The National IOR Centre of Norway

Temporary Project Report (30/11-2021)

1.1.15 Upscaling of polymer and Smart Water processes

Project number, location:	PR-10371-17, University of Stavanger (UiS)
Theme, task:	1, 1
Project duration:	March 2019 - March 2022
Project manager:	Oddbjørn Nødland (postdoctoral researcher)
PhD students and postdocs:	Oddbjørn Nødland
Other key personnel:	Aksel Hiorth, Arild Lohne, Arne Stavland, Ove Sævareid

Executive summary

Numerous laboratory experiments have shown that modifying the ionic composition of the injection brine can lead to improved oil recovery. However, the mechanisms responsible are not fully understood. To complement observational work, this project uses the IORCoreSim simulator to explore mechanisms that may be important during "Smart Water" flooding.

The project also investigates workflows for field scale polymer flooding. Most simulators for polymer flooding do not fully account for effects of apparent shear thickening and mechanical degradation in the near well zone, which could lead to poor forecasts. Particular attention is paid to how these effects can be modelled with commercial and open-source tools. Comparisons are made with IORCoreSim.

Project background

The project is a postdoctoral research project. Collaborators are from UiS and NORCE.

Results

Findings are presented in the following pages, presented under four main headings:

- *"Simulating pH changes during core reactivity tests in sandstones"*
  - A simulation study to see whether a simple cation exchange model can explain observed pH changes during one-phase core floods in reservoir sandstones (suggested answer: no)
- *"Near-well polymer flow in IORCoreSim and INTERSECT"*
  - A presentation of the IORCoreSim well model for polymer flooding.
  - Application to a small sector-scale example in Cartesian geometry, mimicking radial flow.
  - Comparison with analytical calculations and INTERSECT (IX).
- *"IORCoreSim case study in a channeled reservoir"*
  - A slightly extended version of the case study reported in the "recommended practices" report on polymer upscaling (i.e., mostly the same content)
- *"Extending polymer flooding functionality in OPM"*
  - A brief presentation of efforts to add extra polymer functionality to the Open Porous Media (OPM) simulator.

At least for now, there is no general conclusion at the end of the report, but we briefly mention plans for future work and dissemination.

I. Simulating pH changes during core reactivity tests in sandstones

Injection of low salinity (LS) water into sandstone cores is frequently accompanied by an increase in effluent pH. It is well known that chemical reactions governing rock wettability are pH-dependent, hence it has been suggested that local pH changes at mineral surfaces could trigger wettability alteration, and in turn lead to improved oil production (Austad, RezaeiDoust and Puntervold 2010) (RezaeiDoust, Puntervold og Austad 2011) .

What can we explain with a «generic» cation exchange model?

One proposed mechanism for wettability alteration is cation exchange. An initial study was conducted to ascertain whether a simple cation exchange model can explain observed pH trends in one-phase corefloods. To this end, reactive transport simulations were performed with IORCoreSim. The simplest geochemical models ignore mineral-specific effects, and account for cation exchange by introducing a generic ion exchanger, X^-, to the model equations. If sodium, calcium, and protons are the only cations competing for access to the surface, we need two linearly independent reactions, e.g.:

$$Na^+ + \frac{1}{2}Ca - X_2 \leftrightarrow NaX + \frac{1}{2}Ca^{2+}$$

$$Na^+ + HX \leftrightarrow NaX + H^+$$

In IORCoreSim, the Gaines-Thomas convention is used to describe ion exchange (Gaines Jr og Thomas 1953). Selectivity coefficients ("equilibrium constants") for cation exchange reactions are input indirectly, by reformulating them as half-reactions involving a fictive "free surface" species, X^-:

$$NaX \leftrightarrow Na^+ + X^-$$

$$CaX_2 \leftrightarrow Ca^{2+} + 2X^-$$

$$HX \leftrightarrow H^+ + X^-$$

For the work presented in this report, constant values were assumed for the cation exchange selectivity coefficients. The sodium reaction was used as a reference (i.e., $\log K_{NaX} = 0$), while the selectivity coefficient for the calcium and proton half-reactions were:

$$\log K_{CaX_2} = -0.8 \text{ (Appelo og Postma 2004)}$$

$$\log K_{HX} = -4.6 \text{ (Wieland, et al. 1994)}$$

Note that the selectivity coefficient for proton exchange is based on a more advanced model (Wieland, et al. 1994) than the one considered here.

Simulation results compared to data

Data reported by (Aksulu, et al. 2012) were used for comparison with IORCoreSim; specifically, a set of one-phase core floods conducted in the same reservoir core plug, named *RC2*. The core floods were performed at three temperatures: 40 °C, 90 °C, and 130 °C. At each temperature, the core was subjected to a cycle of High Salinity (HS) → LS → HS injection at a constant flow rate of 4 pore volumes (PV) per day. The HS brine consisted of 1.54 M NaCl and 90 mM $CaCl_2$, while the LS brine was pure NaCl (17.1 mM).

An initial simulation was conducted under the following (additional) assumptions:

- T=25 °C
- No dissolved CO_2
- CEC = 0.1 eq/L PV (constant cation exchange capacity)

Figure 1 compares simulated effluent pH to the experimental data points. The pH gradient, ΔpH, increased by ~3 for the injection cycle at °40C. This cannot be reproduced with the model; the simple ion exchange model can explain a reversible pH jump of at most ~2 units following a ~100 times reduction in the total salinity.

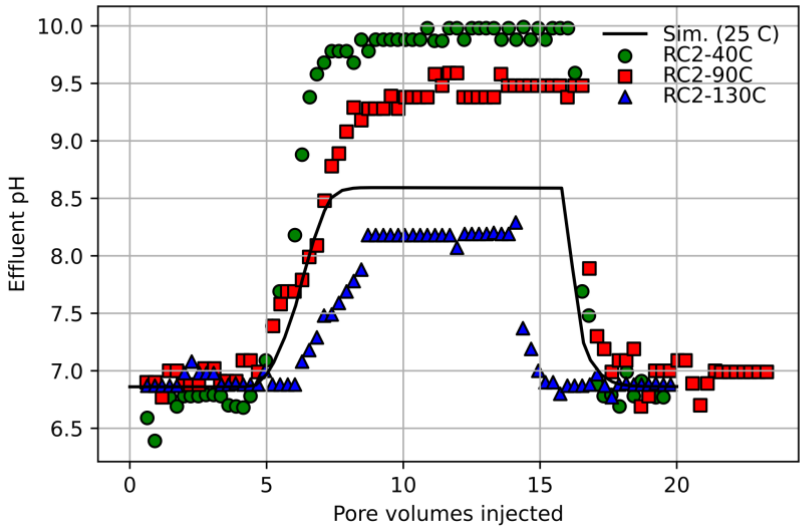

Figure 1: Example simulation at T=25 °C compared to data for core floods performed at different temperatures. Before measuring the pH, the collected samples were cooled to room temperature.

Figure 2 shows that that model cannot explain the temperature trends seen in the data:

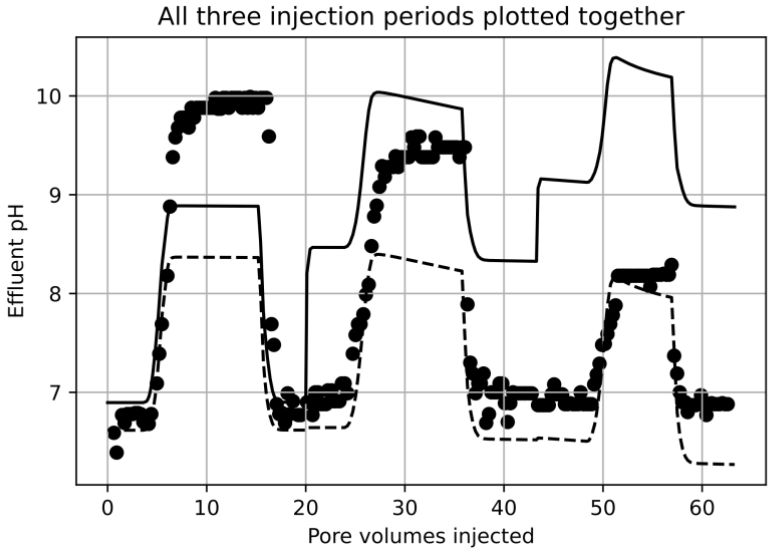

Figure 2: *Effluent pH during HS→LS→HS injection at three different temperatures (40°C, 90°C, 130°C). Circles: pH measurements taken at room temperature (same as in* Figure 1 *). Dashed line: Effluent pH reported by IORCoreSim for in-situ temperature conditions. Solid line: Simulated effluent pH corrected to room temperature by performing an additional geochemical equilibrium calculation.*

The simulated pH is very sensitive to temperature (solid line), but in contrast to what is observed, there is a large jump in effluent pH immediately after increasing the temperature, i.e., even at the same injected salinity. As indicated by the dashed line in Figure 2, this pH jump is caused by a slight increase in the simulated *in-situ* pH, which in turn is a consequence of cation realignment on the cation exchanger. In other words, a small difference in the in-situ pH leads to a *dramatic* increase in the predicted pH at room temperature. The explanation is that the dominant reaction in the geochemical model is water dissociation, which is an endothermic reaction:

$$H_2O + heat \leftrightarrow H^+ + OH^-$$

As the temperature increases, more protons and hydroxide ions are formed, which lowers the neutral pH (e.g., from ~7 at room temperature to about ~6 at 100 °C). When going the other way - correcting a pH value measured at high temperature to lower temperature - the pH increases, and in the absence of any strong buffers, this pH change can be quite large even for small variations in the ion concentrations.

