000000/3;
% for i=1:GridX
%     for j=1:GridY
%         for k=Z_Counter:GridZ
%             KXO(i,j,k) = AMOUNT * KXO(i,j,k); % X-Direction
%             KXW(i,j,k) = AMOUNT * KXW(i,j,k); % X-Direction
%             KYO(i,j,k) = AMOUNT * KYO(i,j,k); % Y-Direction
%             KYW(i,j,k) = AMOUNT * KYW(i,j,k); % Y-Direction
%             %KZO(i,j,k) = AMOUNT * KZO(i,j,k); % Z-Direction
%             %KZW(i,j,k) = AMOUNT * KZW(i,j,k); % Z-Direction
%         end
%     end
% end
% for i=1:GridX
%     for j=1:GridY
%         for k=Z_Counter1:GridZ
%             KZO(i,j,k) = AMOUNT2 * KZO(i,j,k); % Z-Direction
%             KZW(i,j,k) = AMOUNT2 * KZW(i,j,k); % Z-Direction
%         end
%     end
% end

%%%%NEW SYSTEM%%%

% %Z_Counter = floor(GridZ/2)+1; %round(GridZ/2);
% %  crossflow make 2 runs
% Z_Counter=1;
% Z_Counter1=1;
% AMOUNT = 3;
% AMOUNT2 = 1;%1E13/3;%400;%1000;%1;%10000000000000/3;
% for i=1:GridX
%     for j=1:GridY
%         for k=Z_Counter:GridZ
%             AMOUNTNEW = AMOUNT * k;
%             KXO(i,j,k) = AMOUNTNEW * KXO(i,j,k); % X-Direction
%             KXW(i,j,k) = AMOUNTNEW * KXW(i,j,k); % X-Direction
%             KYO(i,j,k) = AMOUNTNEW * KYO(i,j,k); % Y-Direction
%             KYW(i,j,k) = AMOUNTNEW * KYW(i,j,k); % Y-Direction
%             %KZO(i,j,k) = AMOUNT * KZO(i,j,k); % Z-Direction
%             %KZW(i,j,k) = AMOUNT * KZW(i,j,k); % Z-Direction
%         end
%     end
% end
% for i=1:GridX
%     for j=1:GridY
%         for k=Z_Counter1:GridZ
%             KZO(i,j,k) = AMOUNT2 * KZO(i,j,k); % Z-Direction
%             KZW(i,j,k) = AMOUNT2 * KZW(i,j,k); % Z-Direction
%         end
%     end
% end


Y4=ones(GridX+1+1+1,GridY+1+1+1,GridZ+1+1+1);
```

File name: transmissibilityxyz2.m

```
%calculation of transmissibility

GridX=size(DX,1);
GridY=size(DY,2);
GridZ=size(DZ,3);

%WU VISCOSITY NON NEWTONIAN
%Analytical formula
```

```
%Type=1;  WU viscosity
%Type=0;  WU viscosity without velocity effects
if Type == 0 || Type ==1
    if SW_counter > 1
        for i=1:GridX
            for j=1:GridY
                for k=1:GridZ
                    if SW(i,j,k)==S_wr
                        SW(i,j,k) = SW(i,j,k) + 0.00001;
                    end

ViscosityPOLYMER2(i,j,k) = ViscosityPOLYMER/12 * (9+3/N)^N * (150*KXO(i,j,k)*RKW(i,j,k)...
*SW(i,j,k)*Por(i,j,k)*(SW(i,j,k)-S_wr))^((1-N)/2);
                end
            end
        end
    else
        ViscosityPOLYMER2(1:GridX,1:GridY,1:GridZ) = ones(GridX,GridY,GridZ)*ViscosityPOLYMER;
    end
else
    %Type = 2; %2 means its not 0 = Wu which means it should be 1 = Benavides
    ViscosityPOLYMER2(1:GridX,1:GridY,1:GridZ) = ones(GridX,GridY,GridZ)*ViscosityPOLYMER;
end

ViscTermPOLYMER(1:GridX,1:GridY,1:GridZ) = 1 ./ ViscosityPOLYMER2(1:GridX,1:GridY,1:GridZ)*1000;

%RelativeKO ->RKO  ->Depends on saturation
%RelativeKW ->RKW  ->Depends on saturation

%Base case --> Assume for every grid point
k_rwo=0.1;
k_roo=1;
S_or =0.3;
S_wr =0.3;
nw   =2;
no   =2;

%North Slope Case --> Assume for every grid point
% k_rwo=0.1;
% k_roo=1;
% S_or =0.12;
% S_wr =0.12;
% nw   =4;
% no   =2.5;

%Alternative relative permeability 1--> Assume constant for every grid point
% k_rwo=1;
% k_roo=1;
% S_or =0.12;
% S_wr =0.12;
% nw   =2;
% no   =2;

%Alternative relative permeability 2--> Assume constant for every grid point
% k_rwo=1;
% k_roo=1;
% S_or =0.3;
% S_wr =0.3;
% nw   =2;
% no   =2;

%Alternative relative permeability 3--> Assume constant for every grid point
% k_rwo=0.16;
% k_roo=1;
% S_or =0.25;
% S_wr =0.12;
% nw   =1.5;
% no   =1.5;

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            S_S(i,j,k)=(SW(i,j,k)-S_wr) / (1- S_or - S_wr);
        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            %RKW(i,j,k) = k_rw = k_rwo * S_S;
            %RKO(i,j,k) = k_ro = k_roo * (1 - S_S);
            RKW(i,j,k) = k_rwo .* S_S(i,j,k).^nw;
            RKO(i,j,k) = k_roo .* (1 - S_S(i,j,k)).^no;
        end
    end
end

for j=1:(GridX-1)

    for k=1:(GridY)

        for l=1:(GridZ)


KAVGX(j+1,k,l) = (DX(j,k,l)+DX(j+1,k,l)) /( DX(j,k,l)/KXO(j,k,l)+DX(j+1,k,l)/KXO(j+1,k,l));

KAVGX(j+1,k,l) = (DX(j,k,l)+DX(j+1,k,l)) /( DX(j,k,l)/KXW(j,k,l)+DX(j+1,k,l)/KXW(j+1,k,l));

            if k==1
                KAVGX(1,k,l)   = 0; %KXO(1,k,l); % Grid open for the injection well
                KAVGX(1,k,l)   = 0; %KXW(1,k,l); % Grid open for the injection well
            else
                KAVGX(1,k,l)   = 0; % KXO(1,k,l) For open walls
                KAVGX(1,k,l)   = 0; % KXO(1,j,l) For open walls
            end

            if k==GridZ
                KAVGX(GridX+1,k,l)    = 0; %KXO(GridX,k,l); %Grid open for production well
                KAVGX(GridX+1,k,l)    = 0; %KXW(GridX,k,l); %Grid open for production well
            else
                KAVGX(GridX+1,k,l)    = 0; % KXO(1,j,l) For open walls
                KAVGX(GridX+1,k,l)    = 0; % KXO(1,j,l) For open walls
            end

        end
    end
end

for j=1:(GridX)

    for k=1:(GridY-1)

        for l=1:(GridZ)


KAVGY(j,k+1,l) = (DY(j,k,l)+DY(j,k+1,l)) /( DY(j,k,l)/KYO(j,k,l)+DY(j,k+1,l)/KYO(j,k+1,l));

KAVGY(j,k+1,l) = (DY(j,k,l)+DY(j,k+1,l)) /( DY(j,k,l)/KYW(j,k,l)+DY(j,k+1,l)/KYW(j,k+1,l));

            KAVGY(j,1,l)   = 0; % KYO(j,1,l) For open walls
            KAVGY(j,1,l)   = 0; % KYW(j,1,l) For open walls
            KAVGY(j,GridY+1,l)    = 0; % KYO(j,GridY,l) For open walls
            KAVGY(j,GridY+1,l)    = 0; % KYW(j,GridY,l) For open walls

        end
    end
end

for j=1:(GridX)

    for k=1:(GridY)

        for l=1:(GridZ-1)


KAVGZ(j,k,l+1) = (DZ(j,k,l)+DZ(j,k,l+1)) /( DZ(j,k,l)/KZO(j,k,l)+DZ(j,k,l+1)/KZO(j,k,l+1));

KAVGZ(j,k,l+1) = (DZ(j,k,l)+DZ(j,k,l+1)) /( DZ(j,k,l)/KZW(j,k,l)+DZ(j,k,l+1)/KZW(j,k,l+1));
            KAVGZ(j,k,1)   = 0; % KZO(j,k,1) For open walls
            KAVGZ(j,k,1)   = 0; % KZW(j,k,1) For open walls
            KAVGZ(j,k,GridZ+1)   = 0; % KZO(j,k,GridZ) For open walls
            KAVGZ(j,k,GridZ+1)   = 0; % KZW(j,k,GridZ) For open walls

        end
    end
end

DX2=DX; DY2=DY; DZ2=DZ;
RKOX=RKO; RKOY=RKO; RKOZ=RKO;
RKWX=RKW; RKWY=RKW; RKWZ=RKW;
for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            ViscTermOIL(GridX+1,j,k) = ViscTermOIL(GridX,j,k);
            ViscTermWATER(GridX+1,j,k) = ViscTermWATER(GridX,j,k);
            ViscTermPOLYMER(GridX+1,j,k) = ViscTermPOLYMER(GridX,j,k);
            ViscTermOIL(i,GridY+1,k) = ViscTermOIL(i,GridY,k);
            ViscTermWATER(i,GridY+1,k) = ViscTermWATER(i,GridY,k);
            ViscTermPOLYMER(i,GridY+1,k) = ViscTermPOLYMER(i,GridY,k);

            ViscTermOIL(i,j,GridZ+1) = ViscTermOIL(i,j,GridZ);
            ViscTermWATER(i,j,GridZ+1) = ViscTermWATER(i,j,GridZ);
            ViscTermPOLYMER(i,j,GridZ+1) = ViscTermPOLYMER(i,j,GridZ);

            KXO(GridX+1,j,k) = KXO(GridX,j,k);
            KXW(GridX+1,j,k) = KXW(GridX,j,k);

            KYO(i,GridY+1,k) = KYO(i,GridY,k);
            KYW(i,GridY+1,k) = KYW(i,GridY,k);

            KZO(i,j,GridZ+1) = KZO(i,j,GridZ);
            KZW(i,j,GridZ+1) = KZW(i,j,GridZ);

            %put it outside instead
            %DX2=DX; DY2=DY; DZ2=DZ;
            DX2(GridX+1,j,k) = DX2(GridX,j,k);
            DY2(i,GridY+1,k) = DY2(i,GridY,k);
            DZ2(i,j,GridZ+1) = DZ2(i,j,GridZ);

            RKOZ(i,j,GridZ+1)=RKOZ(i,j,GridZ);
            RKWZ(i,j,GridZ+1)=RKWZ(i,j,GridZ);

            RKOY(i,GridY+1,k)=RKOY(i,GridY,k);
            RKWY(i,GridY+1,k)=RKWY(i,GridY,k);

            RKOX(GridX+1,j,k)=RKOX(GridX,j,k);
            RKWX(GridX+1,j,k)=RKWX(GridX,j,k);

        end
    end
end

for i=1:(GridX+1)
    for j=1:GridY
        for k=1:GridZ

            MOX(i,j,k) = ViscTermOIL(i,j,k) * RKOX(i,j,k);
            MOX(GridX+2,j,k) = 0;
            MWX(i,j,k) = ViscTermWATER(i,j,k) * RKWX(i,j,k);
            MWX(GridX+2,j,k) = 0;
            MPX(i,j,k) = ViscTermPOLYMER(i,j,k) * RKWX(i,j,k);
            MPX(GridX+2,j,k) = 0;
        end
    end
end
for i=1:GridX
    for j=1:(GridY+1)
        for k=1:GridZ

            MOY(i,j,k) = ViscTermOIL(i,j,k) * RKOY(i,j,k);
            MOY(i,GridY+2,k) = 0;
            MWY(i,j,k) = ViscTermWATER(i,j,k) * RKWY(i,j,k);
            MWY(i,GridY+2,k) = 0;
            MPY(i,j,k) = ViscTermPOLYMER(i,j,k) * RKWY(i,j,k);
            MPY(i,GridY+2,k) = 0;
        end
    end
end
for i=1:GridX
    for j=1:GridY
        for k=1:(GridZ+1)

            MOZ(i,j,k) = ViscTermOIL(i,j,k) * RKOZ(i,j,k);
            MOZ(i,j,GridZ+2) = 0;
            MWZ(i,j,k) = ViscTermWATER(i,j,k) * RKWZ(i,j,k);
            MWZ(i,j,GridZ+2) = 0;
            MPZ(i,j,k) = ViscTermPOLYMER(i,j,k) * RKWZ(i,j,k);
            MPZ(i,j,GridZ+2) = 0;
        end
    end
end

%     % We will define Transmissibility in the X direction
%     % as TRXO for oil and TRXW for water
%     Constant(j) = 1 / ( ( DX(j)+DX(j+1) ) );
%     Constant2(j) = 1 / DX(j);
%     % Transmissibility for OIL
%     COIL(j) = Constant(j) * Constant2(j);
%     TrXO(j+1) = COIL(j) * KAVGX(j) * KXO(j) * MOXAVG(j);
%     % Transmissibility for WATER+POLYMER
%     CWATER(j) = Constant(j) * Constant2(j);
%     TrXW(j+1) = CWATER(j) * KAVGX(j) * KXW(j) * MWXAVG(j);
for i=1:(GridX+1)
    for j=1:GridY
        for k=1:GridZ
            % This increases the amount of DX to include boundaries
            DXTr(i+1,j,k) = DX2(i,j,k);
            DXTr(1,j,k) = DX2(1,j,k);
        end
    end
end

for i=1:GridX
    for j=1:(GridY+1)
        for k=1:GridZ
            % This increases the amount of DY to include boundaries
            DYTr(i,j+1,k) = DY2(i,j,k);
            DYTr(i,1,k) = DY2(i,1,k);
        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:(GridZ+1)
            % This increases the amount of DZ to include boundaries
            DZTr(i,j,k+1) = DZ2(i,j,k);
            DZTr(i,j,1) =DZ2(i,j,1);
        end
    end
end

for i=1:(GridX+1)
    for j=1:GridY
        for k=1:GridZ
            % Since now DX include boundaries, CX(i,j,k) for i,j,k=1
            % represent boundary values, by addint another Grid before and
            % after. The size of DXTr is of (GridX+2)

            % Left Side is defined
            CX1(i,j,k) = 1/ ((DXTr(i,j,k) + DXTr(i+1,j,k)));
            CX2(i,j,k) = 1/ DXTr(i,j,k);

            % For OIL
            CXO(i,j,k) = CX1(i,j,k) * CX2(i,j,k);
            % For WATER/POLYMER Newtonian
            CXW(i,j,k) = CX1(i,j,k) * CX2(i,j,k);

            if Y4(i+1,j+1,k+1) >= Y4(i,j+1,k+1)
                % Left side is defined, oil and water
                TrXO(i,j,k) = CXO(i,j,k) * KAVGX(i,j,k) * MOX(i+1,j,k) ;
                TrXW(i,j,k) = CXW(i,j,k) * KAVGX(i,j,k) * MWX(i+1,j,k) ;
                TrXP(i,j,k) = CXW(i,j,k) * KAVGX(i,j,k) * MPX(i+1,j,k) ;

% Left side is defined, non newtonian polymer
% Notice we use MWX(i+1+1,j,k) for both because of the
% restriction
if Y4(i+1,j+1,k+1) == Y4(i,j+1,k+1)
        TrXPN1(i,j,k) = 1;
else
if i<(GridX+1)

TrXPN1(i,j,k) = 1/(FactorN * abs( KXO(i+1,j,k)*MPX(i+1,j,k)...
*( ( Y4(i+1,j+1,k+1)-Y4(i,j+1,k+1) ) / ( 1/2*(DXTr(i+1,j,k)+DXTr(i,j,k) ) ) ))^(1-1/FactorN));

else
        TrXPN1(i,j,k) = TrXPN1(i-1,j,k);

end
end
            else
        % Left side is defined, oil and water
        TrXO(i,j,k) = CXO(i,j,k) * KAVGX(i,j,k) * MOX(i,j,k) ;
        TrXW(i,j,k) = CXW(i,j,k) * KAVGX(i,j,k) * MWX(i,j,k) ;
        TrXP(i,j,k) = CXW(i,j,k) * KAVGX(i,j,k) * MPX(i,j,k) ;

% Left side is defined, non newtonian polymer
% Notice we use MWX(i+1,j,k) for both because of the
% restriction
if i<(GridX+1)