If we disregard the initial pH jump, and only focus on what happens after LS brine is injected, the observed trend is that of a decreasing ΔpH with increasing temperature. To some extent this happens in the simulations too, but there is not a quantitative match between model and data. A few test runs were conducted to see whether the match could be improved by making the cation exchange reactions temperature dependent, but a satisfactory match was still not achieved. To overcome the dominance of the water dissociation reaction in the model, an unrealistically large temperature effect had to be assumed.

Figure 3 shows that the calculated pH is sensitive to the previous history of the core plug. If we model each HS→LS→HS cycle with a separate simulation run, there is no longer a jump in pH as the temperature is increased, and the trend in ΔpH is opposite to what is observed:

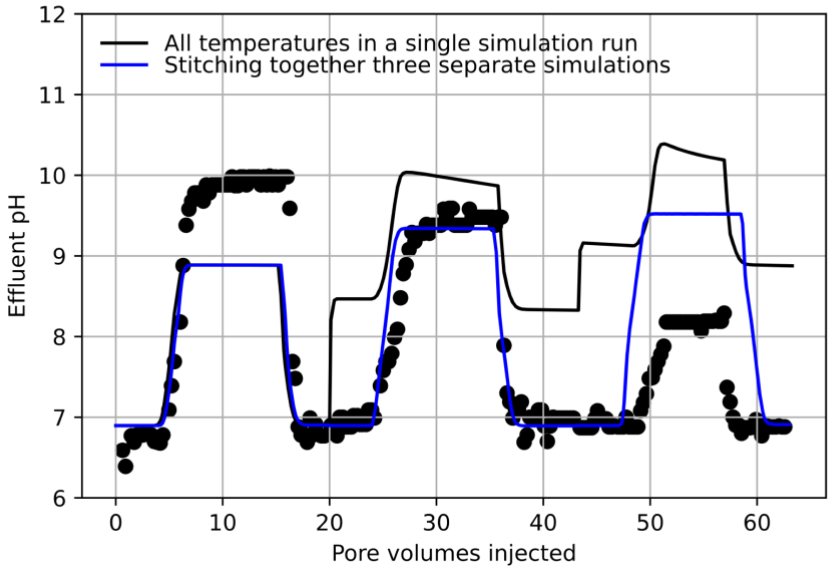

Figure 3: Effluent pH during HS→LS→HS injection at three different temperatures (40°C, 90°C, 130°C). Circles: pH measurements taken at room temperatures. Black line: The simulated effluent pH corrected to room temperature (same as in Figure 2). Blue line: The simulated pH when running three separate simulation runs, one for each temperature, and then stitching the curves together. Again, the simulated pH was corrected to room temperature by performing geochemical equilibrium calculations with the produced water as input.

Figure 4 gives an indication of why this happens: at the end of each HS→LS→HS cycle, the total number of protons on the exchanger is larger than at the beginning of the cycle. Thus, if we continue to simulate at a new temperature without equilibrating all grid blocks with the same HS brine, there will be gradients in adsorbed concentrations and pH inside the core which affect subsequent calculations.

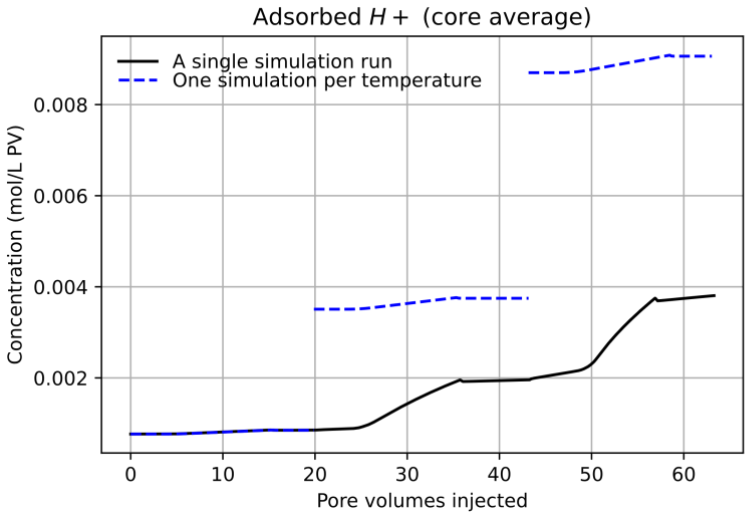

Figure 4: Average amount of protons on the cation exchanger vs time for the two simulation runs in Figure 3. The blue dashed lines represent the case when we equilibrate the cation exchanger with the HS brine separately for each temperature. The solid black line shows that we get a different result if we instead modify the temperature during a single simulation run.

Finally, the presence of minerals to buffer the pH is very important in experiments such as these. Some simulation runs were conducted to evaluate the effect of mineral dissolution / precipitation reactions. For instance, it was found that adding calcite leads to a large increase in the modelled pH, both during HS and LS injection (Figure 5). This suggests that very little calcite was present. Since the same core plug had been used in several previous experiments, it is likely that any calcite initially present had already been flushed out.

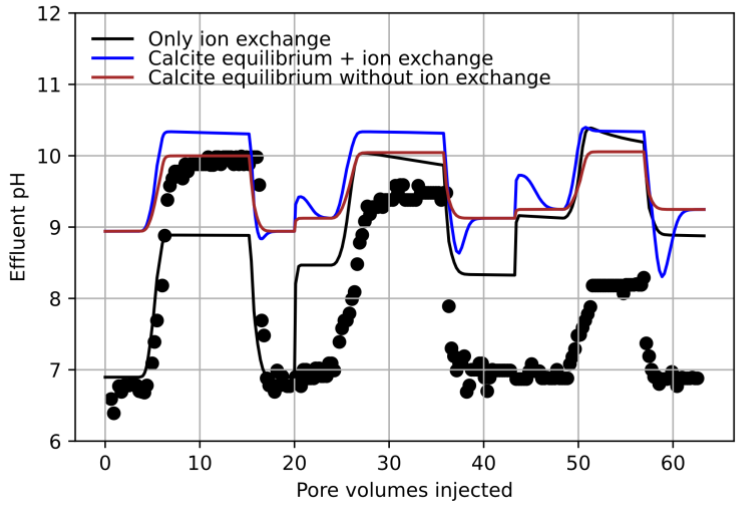

Figure 5: The impact of calcite dissolution / precipitation in the model. Circles: Effluent pH of samples taken during HS→LS→HS injection at three different temperatures (40°C, 90°C, 130°C). The measurements were performed at room temperature. Solid lines: Simulated effluent pH corrected to room conditions.

To sum up, it was not possible to obtain a quantitative match between the observed pH and the proposed cation exchange model. The impact of temperature was especially hard to model: Because the model did not include any strong pH buffers, simulated results were completely dominated by the effect of water self-ionization. Adding calcite led to an even poorer fit between model and data, which indicates that little calcite was present in the experiments.

In future work, models accounting for multiple exchange sites and mineral-specific effects should be considered, e.g., effects of clay minerals and Feldspars. The interpretation of the data is further complicated by the fact that pH is CO_2-dependent. This should be investigated in more detail, e.g., potential effects of CO_2-contamination as effluent samples are collected and cooled before measurements are taken.