TrXPN1(i,j,k) = 1/(FactorN * abs( KXO(i+1,j,k)*MPX(i,j,k)...
*( ( Y4(i+1,j+1,k+1)-Y4(i,j+1,k+1) ) / ( 1/2*(DXTr(i+1,j,k)+DXTr(i,j,k) ) ) ))^(1-1/FactorN));
```

```
else
        TrXPN1(i,j,k) = TrXPN1(i-1,j,k);

end
end

        end
    end
end
TrXPN1(1,:,:) = 0; TrXPN1(GridX+1,:,:) = 0; %TrXPN1(2,:,:);
TrXPN1(:,1,:) = TrXPN1(:,2,:);
TrXPN1(:,:,1) = TrXPN1(:,:,2);
for i=1:GridX
    for j=1:(GridY+1)
        for k=1:GridZ
            % Since now DY include boundaries, CY(i,j,k) for i,j,k=1
            % represent boundary values, by addint another Grid before and
            % after. The size of DYTr is of (GridX+2)
            CY1(i,j,k) = 1/ ((DYTr(i,j,k) + DYTr(i,j+1,k)));
            CY2(i,j,k) = 1/ DYTr(i,j,k);
            % For OIL
            CYO(i,j,k) = CY1(i,j,k) * CY2(i,j,k);
            % For WATER
            CYW(i,j,k) = CY1(i,j,k) * CY2(i,j,k);

            if Y4(i+1,j+1,k+1) >= Y4(i+1,j,k+1)
                TrYO(i,j,k) = CYO(i,j,k) * KAVGY(i,j,k) * MOY(i,j+1,k) ;
                TrYW(i,j,k) = CYW(i,j,k) * KAVGY(i,j,k) * MWY(i,j+1,k) ;
                TrYP(i,j,k) = CYW(i,j,k) * KAVGY(i,j,k) * MPY(i,j+1,k) ;

% Left side is defined, non newtonian polymer
% Notice we use MWY(i,j+1+1,k) for both because of the
% restriction
if Y4(i+1,j+1,k+1) == Y4(i+1,j,k+1)
        TrYPN1(i,j,k) = 1;
else
        if j<(GridY+1)

TrYPN1(i,j,k) = 1/(FactorN * abs( KYO(i,j+1,k)*MPY(i,j+1,k)...
*( ( Y4(i+1,j+1,k+1)-Y4(i+1,j,k+1) ) / ( 1/2*(DYTr(i,j+1,k)+DYTr(i,j,k) ) ) ))^(1-1/FactorN));

else
        TrYPN1(i,j,k) = TrYPN1(i,j-1,k);

end
end
            else
                TrYO(i,j,k) = CYO(i,j,k) * KAVGY(i,j,k) * MOY(i,j,k) ;
                TrYW(i,j,k) = CYW(i,j,k) * KAVGY(i,j,k) * MWY(i,j,k) ;
                TrYP(i,j,k) = CYW(i,j,k) * KAVGY(i,j,k) * MPY(i,j,k) ;

% Left side is defined, non newtonian polymer
% Notice we use MWY(i,j+1,k) for both because of the
% restriction
if j<(GridY+1)
```

```
TrYPN1(i,j,k) = 1/(FactorN * abs( KYO(i,j+1,k)*MPY(i,j,k)...
*( ( Y4(i+1,j+1,k+1)-Y4(i+1,j,k+1) ) / ( 1/2*(DYTr(i,j+1,k)+DYTr(i,j,k) ) ) ))^(1-1/FactorN));

else
TrYPN1(i,j,k) = TrYPN1(i,j-1,k);

end
end

        end
    end
end
TrYPN1(:,1,:) = 0; TrYPN1(:,GridY+1,:) = 0; %TrYPN1(:,2,:);
TrYPN1(1,:,:) = TrYPN1(2,:,:);
TrYPN1(:,:,1) = TrYPN1(:,:,2);
for i=1:GridX
    for j=1:GridY
        for k=1:(GridZ+1)
            % Since now DZ include boundaries, CZ(i,j,k) for i,j,k=1
            % represent boundary values, by addint another Grid before and
            % after. The size of DZTr is of (GridX+2)
            CZ1(i,j,k) = 1/ ((DZTr(i,j,k) + DZTr(i,j,k+1)));
            CZ2(i,j,k) = 1/ DZTr(i,j,k);

            % For OIL
            CZO(i,j,k) = CZ1(i,j,k) * CZ2(i,j,k);
            % Fot WATER
            CZW(i,j,k) = CZ1(i,j,k) * CZ2(i,j,k);

            if Y4(i+1,j+1,k+1) >= Y4(i+1,j+1,k)
                TrZO(i,j,k) = CZO(i,j,k) * KAVGZ(i,j,k) * MOZ(i,j,k+1) ;
                TrZW(i,j,k) = CZW(i,j,k) * KAVGZ(i,j,k) * MWZ(i,j,k+1) ;
                TrZP(i,j,k) = CZW(i,j,k) * KAVGZ(i,j,k) * MPZ(i,j,k+1) ;

% Left side is defined, non newtonian polymer
% Notice we use MWZ(i,j,k+1+1) for both because of the
% restriction
if Y4(i+1,j+1,k+1) == Y4(i+1,j+1,k)
        TrZPN1(i,j,k) = 1;
else
        if k<(GridZ+1)

TrZPN1(i,j,k) = 1/(FactorN * abs( KZO(i,j,k+1)*MPZ(i,j,k+1)...
*( ( Y4(i+1,j+1,k+1)-Y4(i+1,j+1,k) ) / ( 1/2*(DZTr(i,j,k+1)+DZTr(i,j,k) ) ) ))^(1-1/FactorN));
else
TrZPN1(i,j,k) = TrZPN1(i,j,k-1);

end
end
            else
                TrZO(i,j,k) = CZO(i,j,k) * KAVGZ(i,j,k) * MOZ(i,j,k) ;
                TrZW(i,j,k) = CZW(i,j,k) * KAVGZ(i,j,k) * MWZ(i,j,k) ;
                TrZP(i,j,k) = CZW(i,j,k) * KAVGZ(i,j,k) * MPZ(i,j,k) ;

% Left side is defined, non newtonian polymer
% Notice we use MWZ(i,j,k+1) for both because of the
% restriction
if k<(GridZ+1)

TrZPN1(i,j,k) = 1/(FactorN * abs( KZO(i,j,k+1)*MPZ(i,j,k)...
*( ( Y4(i+1,j+1,k+1)-Y4(i+1,j+1,k) ) / ( 1/2*(DZTr(i,j,k+1)+DZTr(i,j,k) ) ) ))^(1-1/FactorN));

else
TrZPN1(i,j,k) = TrZPN1(i,j,k-1);

end
end

        end
    end
end
TrZPN1(:,:,1) = 0; TrZPN1(:,:,GridZ+1) = 0; %TrZPN1(:,:,2);
TrZPN1(:,1,:) = TrZPN1(:,2,:);
TrZPN1(1,:,:) = TrZPN1(2,:,:);
```

File name: continuityxyz.m

```
% Account for different type of viscosities behaviors
% Define injection and production rates
% Define particular transmissibilities

% Account for different type of viscosities behaviors
if Type == 1
        TrXW = TrXPN1.*TrXP; MWX = MPX;
        TrYW = TrYPN1.*TrYP; MWY = MPY;
        TrZW = TrZPN1.*TrZP; MWZ = MPZ;
elseif Type == 0
        TrXW = TrXP; MWX = MPX;
        TrYW = TrYP; MWY = MPY;
        TrZW = TrZP; MWZ = MPZ;
else
        lalala=1;
end

MOX2(:,:,:)=MOX(1:GridX+1,:,:);   MWX2(:,:,:)=MWX(1:GridX+1,:,:); ...
MPX2(:,:,:)=MPX(1:GridX+1,:,:);
MOY2(:,:,:)=MOY(:,1:GridY+1,:);   MWY2(:,:,:)=MWY(:,1:GridY+1,:); ...
MPY2(:,:,:)=MPY(:,1:GridY+1,:);
MOZ2(:,:,:)=MOZ(:,:,1:GridZ+1);   MWZ2(:,:,:)=MWZ(:,:,1:GridZ+1); ...
MPZ2(:,:,:)=MPZ(:,:,1:GridZ+1);

% Define injection and production rates
for i=1:GridX
    for j=1:GridY
        for k=1:GridZ