II. Near-well polymer flow in IORCoreSim and INTERSECT

The flow resistance of polymer solutions in porous media depends on the in-situ conditions, such as flow rate, permeability, temperature, and fluid chemistry. The effect of flow rate, or shear rate, can be very important to capture, especially when making injectivity forecasts. Typically, biopolymer solutions act as either shear thinning or near-Newtonian fluids, while synthetic polymers additionally display dilatant, or *apparent shear thickening*, behaviour at large rates (Chauveteau 1981). Depending on the value of the local shear rate, the flow resistance may vary by several orders of magnitude (Sorbie 1991).

The non-Newtonian polymer rheology is hard to model accurately at the field scale. This is because field scale simulations require the use of very large grid blocks, frequently on the order of tens to hundreds of metres in the horizontal direction. When using conventional numerical discretization techniques, fluid velocities near wells, and thus shear rates, are severely underestimated. An implication is that well pressures and aqueous phase pressures in the reservoir will be incorrectly computed. For shear thinning polymer solutions, the numerical error leans in the conservative direction, meaning that calculated injectivities are lower than what is really expected to be the case. For polymers exhibiting shear thickening, models are likely biased towards an overoptimistic forecast, though the exact outcome also depends on whether, and how much, the polymer is degraded (Nødland, et al. 2019).

Mechanical degradation is hard to capture in reservoir scale models. In a previous study (Nødland, et al. 2019), it was found that mm- or cm-sized grid blocks must be used to accurately resolve the IORCoreSim degradation equation in radial flow (Figure 6).

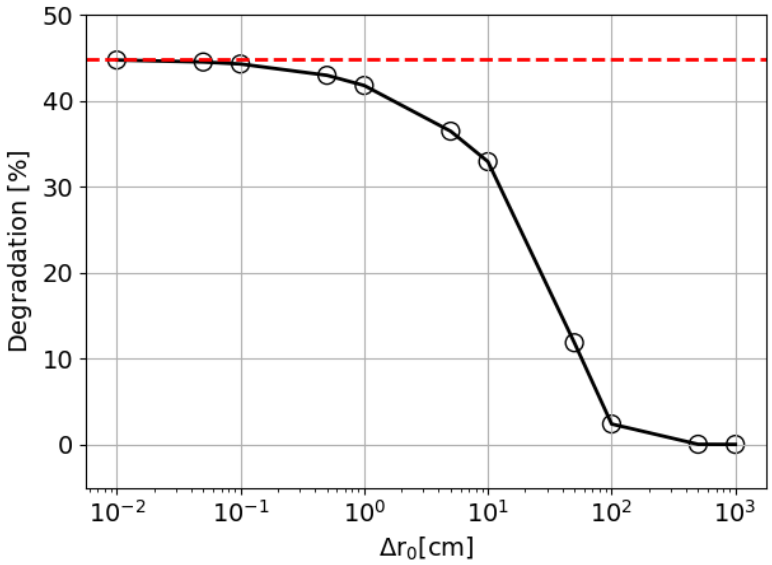

Figure 6: Example of how the IORCoreSim mechanical degradation model depends on numerical resolution. Hollow circles: Percentage reduction in modelled polymer molecular weight, at steady state, as a function of grid block size near the injector. A constant block size, Δr_0, was used for the first ~20 cm of the reservoir. Red, stippled line: Prediction of the same molecular weight based on analytical calculations (numerical integration). For more details, see (Nødland, et al. 2019), especially Fig.2.

IORCoreSim well model for polymer flooding

IORCoreSim has a special well model option for handling near-well polymer flow. The method is based on the calculation of an average apparent viscosity for grid blocks containing wells. Assuming radial flow close to a well, and constant permeability, the pressure difference between the well and a point further out in the reservoir is (Thompson og Reynolds 1997)

$$\Delta p(r,t) = p(r,t) - p(r_w,t) = \frac{1}{2\pi hk}\int_{r_w}^{r} \frac{q_t(r',t)}{r'\lambda_t(r',t)}\,dr',$$

where q_t is the total flow rate at radial distance r and time t, r_w is the well radius, h is the penetrated thickness, k is absolute permeability, and λ_t is the total (relative) mobility of the fluids. For single-phase flow involving polymer,

$$\frac{1}{\lambda_t(r,t)} = \eta(r,t)\cdot R_k(r,t)$$

where η is the apparent viscosity of the aqueous phase containing dissolved polymer, and R_k is a permeability reduction factor to account for restricted flow due to polymer adsorption. If the flow rate can be treated as constant, $q_t(r,t) = Q$:

$$\Delta p(r,t) = \frac{Q}{2\pi hk}\int_{r_w}^{r} \frac{\eta(r',t)\cdot R_k(r',t)}{r'}\,dr'.$$

For a given time,

$$\Delta p(r) = \frac{Q\overline{\eta R_k}}{2\pi hk}\cdot \ln\left(\frac{r}{r_w}\right).$$

in which $\overline{\eta R_k}$ provides a measure of the average mobility reduction factor, or resistance factor (RF) in the spatial interval from r to r_w; letting η_s denote the solvent viscosity, we have:

$$\overline{\eta R_k} \equiv \frac{1}{\ln\left(\frac{r}{r_w}\right)}\cdot\int_{r_w}^{r}\frac{\eta(r')\cdot R_k(r')}{r'}dr' = \frac{\eta_s}{\ln\left(\frac{r}{r_w}\right)}\cdot\int_{r_w}^{r}\frac{RF(r')}{r'}dr'.$$

For simplicity, near-well gradients in the permeability reduction factor are ignored, hence we simplify to $\overline{\eta R_k} = \overline{\eta}\cdot R_k$, where $\overline{\eta}$ is the average apparent viscosity in the radial interval. By performing the coordinate transformation $r' \rightarrow \mathrm{u} = \ln(\mathrm{r}')$, we get:

$$\overline{\eta} = \frac{1}{\ln\left(\frac{r}{r_w}\right)}\cdot\int_{r_w}^{r}\frac{\eta(r')}{r'}dr' = \frac{1}{\ln\left(\frac{r}{r_w}\right)}\cdot\int_{\ln(r_w)}^{\ln(r)}\eta(e^u)\,du.$$

In IORCoreSim, the areal-average radius, r_A, is used as upper limit for the integration; for a vertical well:

$$r_A = \sqrt{\frac{\Delta x\cdot\Delta y}{\pi}}.$$

The integral is approximated with the midpoint rule: By dividing the total interval from r_w to r_A into N subintervals, and letting $r_0, r_1, \ldots, r_n$ denote the subinterval edges, we get

$$\overline{\eta} \approx \frac{1}{\ln\left(\frac{r}{r_w}\right)}\cdot\sum_{i=1}^{N}\eta_i\cdot\ln\left(\frac{r_i}{r_{i-1}}\right),$$

where $\eta_i = \eta(\frac{r_{i-1}+r_i}{2})$ is the apparent viscosity evaluated at the midpoint of interval i. To calculate η_i, the in-situ shear rate at each interval midpoint, $\dot{\gamma}_i$, is needed. Assuming velocity to scale as $1/r$, we compute each $\dot{\gamma}_i$ from a single reference shear rate, $\dot{\gamma_m}$:

$$\dot{\gamma}_i = \dot{\gamma_m} \cdot \frac{r_{i-1} + r_i}{r_w + r_A}$$

The reference shear rate is taken to be the shear rate at the well block midpoint. In turn, this reference shear rate is estimated from a finite-difference discretization appropriate for approximating radial flow near the well, see the IORCoreSim manual for more details.

Accounting for mechanical degradation near wells

When mechanical degradation is included, the molecular weight of the polymer should vary with distance. Capturing this gradient directly is impossible with standard grid sizes used in field scale reservoir simulation, however it might not be necessary because in radial flow, almost all of the degradation takes place within the first few mm or cm after the injection point (Nødland, et al. 2019) (Stavland, et al. 2021).

As an approximation, the IORCoreSim well model uses the degradation equation proposed in (Lohne, et al. 2017) to iterate towards a single steady-state molecular weight for the polymer in the well block. The main difference from the regular degradation solver is that we employ an "integrated" or “effective” degradation rate, f_{rup}^{eff}, to account for varying residence times in different areas close to the well:

$$f_{rup}^{eff} = \sum_{i=1}^{N} f_{rup,i} \cdot \frac{r_i^2 - r_{i-1}^2}{r_A^2 - r_w^2} \cdot \Delta t.$$

In the above expression, Δt is the time step and $f_{rup,i}$ is the molecular rupturing rate computed by assuming shear rate $\dot{\gamma}_i$ and apparent viscosity $\eta_i = \eta(\dot{\gamma}_i)$.

Example: Approximating radial flow with Cartesian grids in IORCoreSim

To exemplify the impact of numerical errors on polymer injectivity calculations, we revisit validation case 2 from (Li and Delshad 2014). This is a single-phase 2D scenario, in which a 0.3 wt% polymer solution is continuously injected from an injection well placed in the middle of a homogeneous and isotropic reservoir. The injection rate is kept constant at 4000 cubic feet per day. Sixteen peripheral production wells are placed out at a fixed radial distance from the injector (r_e = 350 ft) and set to operate at a constant pressure of 1200 psi. This is done to mimic a fixed pressure boundary condition at the exterior radius. Polymer adsorption and inaccessible pore volume effects are ignored. The apparent viscosity of the polymer solution is calculated with Meter's equation (Meter and Bird 1964).

Figure 7 compares injection well pressures for three different simulation runs: two runs made with a coarse Cartesian grid ($\Delta x = \Delta y$= 70 ft), and one conducted in a much finer, radial grid. The well pressures are plotted versus the number of pore volumes injected in the radial grid. For reference, the figure also includes an analytical solution based on assuming a perfect frontal displacement.

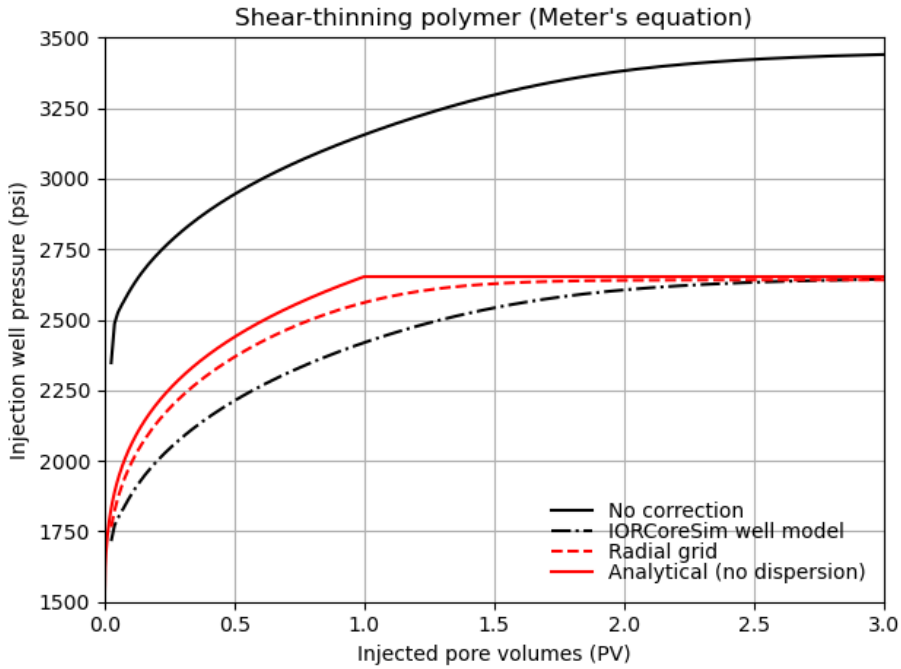

Figure 7: Injector bottom-hole pressure during polymer injection for (Li and Delshad 2014) example. The figure demonstrates the effect of using a Cartesian grid to mimic radial flow, as well as the impact of using the IORCoreSim polymer well model.

Using the coarse grid without modification overestimates the well pressure by about ~30 %. Even the fine scale radial model gives some error, mostly due to numerical dispersion of the polymer front (Lantz 1971). However, at least in this case, the IORCoreSim well model yields the correct well pressure after several pore volumes of polymer injection. On the other hand, Figure 8 shows that if we use the Peaceman radius as upper limit for the numerical integration, the well pressure becomes significantly lower.

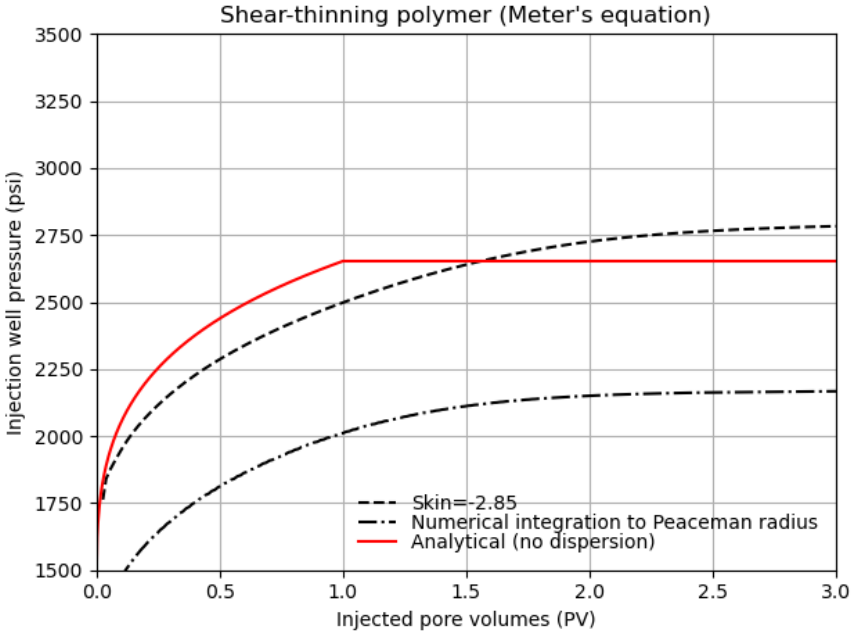

Figure 8: Injector bottom-hole pressure during polymer injection for (Li and Delshad 2014) example. The dashed-dotted line shows the result when we use the Peaceman radius as upper limit in the IORCoreSim well model. The dashed line shows an example where the near-well shear thinning effect is approximated by assuming a negative skin factor for the well.

In a square Cartesian grid, the areal-average radius is larger than the Peaceman radius by a factor of $\frac{4}{\sqrt{2\pi}e^{-\gamma}} \approx 2.84$ (Peaceman 1983), which leads to a substantial decrease in the average apparent viscosity compared to when r_A is used. Because (Li and Delshad 2014) integrated out to the Peaceman radius, we are not able to reproduce their reported results.

For comparison, Figure 8 also shows the outcome of using the negative skin factor approach proposed by (Li, R. and Delshad 2017),

$$skin = \frac{\int_{r_w}^{r_o} \frac{\eta(r)}{r}\, dr}{\eta(r_A)} - \ln\left(\frac{r_o}{r_w}\right),$$

where r_o is the Peaceman radius.

Example: Approximating radial flow in INTERSECT

Figure 9 shows two examples of using INTERSECT to simulate the same scenario. In one case, the simulator greatly overpredicts the well pressure (left plot, blue solid line). The explanation is that in this run, the in-situ shear rate for the well blocks was computed from arithmetic averages of neighbouring edge-velocities in the x- and y-directions. Due to the symmetry of the problem, these component velocities are similar in magnitude, but have opposite signs, hence the shear rate becomes zero, and the polymer is falsely predicted to be in the Newtonian flow regime ($\eta \approx \eta_0$).

If quadratic averaging is used instead, results are more in line with IORCoreSim (not shown). While this example scenario is not very representative for real field cases, it shows that it is important to be aware of how shear rates are calculated in the simulator.

Attempts were made to capture near-well polymer flow by using Python scripts to override the default well block calculations in INTERSECT; that is, by modifying grid block properties from outside the simulator. In INTERSECT 2020.1, it is impossible to modify the apparent viscosity from the outside during a simulation run, therefore the endpoint relative permeability to water was modified instead. Figure 9 demonstrates that for one-phase flow, it was possible to obtain a very good match with IORCoreSim; see especially the right plot.