            % Give values for injection rate at every grid point
            % The injection are done in the X-direction (DX)
            % We change it to m^3/s volume rate.
            % For pore rate /mobility
            if i==1 & j==1 & k==1
                qO(i,j,k) = qOIL*DY(1,1,1)*DZ(1,1,1);%/(KXO(i,j,k)/MO(i,j,k));
                qW(i,j,k) = qWATER*DY(1,1,1)*DZ(1,1,1);%/(KXW(i,j,k)/MW(i,j,k));
                qP(i,j,k) = qPOLYMER*DY(1,1,1)*DZ(1,1,1);%/(KXW(i,j,k)/MP(i,j,k));
            else
                qO(i,j,k) = 0;
                qW(i,j,k) = 0;
                qP(i,j,k) = 0;
            end
        end
    end
end

TrXP2(:,:,:) = TrXP(:,:,:) .* TrXPN1(:,:,:);
TrYP2(:,:,:) = TrYP(:,:,:) .* TrYPN1(:,:,:);
TrZP2(:,:,:) = TrZP(:,:,:) .* TrZPN1(:,:,:);

%EDIT of TRANSMISSIBILITIES TO SIMULATE A FAULT

% TrXO(5,1,1)=1;  TrXO(6,1,1)=1;
% TrXW(5,1,1)=1;  TrXW(6,1,1)=1;
% TrXP(5,1,1)=1;  TrXP(6,1,1)=1;
%
% TrYO(5,1,1)=1;  TrYO(5,2,1)=1;
% TrYW(5,1,1)=1;  TrYW(5,2,1)=1;
% TrYP(5,1,1)=1;  TrYP(5,2,1)=1;
%
% TrZO(5,1,2)=1;
GridXzone1 = LengthzoneX(1)/DX(1);
edgezone1 = GridXzone1 + 1;
zone2 = edgezone1;
GridXzone2 = LengthzoneX(2)/DX(edgezone1);
edgezone2 = edgezone1 + GridXzone2;
zone3 = edgezone2;
GridXzone3 = LengthzoneX(3)/DX(edgezone2);
edgezone3 = edgezone2 + GridXzone3;

% %Permeability Profile 2
% for i=1:(GridX-4)
%         for k=1:GridZ
%             TrXO(i+2,i+2,k)=1E-50;%0;
%             TrXW(i+2,i+2,k)=1E-50;%0;
%             TrXP(i+2,i+2,k)=1E-50;%0;
%         end
% end
% for i=1:(GridX-4)
%         for k=1:GridZ
%             TrYO(i+2,i+3,k)=1E-50;%0;
%             TrYW(i+2,i+3,k)=1E-50;%0;
%             TrYP(i+2,i+3,k)=1E-50;%0;
%         end
% end

% %Boundary Profile 3
% TrXO(7:10,2:4,:)=1E-50;%for some reason they cannot be zero! -->unstable!
% TrXW(7:10,2:4,:)=1E-50;
% TrXP(7:10,2:4,:)=1E-50;
% TrYO(7:10,2:4,:)=1E-50;
% TrYW(7:10,2:4,:)=1E-50;
% TrYP(7:10,2:4,:)=1E-50;
% TrZO(7:10,2:4,:)=1E-50;%0;%1E-50;
% TrZW(7:10,2:4,:)=1E-50;%0;%1E-50;
% TrZP(7:10,2:4,:)=1E-50;%0;%1E-50;
% TrXO(5:7,12:14,:)=1E-50;%for some reason they cannot be zero! -->unstable!
% TrXW(5:7,12:14,:)=1E-50;
% TrXP(5:7,12:14,:)=1E-50;
% TrYO(5:7,12:14,:)=1E-50;
% TrYW(5:7,12:14,:)=1E-50;
% TrYP(5:7,12:14,:)=1E-50;
% TrZO(5:7,12:14,:)=1E-50;%0;%1E-50;
% TrZW(5:7,12:14,:)=1E-50;%0;%1E-50;
% TrZP(5:7,12:14,:)=1E-50;%0;%1E-50;


% Edit of injection and production rates
for k=1:GridZ
    qW(1,1,k)=qW(1,1,1);
end

%%%qO(GridX,1,1)=-qW(1,1,1)/5;
%qO(GridX,GridY,GridZ)=-qW(1,1,1);
qO(GridX,GridY,GridZ)=-qW(1,1,1);

for k=1:GridZ
    qO(GridX,GridY,k)=qO(GridX,GridY,GridZ);
end
```

File name: equation.m

```
for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            %Pre(k+(j-1)*GridZ+(i-1)*GridZ*GridY)

            a_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrXO(i,j,k);
            b_o1(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =-(TrXO(i,j,k) ...
                + TrYO(i,j,k) + TrZO(i,j,k));
            b_o2(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =-(TrXO(i+1,j,k) ...
                + TrYO(i,j+1,k) + TrZO(i,j,k+1));
            c_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrXO(i+1,j,k);
            d_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  qO(i,j,k);
            e_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrYO(i,j,k);
            f_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrYO(i,j+1,k);
            g_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrZO(i,j,k);
            h_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrZO(i,j,k+1);

            a_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrXW(i,j,k);
            b_w1(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = -(TrXW(i,j,k) ...
                + TrYW(i,j,k) + TrZW(i,j,k));
            b_w2(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = -(TrXW(i+1,j,k) ...
                + TrYW(i,j+1,k) + TrZW(i,j,k+1));
            c_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  +TrXW(i+1,j,k);
            d_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  qW(i,j,k);
            e_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrYW(i,j,k);
            f_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrYW(i,j+1,k);
            g_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrZW(i,j,k);
            h_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY) =  TrZW(i,j,k+1);

        end
    end
end


for i=1:GridX
    for j=1:GridY
        for k=1:GridZ

aa(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = ...
    a_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY)  + a_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
bb1(k+(j-1)*GridZ+(i-1)*GridZ*GridY)= ...
    b_o1(k+(j-1)*GridZ+(i-1)*GridZ*GridY) + b_w1(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
bb2(k+(j-1)*GridZ+(i-1)*GridZ*GridY)= ...
    b_o2(k+(j-1)*GridZ+(i-1)*GridZ*GridY) + b_w2(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
cc(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = ...
    c_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY)  + c_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
dd(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = ...
    d_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY)  + d_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
ee(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = ...
    e_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY)  + e_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
ff(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = ...
    f_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY)  + f_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
gg(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = ...
    g_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY)  + g_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
hh(k+(j-1)*GridZ+(i-1)*GridZ*GridY) = ...
    h_o(k+(j-1)*GridZ+(i-1)*GridZ*GridY)  + h_w(k+(j-1)*GridZ+(i-1)*GridZ*GridY);

        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ

            bb(k+(j-1)*GridZ+(i-1)*GridZ*GridY)= bb1(k+(j-1)*GridZ+(i-1)*GridZ*GridY) ...
                + bb2(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            if i==1 & j==1 & k==1
                bb(1)=bb(1)-cc(1)-ff(1)-hh(1);
            end
            AA(k+(j-1)*GridZ+(i-1)*GridZ*GridY,k+(j-1)*GridZ+(i-1)*GridZ*GridY)...
                =bb(k+(j-1)*GridZ+(i-1)*GridZ*GridY);

            if k>1
```

```
                AA(k+(j-1)*GridZ+(i-1)*GridZ*GridY,k-1+(j-1)*GridZ+(i-1)*GridZ*GridY) ...
                    =  gg(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            end

            if j>1
                AA(k+(j-1)*GridZ+(i-1)*GridZ*GridY,k+(j-1-1)*GridZ+(i-1)*GridZ*GridY) ...
                    =  ee(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            end

            if i>1
                AA(k+(j-1)*GridZ+(i-1)*GridZ*GridY,k+(j-1)*GridZ+(i-1-1)*GridZ*GridY) ...
                    =  aa(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            end

            if k<GridZ
                AA(k+(j-1)*GridZ+(i-1)*GridZ*GridY,k+1+(j-1)*GridZ+(i-1)*GridZ*GridY) ...
                    =  hh(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            end

            if j<GridY
                AA(k+(j-1)*GridZ+(i-1)*GridZ*GridY,k+(j+1-1)*GridZ+(i-1)*GridZ*GridY) ...
                    =  ff(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            end

            if i<GridX
                AA(k+(j-1)*GridZ+(i-1)*GridZ*GridY,k+(j-1)*GridZ+(i+1-1)*GridZ*GridY) ...
                    =  cc(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            end

        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            dd(k+(j-1)*GridZ+(i-1)*GridZ*GridY)=dd(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
        end
    end
end
```

File name: velocity.m

```
V.x = zeros(GridX+1,GridY,GridZ);
V.y = zeros(GridX,GridY+1,GridZ);
V.z = zeros(GridX,GridY,GridZ+1);
Nx=GridX; Ny=GridY; Nz=GridZ;
if Type == 0
    TrXPN1(1:GridX+1,1:GridY,1:GridZ)=1;
    TrYPN1(1:GridX,1:GridY+1,1:GridZ)=1;
    TrZPN1(1:GridX,1:GridY,1:GridZ+1)=1;
end
```

```
V.x(2:Nx ,:,:) = (Y45 ( 1:Nx-1,:,:)-Y45(2:Nx,:,:))...
.*KXO(2:Nx,:,:).*CXO(2:Nx,:,:).*(MOX2(2:Nx,1:Ny,1:Nz)+MWX2(2:Nx,1:Ny,1:Nz).*TrXPN1(2:Nx,:,:));
V.y(:,2: Ny,:) = (Y45 (:,1: Ny-1,:)-Y45(:,2:Ny,:))...
.*KYO(:,2:Ny,:).*CYO(:,2:Ny,:).*(MOY2(1:Nx,2:Ny,1:Nz)+MWY2(1:Nx,2:Ny,1:Nz).*TrYPN1(:,2:Ny,:));
V.z(:,:,2: Nz) = (Y45 (:,:,1: Nz-1)-Y45(:,:,2:Nz))...
.*KZO(:,:,2:Nz).*CZO(:,:,2:Nz).*(MOZ2(1:Nx,1:Ny,2:Nz)+MWZ2(1:Nx,1:Ny,2:Nz).*TrZPN1(:,:,2:Nz));

% Calculated just to check
a1=V.x(2,1,1);
a2=V.y(1,2,1);
a3=V.z(1,1,2);


%VelocityTotal
V.x(GridX+1,:,:)=0;
V.y(:,GridY+1,:)=0;
V.z(:,:,GridZ+1)=0;

%MobilityOil(:,:,:)   = RKO(:,:,:) ./ ViscosityOIL;
MobilityOilX(:,:,:)   = MOX2(:,:,:);
MobilityOilY(:,:,:)   = MOY2(:,:,:);
MobilityOilZ(:,:,:)   = MOZ2(:,:,:);
%MobilityWater(:,:,:) = RKW(:,:,:) ./ ViscosityWATER;
MobilityWaterX(:,:,:) = MWX2(:,:,:);
MobilityWaterY(:,:,:) = MWY2(:,:,:);
MobilityWaterZ(:,:,:) = MWZ2(:,:,:);

%Frac_o(:,:,:) = MobilityOil(:,:,:)   ./ (MobilityOil(:,:,:) + MobilityWater(:,:,:));

%Frac_w(:,:,:) = MobilityWater(:,:,:)   ./ (MobilityOil(:,:,:) + MobilityWater(:,:,:));
XFrac_w(:,:,:) = MobilityWaterX(:,:,:)   ./ (MobilityOilX(:,:,:) + MobilityWaterX(:,:,:));
YFrac_w(:,:,:) = MobilityWaterY(:,:,:)   ./ (MobilityOilY(:,:,:) + MobilityWaterY(:,:,:));
ZFrac_w(:,:,:) = MobilityWaterZ(:,:,:)   ./ (MobilityOilZ(:,:,:) + MobilityWaterZ(:,:,:));

Frac_wX=zeros(GridX+2,GridY,GridZ);
Frac_wY=zeros(GridX,GridY+2,GridZ);
Frac_wZ=zeros(GridX,GridY,GridZ+2);

Frac_wX(2:GridX+1,:,:)=XFrac_w(1:GridX,:,:);
Frac_wY(:,2:GridY+1,:)=YFrac_w(:,1:GridY,:);
Frac_wZ(:,:,2:GridZ+1)=ZFrac_w(:,:,1:GridZ);

% To prevent errors
for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            if Frac_wX(i,j,k) == 1
                Frac_wX(i,j,k) = 1-1E-50;
            end
            if Frac_wY(i,j,k) == 1
                Frac_wY(i,j,k) = 1-1E-50;
            end
            if Frac_wZ(i,j,k) == 1
                Frac_wZ(i,j,k) = 1-1E-50;
            end
        end
    end
end

V.x(GridX+2,:,:)=0;
V.y(:,GridY+2,:)=0;
V.z(:,:,GridZ+2)=0;

%Accounts for the progress of fractional flow on every grid point as time
%goes by.
for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            ProgressFrac_wX(k+(j-1)*GridZ+(i-1)*GridZ*GridY,SW_counter) = XFrac_w(i,j,k);
            ProgressFrac_wY(k+(j-1)*GridZ+(i-1)*GridZ*GridY,SW_counter) = YFrac_w(i,j,k);
            ProgressFrac_wZ(k+(j-1)*GridZ+(i-1)*GridZ*GridY,SW_counter) = ZFrac_w(i,j,k);
            ProgressSW(k+(j-1)*GridZ+(i-1)*GridZ*GridY,SW_counter) = SW(i,j,k);
        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            if SW(i,j,k) > 1-S_wr;%-1E-3;
                MovableOIL(i,j,k)   = 1-SW(i,j,k)-S_wr-1E-3;
            else
                MovableOIL(i,j,k)   = 0;
            end
            if SW(i,j,k) > S_wr;%-1E-3
                MovableWATER(i,j,k) = SW(i,j,k)-S_wr-1E-3;
            else
                MovableWATER(i,j,k) = 0;
            end
        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ

%USUAL FractionX
if V.x(i,j,k) + V.x(i+1,j,k) >= 0
        FractionX(i,j,k) = V.x(i,j,k) * Frac_wX(i,j,k) - V.x(i+1,j,k) * Frac_wX(i+1,j,k);
else
        FractionX(i,j,k) = V.x(i,j,k) * Frac_wX(i+1,j,k) - V.x(i+1,j,k) * Frac_wX(i+1+1,j,k);
end
%ALTERNATIVE ROSSEN
%FractionX(i,j,k) = (V.x(i,j,k)+V.x(i+1,j,k))/2 * (Frac_wX(i,j,k) ...
- Frac_wX(i+1,j,k)) -Frac_wX(i,j,k)*qW(i,j,k);

%USUAL FractionY
if V.y(i,j,k) + V.y(i,j+1,k) >= 0
        FractionY(i,j,k) = V.y(i,j,k) * Frac_wY(i,j,k) - V.y(i,j+1,k) * Frac_wY(i,j+1,k);
else
        FractionY(i,j,k) = V.y(i,j,k) * Frac_wY(i,j+1,k) - V.y(i,j+1,k) * Frac_wY(i,j+1+1,k);
```

```
end
%ALTERNATIVE ROSSEN
%FractionY(i,j,k) = (V.y(i,j,k)+V.y(i,j+1,k))/2 * (Frac_wY(i,j,k) ...
- Frac_wY(i,j+1,k)) -Frac_wY(i,j,k)*qW(i,j,k);

%USUAL FractionZ
if V.z(i,j,k) + V.z(i,j,k+1) >= 0
        FractionZ(i,j,k) = V.z(i,j,k) * Frac_wZ(i,j,k) - V.z(i,j,k+1) * Frac_wZ(i,j,k+1);
else
        FractionZ(i,j,k) = V.z(i,j,k) * Frac_wZ(i,j,k+1) - V.z(i,j,k+1) * Frac_wZ(i,j,k+1+1);
end
%ALTERNATIVE ROSSEN
%FractionZ(i,j,k) = (V.z(i,j,k)-V.z(i,j,k+1))/2 * (Frac_wZ(i,j,k) ...
- Frac_wZ(i,j,k+1)) -Frac_wZ(i,j,k)*qW(i,j,k);

        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ

Waterdisplaced(i,j,k)  = ( FractionX(i,j,k) + FractionY(i,j,k) + FractionZ(i,j,k)  )...
*1/(DX(i,j,k)*DY(i,j,k)*DZ(i,j,k))*DeltaT/Por(i,j,k);
Waterdisplaced2(i,j,k) = (max(q0(i,j,k),0) +  (Frac_wX(i,j,k)+Frac_wY(i,j,k)+Frac_wZ(i,j,k))...
* min(q0(i,j,k),0)  ) *1/(DX(i,j,k)*DY(i,j,k)*DZ(i,j,k))*DeltaT/Por(i,j,k);

if i==GridX & j==GridY %& k==GridZ
        if Waterdisplaced2(i,j,k) < 0 & abs(Waterdisplaced2(i,j,k)) > Waterdisplaced(i,j,k)
                if abs(Waterdisplaced(i,j,k)+Waterdisplaced2(i,j,k)) > MovableWATER(i,j,k)