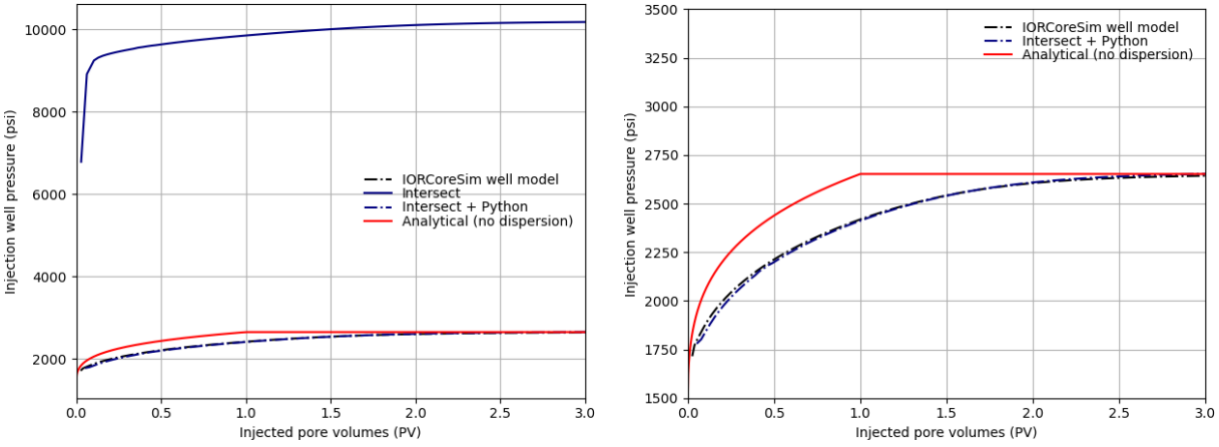

Figure 9: Injector bottom-hole pressure during polymer injection for (Li and Delshad 2014) example: Comparison between IORCoreSim well model and two INTERSECT runs. Note that no maximum limit was set for the pressure in these simulations.

Attempts to generalize the approach to two-phase flow of water and oil has so far been unsuccessful. Even for one-phase flow, using Python to modify well block properties led to a large increase in the CPU running times, due to extra convergence difficulties.

However, it should be mentioned that INTERSECT also has a special well block option: the user can input, for each well-cell connection, an effective radius from the well at which to calculate the well block shear rate. The simulator will then compute the magnitude of the well block velocity as the maximum of the ordinary grid block velocity and

$$|u_w| = \frac{|q_w^k|}{2\pi r_{eff} h^k},$$

where q_w^k is the water volumetric flow rate through the connection, and h^k is the connection thickness. For shear thinning polymers, IORCoreSim results can be approximated by choosing the effective radius so that the calculated viscosity at r_{eff} equals the average viscosity in the IORCoreSim well model, see Figure 10 for an example.

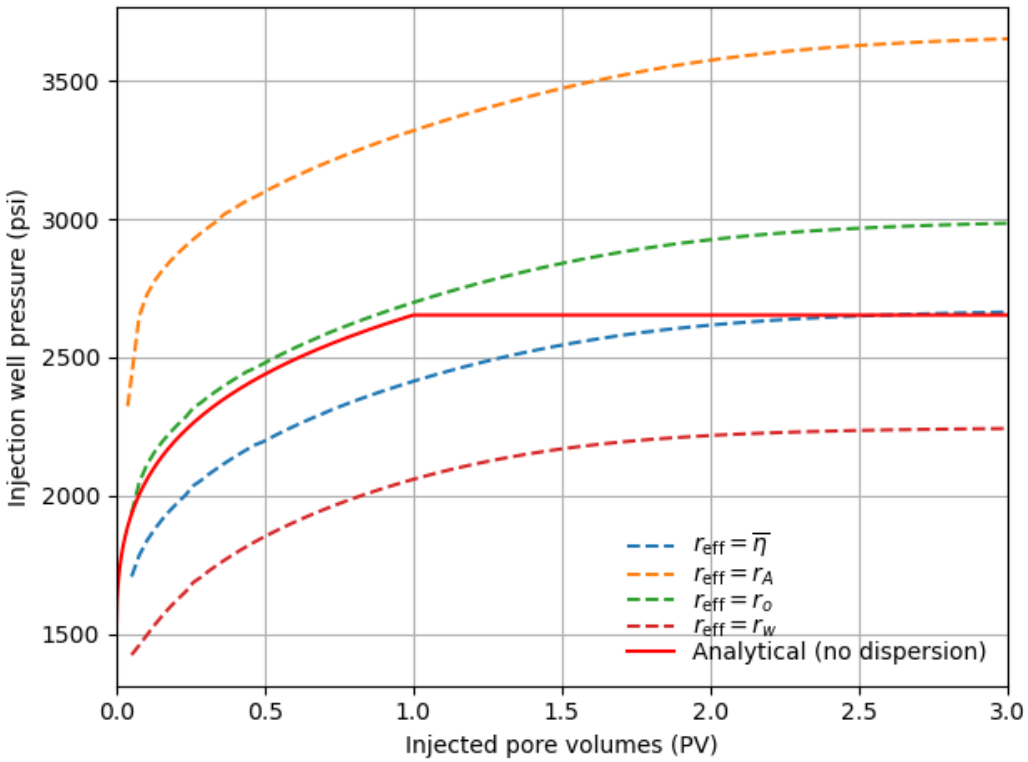

Figure 10: Injector bottom-hole pressure for (Li and Delshad 2014) example. Comparison of analytical solution assuming a perfect frontal displacement (red, solid lines) and four INTERSECT simulations, using different choices of the effective well-block radius. The blue, stippled line represents the case when r_{eff} was selected to produce the same apparent viscosity as the IORCoreSim well model (for a well flow rate of 4000 ft^3/d).

Note that the effective radius option presupposes a constant flow rate at the well. However, it may still give acceptable results when flow rates are changing in time, e.g., for constant bottom-hole pressure injectors, provided the viscosity curve is not too sensitive to variations in near-well flow rates.

III. IORCoreSim case study in a channeled reservoir

For performance reasons we choose a relatively small field case: the first permeability realization of the Egg model geological ensemble (Jansen, et al. 2014). The well placement is shown in Figure 11. The eight injectors and four producers are all vertical, and they are perforated throughout the entire formation thickness (28 m). Porosity is assumed uniform, φ = 0.2. For ease of interpretation, a constant initial water saturation is used in all active grid cells, $S_{wi} = 0.25$.

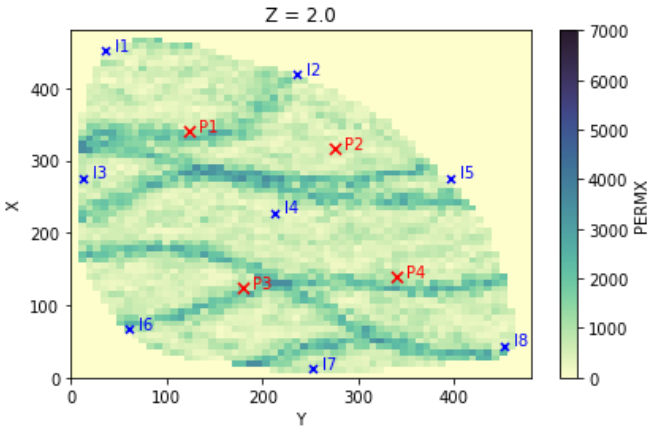

Figure 11: Horizontal permeability in top layer of the Egg model. There are 60x60x7 cells, of which 18553 are active. Grid block sizes are 8x8x4 meter. The seven layers have a strong vertical correlation, such that the permeability field is almost 2-dimensional (82 mD-7 D, arithmetic average 1.12 D). The ratio of horizontal to vertical permeability is ~10.

Rock-fluid properties

We consider injection of synthetic sea water (SSW) into a reservoir filled with oil and, for simplicity, SSW. The temperature is assumed constant, T=20°C. The water viscosity is assumed to be 1.07 mPas, while the oil viscosity is 20.0 mPas. The oil density is 900 kg/m3. Capillary pressure is ignored in the simulations.

A set of generic relative permeability curves, representative of mixed-wet formations, is used (Figure 12, left plot); note that these are different from the ones reported in (Jansen, et al. 2014). Capillary pressure is ignored in the simulations.