                Waterdisplaced2(i,j,k) = -MovableWATER(i,j,k) - Waterdisplaced(i,j,k);
                SumWaterDisplaced(i,j,k) = Waterdisplaced(i,j,k) + Waterdisplaced2(i,j,k);
        else
                SumWaterDisplaced(i,j,k) = Waterdisplaced(i,j,k) + Waterdisplaced2(i,j,k);
        end
        elseif i==1 & j==1 %& k==1
                nothing=1;
        else
                umWaterDisplaced(i,j,k) = Waterdisplaced(i,j,k) + Waterdisplaced2(i,j,k);
        end

else
        if Waterdisplaced(i,j,k) <0 & abs(Waterdisplaced(i,j,k)) > MovableWATER(i,j,k)
                if abs(Waterdisplaced(i,j,k)+Waterdisplaced2(i,j,k)) > MovableWATER(i,j,k)
                  Waterdisplaced2(i,j,k) = -MovableWATER(i,j,k) - Waterdisplaced(i,j,k);
                  SumWaterDisplaced(i,j,k) = Waterdisplaced(i,j,k) + Waterdisplaced2(i,j,k);
                else
                  SumWaterDisplaced(i,j,k) = Waterdisplaced(i,j,k) + Waterdisplaced2(i,j,k);
                end
        end
           SumWaterDisplaced(i,j,k) = Waterdisplaced(i,j,k) + Waterdisplaced2(i,j,k);
        end
        Injection(i,j,k) = qW(i,j,k)*1/(DX(i,j,k)*DY(i,j,k)*DZ(i,j,k))*DeltaT/Por(i,j,k);
        end
    end
end

for i=1:GridX
    for j=1:GridY
        for k=1:GridZ

            SW_T(i,j,k) = SW(i,j,k) + SumWaterDisplaced(i,j,k) + Injection(i,j,k);

            if SW_T(i,j,k) > (1-S_or)
                randomnumber = randi([10 50]);
                SW_T(i,j,k) = 1-S_or -  randomnumber*1E-5;
            elseif SW_T(i,j,k) < S_wr
                SW_T(i,j,k) = S_wr;
            end

        end
    end
end

% Calculation of flow rate
QW_In(SW_counter) = sum(sum(sum(qW(1,1,:)*1/(DX(1,1,1)*DY(1,1,1)*DZ(1,1,1))...
*DeltaT/Por(1,1,1))));
QO_In(SW_counter) = sum(sum(sum(qO(1,1,:)*1/(DX(1,1,1)*DY(1,1,1)*DZ(1,1,1))...
*DeltaT/Por(1,1,1))));
QW_Out(SW_counter) = abs(sum(sum(sum(Waterdisplaced2(GridX,GridY,:)))));
QO_Out(SW_counter) = abs(max(sum(sum(sum(abs(qO(GridX,GridY,:)*1/(DX(1,1,1)...
*DY(1,1,1)*DZ(1,1,1))*DeltaT/Por(1,1,1))))) - QW_Out(SW_counter),0));

% Calculation of volumetric flow rate
VolQW_In(SW_counter) = QW_In(SW_counter);%m^3/D
VolQO_In(SW_counter) = QO_In(SW_counter);%m^3/D
VolQW_Out(SW_counter) = QW_Out(SW_counter);%m^3/D
VolQO_Out(SW_counter) = QO_Out(SW_counter);%m^3/D
SUMVolQO_Out(SW_counter+1)= SUMVolQO_Out(SW_counter)+VolQO_Out(SW_counter);
SUMVolQW_Out(SW_counter+1)= SUMVolQW_Out(SW_counter)+VolQW_Out(SW_counter);
SUMVolQO_In(SW_counter+1)= SUMVolQO_In(SW_counter)+VolQO_In(SW_counter);
SUMVolQW_In(SW_counter+1)= SUMVolQW_In(SW_counter)+VolQW_In(SW_counter);
WATERCUT(SW_counter+1) = SUMVolQW_Out(SW_counter+1)...
/(SUMVolQW_Out(SW_counter+1)+SUMVolQO_Out(SW_counter+1));



SW_T2=SW_T-SW;

% To check everything is going alright
VTotal(1,1,1) = V.x(2,1,1)+V.y(1,2,1)+V.z(1,1,2);
% Added a counter for every update (a plot is updated too)
[VTotal(1,1,1),SW_counter+1]
qW(1,1,1)


%%%%%%%%%% PLOTS %%%%%%%%%
```

```
set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

  what1=1;
% if SW_counter == 1
figure(10+what1);
surf(SW_T(:,:,1))
hold on
contour3(SW_T(:,:,1))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off
% end


if STOP==1
    set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(11+what1);
surf(SW_T(:,:,2))
hold on
contour3(SW_T(:,:,2))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off

if GridZ>2
    set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(12+what1);
surf(SW_T(:,:,3))
hold on
contour3(SW_T(:,:,3))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off
```

```
if GridZ>3
    set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(13+what1);
surf(SW_T(:,:,4))
hold on
contour3(SW_T(:,:,4))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off

if GridZ>4
    set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(14+what1);
surf(SW_T(:,:,5))
hold on
contour3(SW_T(:,:,5))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off
end
end
end
end

if SW_counter==49 || SW_counter==149

set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

% if SW_counter == 1
figure(SW_counter+1);
surf(SW_T(:,:,1))
hold on
contour3(SW_T(:,:,1))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
```

```
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off

set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(SW_counter+2);
surf(SW_T(:,:,2))
hold on
contour3(SW_T(:,:,2))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off

if GridZ>2
    set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(SW_counter+3);
surf(SW_T(:,:,3))
hold on
contour3(SW_T(:,:,3))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off

if GridZ>3
    set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(SW_counter+4);
surf(SW_T(:,:,4))
hold on
contour3(SW_T(:,:,4))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
```

```
hold off

if GridZ>4
    set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(SW_counter+5);
surf(SW_T(:,:,5))
hold on
contour3(SW_T(:,:,5))
xlabel('x'),ylabel('y'),zlabel('Water Saturation')
title('Saturation Distribution')
rotate3d on
zlim([0.3 0.7])
%zlim([0.1 0.9])
%zlim([0 1])
colorbar
hold off

end
end
end
end
```

File name: loop2.m

```
%loop 2
%SW=0;
GridZ=5;
SW(20,20,1:GridZ)=0; SW(19,20,1:GridZ)=0; SW(20,19,1:GridZ)=0;
SW_counter=1; %--_>write manually
STOP=0;
SUMVolQO_Out(1)=0;
SUMVolQW_Out(1)=0;
SUMVolQO_In(1)=0;
SUMVolQW_In(1)=0;



while STOP==0   %%For water breakthrough
    for k=1:GridZ
        if SW(19,20,k)<0.3011 & SW(20,19,k)<0.3011
        STOP1(k)=0;
        else
        STOP1(k)=1;
        end
    end
    if any(STOP1) == 1
        STOP=1;
    end
%while SW_counter<500     %For desired t=SW_counter,
%     if SW_counter ==499 %To make plots in all the grid Z
%         STOP=1;         %
%     end                 %
%transmissibilityxyz
if SW_counter==1
    testingxyz
    transmissibilityxyz1
end

transmissibilityxyz2

continuityxyz

%continuitytxyz

%calculatingxyz

equation

%equation2

%Y2=linsolve(F,BB);

%Y2=AA/dd;
%Y2=linsolve(AA,transpose(dd));
%Y2=mldivide(AA,transpose(dd));
% opts.RECT = true;
% Y2=linsolve(AA,transpose(dd),opts);
Y2=AA\transpose(-dd);
for i=1:GridX
    for j=1:GridY
        for k=1:GridZ
            Y3(i,j,k)=Y2(k+(j-1)*GridZ+(i-1)*GridZ*GridY);
            % Y4 gives pressure in [bar] instead of N/m^2
            Y4(i+1,j+1,k+1)=Y3(i,j,k);%/100000;
            Y45(i,j,k)=Y3(i,j,k);
        end
    end
end

asd1=Y2(1)*bb(1)+Y2(1+1)*hh(1)+Y2(1+GridZ)*ff(1)+Y2(1+GridY*GridZ)*cc(1)
asd2=-qW(1,1,1)%/(DX(1,1,1)*DY(1,1,1)*DZ(1,1,1))

velocity
SW_counter=SW_counter+1;
SW=SW_T;

end

set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(1);
surf(Y45(:,:,1));
hold on
contour3(Y45(:,:,1));
```