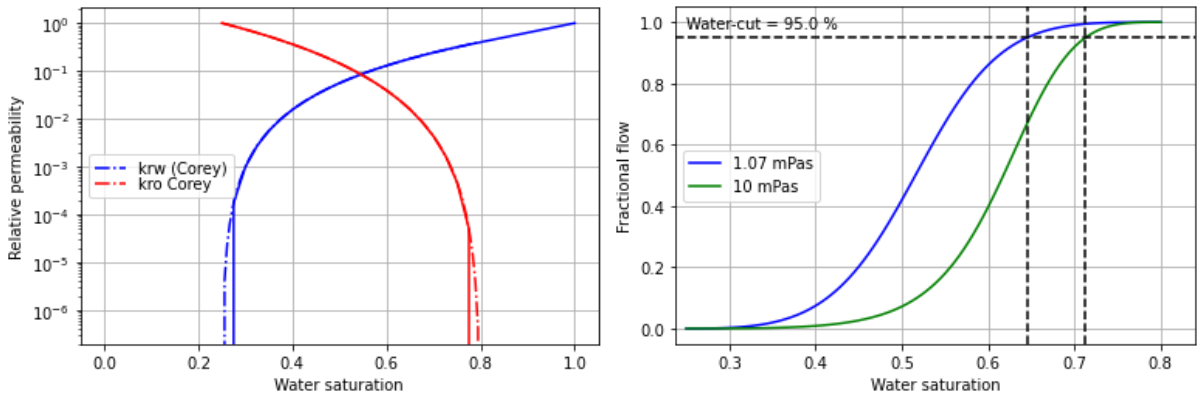

Figure 12: Left: Relative permeability curves used in the Egg model simulations (solid lines, modified compared to Jansen et al. 2014). Also shown are approximate smoothed curves computed with Corey power-law functions (dashed lines). Right: Fractional flow curves calculated from the Corey functions.

From the Welge tangent method (Welge 1952), the front saturation during a linear displacement with sea water is approximately 47 %, while the average saturation behind the front at breakthrough is 53 % (Figure 12, right plot). The expected oil recovery factor during core flooding is 40-47 % of the original oil in place (OOIP) after injecting 0.5-1 pore volumes (Figure 13). If the viscosity of the injected water is increased to 10 mPas, these numbers increase by roughly 15 %, which indicates a good EOR potential for polymer flooding.

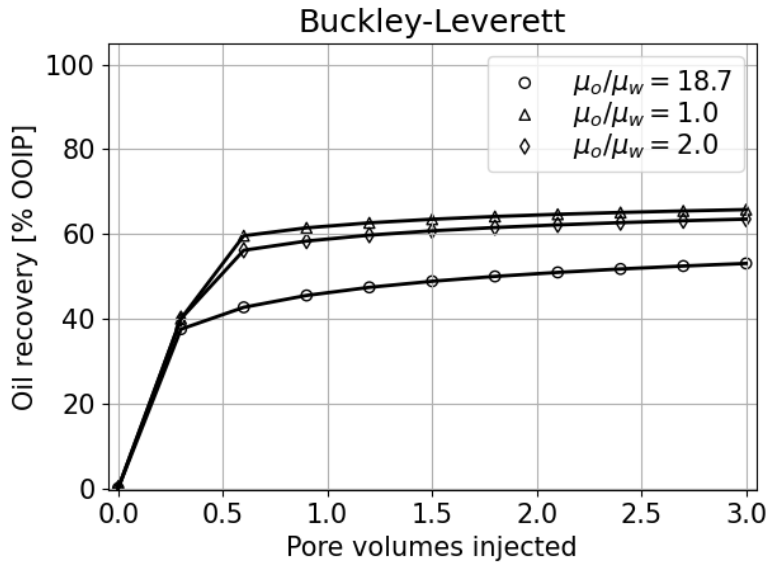

Figure 13: Oil recovery computed from Buckley-Leverett theory, i.e., as expected in a linear core flood with no gravity or capillary pressure effects. The circles correspond to μ_w = 1.07 mPas, μ_o = 20 mPas ("baseline waterflooding")

For cases with polymer, the injected polymer concentration is always 1500 ppm. We consider two different HPAM polymers: FLOPAAM 3530S, with an initial molecular weight 15 MDa, and FLOPAAM 3630S, with molecular weight 20 MDa. The input polymer properties are taken from (Lohne, et al. 2017), with some notable exceptions; see discussion further below.

Table 1 contains additional details on the assumed rock- and fluid properties.

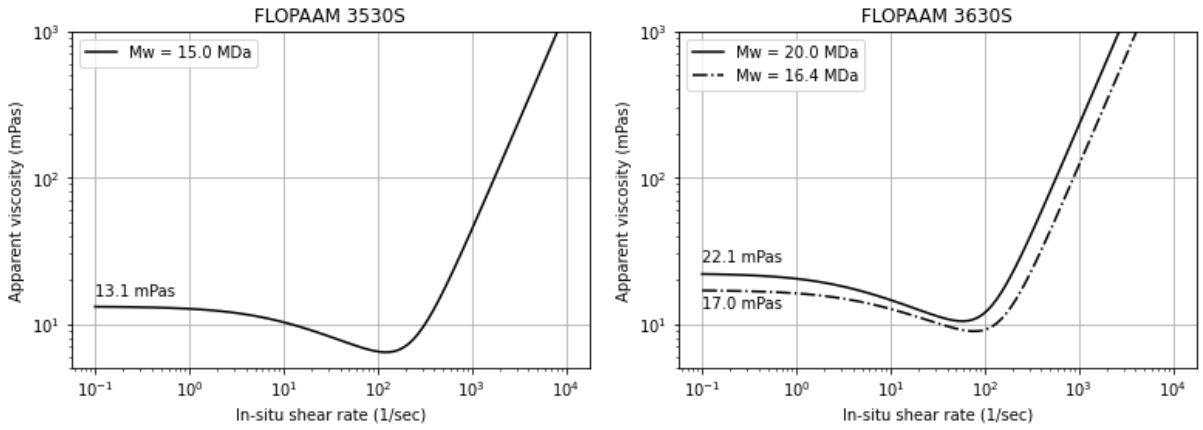

Figure 14: IORCoreSim apparent viscosity for two HPAM polymers dissolved in SSW (T=20°C, 1500 ppm). Note that the curves do not account for the effect of depletion layers. The stippled curve in the right plot corresponds to a ~18 % Mw reduction and indicates that the viscosity loss due to mechanical degradation is relatively small at the chosen flow rate.

Well schedule

The total simulation time is 10 years. Water is injected at constant rate 79.5 m3/day with an upper bottom-hole pressure limit of 271 bar, which is close to the estimated fracturing pressure when assuming a fracture pressure gradient of ~0.14 bar/m. All producers are set to operate at a constant bottom-hole pressure of 195 bar.

We first simulate a baseline waterflood, followed by a "screening run" in which the effects of polymer are approximated by injecting a Newtonian fluid of viscosity 10 mPas, about 10 times as viscous as water; this is in the right "ballpark" for the expected in-situ resistance factor (see Figure 14). For the screening run, as well as for simulations with actual polymer, injection of viscous brine is commenced after 5 years and terminated after 6 years (a slug of approximately ¼ pore volumes).

Simulation results

For waterflooding with SSW, the absolute oil recovery is 47.9 % OOIP after 10 years. This number increases to 51.8 % when injecting the 10 mPas Newtonian fluid, to 51.1 % when injecting the 3530S polymer, and to 52.4 % when injecting 3630S. The left plot of Figure 15 shows the *relative increase* of oil production as a function of time. For both polymers, most of the incremental oil is produced after commencing the second waterflood. For 3530S, the Newtonian model gives the same recovery as the polymer model during the first year of post-polymer water injection. In contrast, the response is much quicker for the 3630S polymer, which is no surprise considering the larger molecular weight of this polymer.

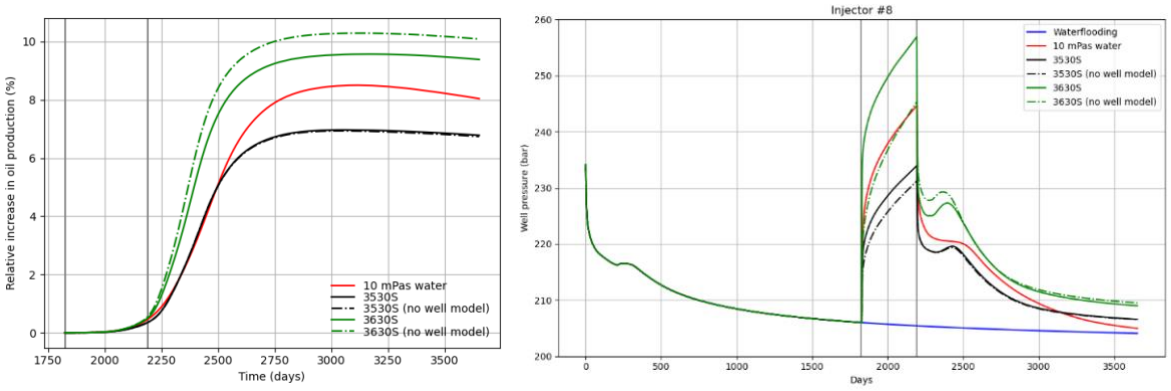

Figure 15: Left plot: Simulated (total) oil production as a function of time for two HPAM polymers and a Newtonian fluid. The y-axis shows the relative increase in oil recovery compared to the baseline waterflood. The grey vertical bars show the timing of the polymer slug. The dashed lines show results in which the IORCoreSim well model was not used. Right plot: Bottom-hole pressure at one of the injection wells.