```
xlabel('x'),ylabel('y'),zlabel('Pressure differential')
title('Pressure Profile, Z=1')
rotate3d on
colorbar
hold off

set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(2);
surf(Y45(:,:,2));
surf(Y45(:,:,2));
hold on
contour3(Y45(:,:,2));
contour3(Y45(:,:,2));
xlabel('x'),ylabel('y'),zlabel('Pressure differential')
title('Pressure Profile, Z=2')
rotate3d on
colorbar
hold off

if GridZ>2
set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(3);
surf(Y45(:,:,3));
hold on
contour3(Y45(:,:,3));
xlabel('x'),ylabel('y'),zlabel('Pressure differential')
title('Pressure Profile, Z=3')
rotate3d on
colorbar
hold off

if GridZ>3
set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')

figure(4);
surf(Y45(:,:,4));
hold on
contour3(Y45(:,:,4));
xlabel('x'),ylabel('y'),zlabel('Pressure differential')
title('Pressure Profile, Z=4')
rotate3d on
colorbar
hold off

if GridZ>4
set(gca,'FontSize',30,'fontWeight','bold')

set(findall(gcf,'type','text'),'FontSize',30,'fontWeight','bold')
```

```
figure(5);
surf(Y45(:,:,5));
hold on
contour3(Y45(:,:,5));
xlabel('x'),ylabel('y'),zlabel('Pressure differential')
title('Pressure Profile, Z=5')
rotate3d on
colorbar
hold off
end
end
end
```

C Files Eclipse

File name: POLYMER-2-2.DATA

```
RUNSPEC
TITLE
TEST 25 BY 25 BY 10 OIL/WATER/POLYMER SYSTEM

DIMENS
   25   25    10  /

OIL

WATER

POLYMER

BRINE

METRIC

TABDIMS
    1    1   50     2    2   20 /

REGDIMS
    2    1    0    0 /

WELLDIMS
    2    10    1    2 /

START
   1 ’JAN’ 1983  /

NSTACK
    8 /

UNIFOUT

UNIFIN

GRID      ==============================================================

EQUALS
      ’DX’    20      /
      ’DY’    20      /
      ’PORO’  0.3       /

      ’DZ’    10        1  25  1  25  1  4  /
      ’PERMX’  300 /
      ’PERMY’  300 /
      ’PERMZ’  30 /
      ’MULTZ’  1.0      /
      ’TOPS’   4000/
      'DZ'    5        1  25  1  25  5  8  /
      'PERMX'  150 /
      'PERMY'  150 /
      'PERMZ'  30 /
      'MULTZ'  1.0      /


      'DZ'    20        1  25  1  25  9  10  /
      'PERMX'  300 /
      'PERMY'  300 /
      'PERMZ'  30 /


/
RPTGRID
-- Report Levels for Grid Section Data
--
'PORO'
'PORV'
 /

PROPS     ==============================================================

SWFN
    .20    .0    4.0
    .7    0.7    2.0
   1.0    1.0    0.0
/

SOF2
    .3000  .0000
    .8000 1.0000
/

PVTW
  .0  1.0  3.03E-06  0.006  0.0 /

PVDO
  .0     1.0     6.0
 8000.0   .92    6.0
/

ROCK
 4000.0    .30E-05 /

DENSITY
 52.0000  64.0000   .04400 /

SALTNODE
  0.0
  0.85
/

PLYVISCS
```

```
  0.0  1.0
       1.0/
  5.0  100.0
       25.0/
/

PLYROCK
   0.1   1.5  1000.0  2  0.005 /

ADSALNOD
  0.0
  0.85
/

PLYADSS
  0.0  0.000
       0.000/
  2.0  0.0015
       0.0150/
  8.0  0.0025
       0.0250/
 /

PLMIXPAR
 1.0 /

PLYMAX
 5.0  0.85 /

RPTPROPS
 -- PROPS Reporting Options
 --
 'PLYVISCS'
/

SOLUTION   ============================================================

PRESSURE
  6250*4000.0 /

SWAT
  6250*0.301 /

SALT
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
 /
RPTSOL
’PRES’ ’SOIL’ ’SWAT’ ’RESTART=2’ ’PBLK’
’SALT’ ’PLYADS’ ’FIPPLY=2’ /


SUMMARY    =============================================================

RUNSUM

FWPR
FOPR
FOPT
FWCT
FCPR
FCPT
FCIR
FCIT
FCIP
FCAD
GCPR
 ’G’ /
GCPT
 ’G’ /
GCIR
 ’G’ /
GCIT
 ’G’ /
```

```
WCPR
 ’P’ /
WCPT
 ’P’ /
WCIR
 ’I’ /
WCIT
 ’I’ /
CCFR
 ’P’ 25 25  1 /
 /
CCPT
 ’P’ 25 25  1 /
 /
CCIT
 ’I’  1  1  1 /
 /
RCIP
 1 2 /
RCFT
 1 2 /
 /
RCAD
 1 2 /
BCCN
 1 1 1 /
 /
BCIP
 1 1 1 /
 /
BCAD
 1 1 1 /
 /
FSPR
FSPT
FSIR
FSIT
FSIP
BSCN
 1 1 1 /
 /
BSIP
 1 1 1 /
 /

RPTSMRY
 1 /

SCHEDULE   ============================================================

RPTSCHED
’PRES’ ’SWAT’ ’RESTART=2’ ’FIP=2’ ’WELLS=2’ ’SUMMARY=2’ ’CPU=2’ ’WELSPECS’
’NEWTON=2’ ’PBLK’ ’SALT’ ’PLYADS’ ’RK’ ’FIPSALT=2’ /

WELSPECS
’I’  ’G’   1  1  4000  ’WAT’  0.0  ’STD’  ’SHUT’  ’NO’  /
```

```
'P'  'G'  25 25  4000  'OIL'  0.0  'STD'  'SHUT'  'NO'  /
/

COMPDAT
'I       '   1   1   1   10 'OPEN'   0  .0   1.0 /
'P       '  25  25   1   10 'OPEN'   0  .0   1.0 /
/

WCONPROD
'P' 'OPEN' 'BHP' 5* 3999.0 /
/

WCONINJE
'I' 'WAT' 'OPEN' 'RATE' 3500.0 /
/

WPOLYMER
 'I' 5.0  0.0 /
 /

TSTEP
1.0 9.0 2*20.0 50.0 100.0 5*200.0 2*500.0 800.0/

END
```

File name: POLYMER-3.DATA

```
RUNSPEC
TITLE
TEST 25 BY 25 BY 10 OIL/WATER/POLYMER SYSTEM

DIMENS
   25   25    10  /

OIL

WATER

POLYMER

BRINE

METRIC

TABDIMS
    1    1   50    2    2   20 /

REGDIMS
    2    1    0    0 /

WELLDIMS
```

```
    2    10    1    2 /

START
   1 'JAN' 1983  /

NSTACK
    8 /

UNIFOUT

UNIFIN

GRID      ==============================================================

EQUALS
      'DX'    20      /
      'DY'    20      /
      'PORO'  0.3       /

      'DZ'    10        1  25  1  25  1  4  /
      'PERMX'  300 /
      'PERMY'  300 /
      'PERMZ'  30 /
      'MULTZ'  1.0      /
      'TOPS'   4000/



      'DZ'    5        1  25  1  25  5  8  /
      'PERMX'  150 /
      'PERMY'  150 /
      'PERMZ'  30 /
      'MULTZ'  1.0      /