Obviously, the additional flow resistance of 3630S compared to 3530S necessitates higher injection pressures to keep a prescribed injection rate. When injecting HPAM polymer solutions, one should always check whether shear thickening could cause problems. It is then important to use the well model implemented in IORCoreSim, which corrects for non-Newtonian flow and mechanical degradation close to the wellbore. The right plot of Figure 15 indicates that injectivity could become an issue for the 3630S polymer, especially if no fracturing of the formation is allowed.

The oil recovery curves suggest that the polymer is not severely degraded in the scenario considered here. Indeed, separate calculations in radial geometry indicate that the 3530S polymer will experience very little, if any, degradation. This number is based on analytical calculations for steady-state, single-phase flow without any inaccessible pore volume, depletion layers, or permeability reduction effects (Nødland, et al. 2019). Even if we account

for these effects, simulations in finely gridded radial geometry indicate that the conclusion still holds. On the other hand, the 3630S polymer is expected to undergo some degradation upon entering the reservoir, but no more than ~20 % in the least permeable injection zones. Since most of the injected polymer is allocated to high permeability layers, this number is likely pessimistic, especially considering that degradation levels are very low around most of the injectors, and that polymer from adjacent layers mix due to crossflow. To summarize, shear thickening and degradation is expected to influence the results to some extent for 3630S, and much less so for 3530S. Of course, this conclusion might no longer be true if the well operating constraints are changed.

The right plot of Figure 15 further highlights the fact that for non-Newtonian fluids, calculated well pressures are affected by the coarse gridding. This error, which occurs for both shear thinning and shear thickening polymers, is usually exacerbated when increasing the grid block dimensions (Li and Delshad 2014), as one would do in a realistic offshore field setting.

If we disregard near-well behavior, the polymer solutions behave as either near-Newtonian or shear thinning fluids. However, simulations of polymer flooding do not just depend on viscosity behavior; predictions are also sensitive to assumptions made about polymer adsorption and depletion layers; a form of inaccessible pore volume. In some respects these are competing effects, but there is one crucial difference: adsorption reduces the permeability to water. Potentially, this could have a large impact on reservoir sweep in the post-polymer period.

In the simulations presented so far, the maximum level of adsorption was set ~5 times lower than values matched to laboratory experiments in water-wet outcrop Berea and Bentheimer cores (Lohne, et al. 2017). Slight adjustments were also made to the depletion layer model, and the tortuosity factor was allowed to depend on porosity and water saturation via Archie's law (Archie 1942). If we instead use the originally reported input values for the adsorption and depletion layer models, the polymer is delayed at the producers, produced polymer concentrations are lower, but the predicted EOR response is greater (Figure 16)

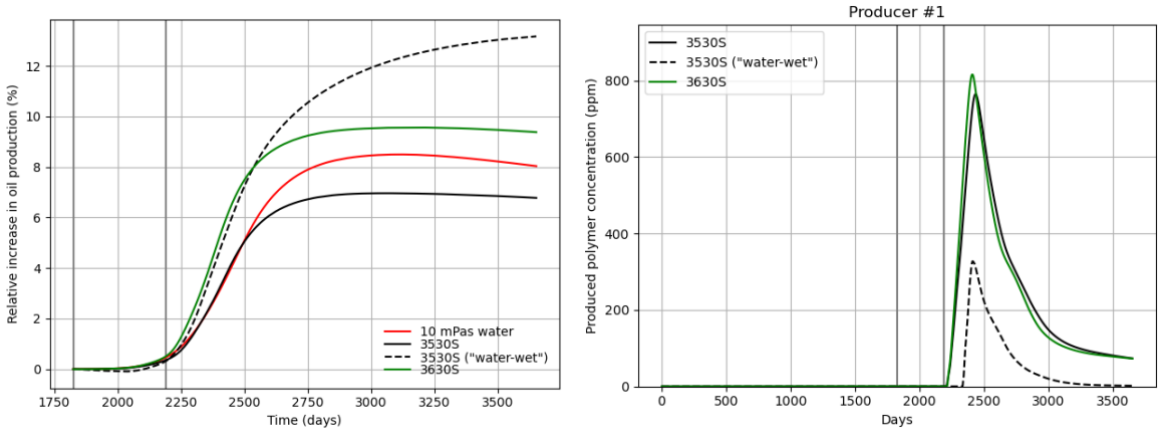

Figure 16: Left plot: Oil recovery results from Figure 7 compared with a new simulation for 3530S with higher levels of adsorption and permeability reduction, plus the original depletion layer model from (Lohne, Nødland, Stavland, & Hiorth, 2017). Right plot: Corresponding polymer production curves at one of the producers.

This is because the "water-wet" model predicts significantly higher permeability reduction than the "mixed-wet" model, and more variability inside the reservoir. The polymer front moves more slowly, but the level of mobility reduction is increased. Moreover, since oil permeability is assumed unaffected by polymer adsorption, reservoir sweep is improved in the second waterflood. Once again, higher resistance factors imply lower injectivity; for this

simulation, some of the injectors had to switch from being rate-controlled to being pressure-controlled during polymer injection (not shown).

A very high level of adsorption may not be realistic for reservoirs that are candidates for polymer flooding, but the simulation results demonstrate that permeability reduction is an important effect to consider in heterogeneous reservoirs. Here, it has a positive impact on oil recovery, but in other circumstances it is going to be detrimental.

IV. Extending polymer flooding functionality in OPM

OPM comes with an implementation of the tabled-based E100 polymer model, which is an "add-on" to the Black-Oil module. The OPM Black-Oil module solves a set of mass balance equations,

$$\frac{\partial}{\partial t}(\phi_{ref} A_\alpha) + \nabla \overrightarrow{u_\alpha} + q_\alpha = 0,$$

for pseudo-components and phases $\alpha = w, o, g$; water, oil, and gas (Rasmussen, et al. 2021). The aqueous phase is assumed to consist of only water, while the other two phases can include a mixture of oil and gas. The accumulation terms, representing the amount of mass of each component, are

$$A_w = m_\phi b_w s_{w,}$$
$$A_o = m_\phi (b_o s_o + r_{og} b_g s_g),$$
$$A_g = m_\phi (b_g s_g + r_{go} b_o s_o),$$

where m_ϕ is a porosity multiplier to account for rock compressibility, $b_\alpha = \frac{1}{B_\alpha} = \frac{V_{surface,\alpha}}{V_{reservoir,\alpha}}$ is the inverse formation volume factor of phase α, r_{og} is the ratio of vaporized oil to gas in the gaseous phase, and r_{go} is the ratio of dissolved gas in the oleic phase. The pseudo-component velocities are similarly decomposed:

$$\overrightarrow{u_w} = b_w \overrightarrow{v_w},$$
$$\overrightarrow{u_o} = b_o \overrightarrow{v_o} + r_{og} b_g \overrightarrow{v_g},$$
$$\overrightarrow{u_g} = b_g \overrightarrow{v_g} + r_{go} b_o \overrightarrow{v_o}.$$

The phase fluxes, $\overrightarrow{v_\alpha}$, are calculated from the extended Darcy law (Muskat and Meres 1936):

$$\overrightarrow{v_\alpha} = -\lambda_\alpha \boldsymbol{K}(\nabla p_\alpha - \rho_\alpha \vec{g}),$$

where $\boldsymbol{K}$ is absolute permeability, $\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}$ is the mobility of phase α, p_α is pressure, ρ_α is density, and $\vec{g}$ is the vector of gravitational acceleration.

When polymer is included, an additional equation is included for the polymer transport:

$$\frac{\partial}{\partial t}(\phi_{ref} A_p) + \nabla\left(b_w \overrightarrow{v_p} C_p\right) + q_w C_p = 0.$$

The accumulation term for polymer is

$$A_p = m_\phi b_w s_w (1 - s_{dpv}) C_p + \rho_r \widehat{C_p} \frac{1 - \phi_{ref}}{\phi_{ref}},$$

where s_{dpv} is an inaccessible (dead) pore volume factor, C_p is the polymer mass concentration, $\widehat{C_p}$ is the adsorbed concentration, and ρ_r is rock density. The polymer flux is computed from:

$$\overrightarrow{v_p} = -\frac{\lambda_p^{eff}}{R_k} \boldsymbol{K}(\nabla p_w - \rho_w \vec{g}).$$

Similarly, the water-flux is modified to

$$\overrightarrow{v_w} = -\frac{\lambda_w^{eff}}{R_k} \boldsymbol{K}(\nabla p_w - \rho_w \vec{g}).$$

Here, R_k is a permeability reduction factor, λ_w^{eff} is the water mobility accounting for effects of polymer, and λ_p^{eff} is the polymer mobility. Note that the two mobilities might differ, if using the Todd-Langstaff mixing model (Todd and Longstaff 1972).