      'DZ'    20        1  25  1  25  9  10  /
      'PERMX'  300 /
      'PERMY'  300 /
      'PERMZ'  30 /


/
RPTGRID
-- Report Levels for Grid Section Data
--
'PORO'
'PORV'
 /

PROPS     ==============================================================

SWFN
    .20    .0    4.0
    .7    0.7    2.0
   1.0    1.0    0.0
/
```

```
SOF2
    .3000  .0000
    .8000 1.0000
/

PVTW
  .0  1.0  3.03E-06  0.06  0.0 /

PVDO
  .0     1.0     6.0
 8000.0   .92    6.0
/

ROCK
 4000.0    .30E-05 /

DENSITY
 52.0000  64.0000   .04400 /

SALTNODE
  0.0
  0.85
/

PLYVISCS
  0.0  1.0
       1.0/
  5.0  100.0
       25.0/
/

PLYROCK
   0.1   1.5  1000.0  2  0.005 /

ADSALNOD
  0.0
  0.85
/

PLYADSS
  0.0  0.000
       0.000/
  2.0  0.0015
       0.0150/
  8.0  0.0025
       0.0250/
 /

PLMIXPAR
 1.0 /

PLYMAX
 5.0  0.85 /

RPTPROPS
```

```
 -- PROPS Reporting Options
 --
 'PLYVISCS'
/

SOLUTION   ==============================================================

PRESSURE
  6250*4000.0 /

SWAT
  6250*0.301 /

SALT
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
 /
RPTSOL
```

```
'PRES' 'SOIL' 'SWAT' 'RESTART=2' 'PBLK'
'SALT' 'PLYADS' 'FIPPLY=2' /


SUMMARY    ============================================================

RUNSUM

FWPR
FOPR
FOPT
FWCT
FCPR
FCPT
FCIR
FCIT
FCIP
FCAD
GCPR
 'G' /
GCPT
 'G' /
GCIR
 'G' /
GCIT
 'G' /
WCPR
 'P' /
WCPT
 'P' /
WCIR
 'I' /
WCIT
 'I' /
CCFR
 'P' 25 25  1 /
 /
CCPT
 'P' 25 25  1 /
 /
CCIT
 'I'  1  1  1 /
 /
RCIP
 1 2 /
RCFT
 1 2 /
 /
RCAD
 1 2 /
BCCN
 1 1 1 /
 /
BCIP
 1 1 1 /
 /
```

```
BCAD
 1 1 1 /
 /
FSPR
FSPT
FSIR
FSIT
FSIP
BSCN
 1 1 1 /
 /
BSIP
 1 1 1 /
 /

RPTSMRY
 1 /

SCHEDULE   =============================================================

RPTSCHED
’PRES’ ’SWAT’ ’RESTART=2’ ’FIP=2’ ’WELLS=2’ ’SUMMARY=2’ ’CPU=2’ ’WELSPECS’
’NEWTON=2’ ’PBLK’ ’SALT’ ’PLYADS’ ’RK’ ’FIPSALT=2’ /

WELSPECS
’I’  ’G’   1  1  4000  ’WAT’  0.0  ’STD’  ’SHUT’  ’NO’  /
’P’  ’G’  25 25  4000  ’OIL’  0.0  ’STD’  ’SHUT’  ’NO’  /
/

COMPDAT
’I       ’   1   1   1   10 ’OPEN’   0  .0   1.0 /
’P       ’  25  25   1   10 ’OPEN’   0  .0   1.0 /
/

WCONPROD
’P’ ’OPEN’ ’BHP’ 5* 3999.0 /
/

WCONINJE
’I’ ’WAT’ ’OPEN’ ’RATE’ 3500.0 /
/

WPOLYMER
 ’I’ 5.0  0.425 /
 /

TSTEP
1.0 9.0 2*20.0 50.0 100.0 5*200.0 2*500.0 800.0/

END
```

File name: POLYMER-4.DATA

```
RUNSPEC
TITLE
TEST 25 BY 25 BY 10 OIL/WATER/POLYMER SYSTEM

DIMENS
   25   25    10  /

OIL

WATER

POLYMER

BRINE

METRIC

TABDIMS
    1    1   50    2    2   20 /

REGDIMS
    2    1    0    0 /

WELLDIMS
    2    10    1    2 /

START
   1 'JAN' 1983  /

NSTACK
    8 /

UNIFOUT

UNIFIN

GRID      ==============================================================

EQUALS
      'DX'    20      /
      'DY'    20      /
      'PORO'  0.3       /

      'DZ'    10        1  25  1  25  1  4  /
      'PERMX'  300 /
      'PERMY'  300 /
      'PERMZ'  30 /
      'MULTZ'  1.0      /
      'TOPS'   4000/



      'DZ'    5        1  25  1  25  5  8  /
      'PERMX'  150 /
      'PERMY'  150 /
      'PERMZ'  30 /
      'MULTZ'  1.0      /


      'DZ'    20        1  25  1  25  9  10  /
      'PERMX'  300 /
      'PERMY'  300 /
      'PERMZ'  30 /


/
RPTGRID
-- Report Levels for Grid Section Data
--
'PORO'
'PORV'
 /

PROPS     ==============================================================

SWFN
    .20    .0    4.0
    .7    0.7    2.0
   1.0    1.0    0.0
/

SOF2
    .3000  .0000
    .8000 1.0000
/

PVTW
  .0  1.0  3.03E-06  0.6  0.0 /

PVDO
  .0     1.0     6.0
 8000.0   .92    6.0
/

ROCK
 4000.0    .30E-05 /

DENSITY
 52.0000  64.0000   .04400 /

SALTNODE
  0.0
  0.85
/

PLYVISCS
  0.0  1.0
       1.0/
  5.0  100.0
       25.0/
/
```

```
PLYROCK
   0.1   1.5  1000.0  2  0.005 /

ADSALNOD
  0.0
  0.85
/

PLYADSS
  0.0  0.000
       0.000/
  2.0  0.0015
       0.0150/
  8.0  0.0025
       0.0250/
 /

PLMIXPAR
 1.0 /

PLYMAX
 5.0  0.85 /

RPTPROPS
 -- PROPS Reporting Options
 --
 'PLYVISCS'
/

SOLUTION   =============================================================

PRESSURE
  6250*4000.0 /

SWAT
  6250*0.301 /

SALT
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  375*0.1
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
  4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
```

```
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  375*0.1
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  375*0.1
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  375*0.1
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  375*0.1
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  375*0.1
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  4*0.1 21*0.85  4*0.1 21*0.85
   4*0.1 21*0.85  375*0.1
 /
RPTSOL
'PRES' 'SOIL' 'SWAT' 'RESTART=2' 'PBLK'
'SALT' 'PLYADS' 'FIPPLY=2' /


SUMMARY    =============================================================

RUNSUM

FWPR
FOPR
FOPT
FWCT
FCPR
FCPT
FCIR
FCIT
FCIP
FCAD
GCPR
 'G' /
GCPT
 'G' /
GCIR
 'G' /
GCIT
 'G' /
WCPR
 'P' /
WCPT
 'P' /
WCIR
 'I' /
```

```
WCIT
 ’I’ /
CCFR
 ’P’ 25 25  1 /
 /
CCPT
 ’P’ 25 25  1 /
 /
CCIT
 ’I’  1  1  1 /
 /
RCIP
 1 2 /
RCFT
 1 2 /
 /
RCAD
 1 2 /
BCCN
 1 1 1 /
 /
BCIP
 1 1 1 /
 /
BCAD
 1 1 1 /
 /
FSPR
FSPT
FSIR
FSIT
FSIP
BSCN
 1 1 1 /
 /
BSIP
 1 1 1 /
 /

RPTSMRY
 1 /

SCHEDULE   =============================================================

RPTSCHED
’PRES’ ’SWAT’ ’RESTART=2’ ’FIP=2’ ’WELLS=2’ ’SUMMARY=2’ ’CPU=2’ ’WELSPECS’
’NEWTON=2’ ’PBLK’ ’SALT’ ’PLYADS’ ’RK’ ’FIPSALT=2’ /

WELSPECS
’I’  ’G’   1  1  4000  ’WAT’  0.0  ’STD’  ’SHUT’  ’NO’  /
’P’  ’G’  25 25  4000  ’OIL’  0.0  ’STD’  ’SHUT’  ’NO’  /
/

COMPDAT
’I       ’   1   1   1   10 ’OPEN’   0  .0   1.0 /
’P       ’  25  25   1   10 ’OPEN’   0  .0   1.0 /
```

```
/

WCONPROD
’P’ ’OPEN’ ’BHP’ 5* 3999.0 /
/

WCONINJE
’I’ ’WAT’ ’OPEN’ ’RATE’ 1000.0 /
/

WPOLYMER
 ’I’ 5.0  0.425 /
 /

TSTEP
1.0 9.0 2*20.0 50.0 100.0 5*200.0 2*500.0 800.0/

WPOLYMER
 ’I’ 5.0  0.000 /
 /

TSTEP
1 2*500.0 799.0/

END
```