In the source code, λ_w^{eff} is computed from the ordinary water mobility, λ_w, by a multiplication with the inverse of the polymer relative viscosity.

Transport of polymer Mw, and linking Mw to viscosity

OPM includes an experimental, and largely undocumented, feature in which a polymer molecular weight component is transported through the reservoir in the same way as the polymer mass concentration. The implemented transport equation for molecular weight is obtained by multiplying the terms in the conservation law for polymer mass by Mw:

$$\frac{\partial}{\partial t}\left(\phi_{ref} A_p M_w\right) + \nabla\left(b_w \vec{v_p} C_p M_w\right) + q_w C_p M_w = 0.$$

This approach makes it possible to compute the apparent viscosity of the polymer solution as function of local rock-fluid properties. In the current release version of OPM, a variant of the Huggins equation (Huggins 1942) is used to compute polymer apparent viscosity:

$$\frac{\eta_0}{\eta_s} = 1 + \gamma(X + \kappa X^2), \qquad X = [\eta] C_p.$$

The model computes the zero-shear viscosity, η_0, as a quadratic function of the product of polymer concentration and polymer intrinsic viscosity, the latter which is related to molecular weight by the Mark-Houwink relationship:

$$[\eta] = K \cdot M_w^a.$$

The model does not account for non-Newtonian flow; the apparent viscosity is simply set equal to the zero-shear viscosity. To improve the situation, some effort has been spent in translating the analytical shear thinning models from IORCoreSim to OPM, i.e., Meter's model (Meter and Bird 1964) and the Carreau-Yasuda model (Yasuda, Armstrong and Cohen 1981). Figure 17 shows a simulation example using the Carreau-Yasuda model; the required input parameters were the same as those fitted to a 20 MDa HPAM polymer in (Lohne, et al. 2017), but polymer adsorption and inaccessible pore volume effects were ignored.

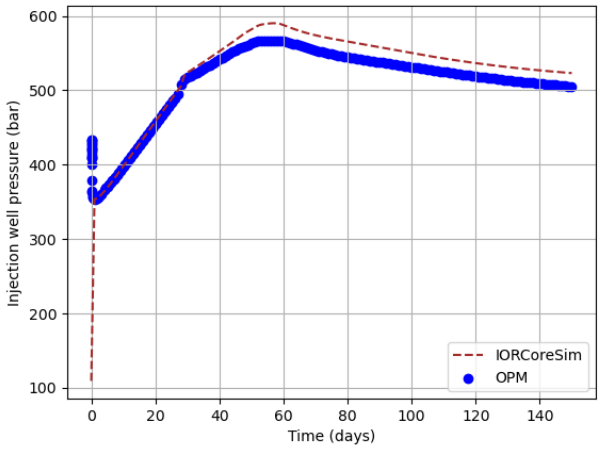

Figure 17: Simulated injection well pressure for a one-phase, 1D test case in which the Carreau-Yasuda model was used to describe shear thinning. A 1500 ppm polymer solution was injected at a constant flow rate of 220 m^3/d for 150 days. The initial water saturation was 0.2, while the oil viscosity was 7.0 mPas. Capillary pressure was ignored. The input parameters were the same as those fitted to a 20 MDa HPAM polymer in (Lohne, et al. 2017).

Note that there is some mismatch between OPM and IORCoreSim. This is mainly due to how in-situ shear rate is calculated: in OPM, the default implementation includes the mobile water saturation in the denominator for the shear rate, specifically $\dot{\gamma} \propto \frac{1}{\sqrt{S_w - S_{wr}}}$. In IORCoreSim, the total water saturation is used instead. The total water saturation can also be used in ECLIPSE 100 (E100), by setting entry 228 of the *OPTIONS* keyword to 1. Since this is not currently possible in OPM, we had to alter our original implementation of the shear-rate formula:

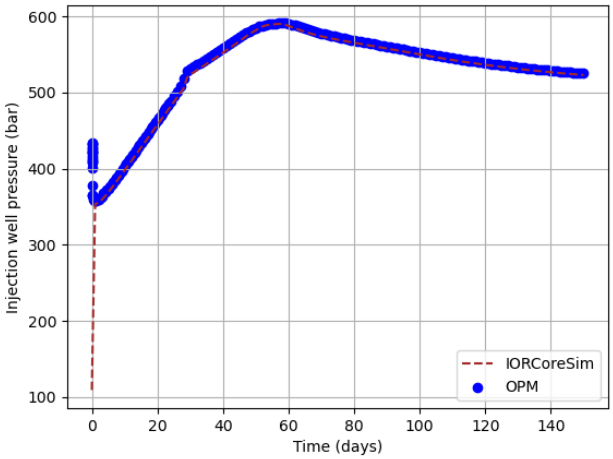

Figure 18: The same case as in Figure 17, except that the in-situ shear rate in OPM was set proportional to $\frac{1}{\sqrt{S_w}}$.

Challenges with implementing Mw-dependent shear thinning models in OPM

In the existing OPM implementation, polymer viscosity correction terms, including shear multipliers, are only computed at edges between grid cells (based on upstream weighting). In IORCoreSim, apparent viscosities are computed on a grid block-basis, and it was desired that the same should be possible in OPM. Thus, one challenge was: How to calculate representative shear rates for *each grid block*? So far, the implemented shear thinning models have only been applied on 1D cases. More work remains on how to average permeability and flow rate terms for 2D and 3D cases, which is needed when modelling anisotropic or heterogeneous formations.

We further remark that the calculation of relative permeability had to be changed. In OPM, relative permeabilities are not kept track of directly, only mobilities. Thus, whenever k_{rw} is requested, it must be computed from the current mobility:

```
Evaluation relativePermeability(unsigned phaseIdx) const
{
    // warning: slow
    return fluidState_.viscosity(phaseIdx) * mobility(phaseIdx);
}
```

This was found to be problematic: The function that updates cell-based shear rates needs to know the relative permeability of the water phase, but due to the way the simulation loop is structured, it calls *relativePermeability()* after multiplying the water mobility with polymer correction terms. In other words, the mobility used inside the above C++ function will not be k_{rw} divided by the solvent viscosity,

$$\lambda_w = \frac{k_{rw}}{\eta_s},$$

as was the intention, but instead:

$$\lambda_p = \frac{\lambda_w}{\eta_{rel} R_k} = \frac{\lambda_w}{\eta_p} \cdot \frac{\eta_s}{R_k} = \frac{k_{rw}}{\eta_p R_k}.$$

To produce the above simulation results, the *relativePermeability()*-function was altered to compute k_{rw} each time the function is called; the extent to which this slows down the simulator has not been tested.

Challenges with implementing shear-thickening and mechanical degradation

The main "conceptual" difficulty was how to account for non-Newtonian flow effects near wells, as well as mechanical degradation, e.g.:

- How to account for a spatially varying viscosity that may be both shear thinning and shear thickening? (i.e., as in the IORCoreSim well model)
    - Is it enough to add extra terms to the well pressure, and/or modify the well productivity index?
    - Or do we need to change the aqueous phase viscosity inside the reservoir too? (more complex in terms of coding)
- How to estimate mechanical degradation when using an implicit time-discretization?
- How to make the degradation solver computationally efficient?
    - Avoid slow convergence… (too many non-linear iterations)

Other challenges

There were challenges associated with "best coding practices", e.g.:

- OPM makes heavy use of modern C++, including template metaprogramming to move numerically intensive operations from runtime to compile-time. While this is done for a good reason, it makes it harder to implement new models into the code base.
- The code base is starting to get quite large, and simultaneous modifications must be made in many areas of the program, which increases the risk of introducing bugs.
    - The main difficulty was related to the well-reservoir coupling.

Finally, convergence problems were sometimes experienced when using the existing table-based polymer model, even for cases that ran perfectly fine with E100. The reason why OPM was less robust than E100 has not been discovered.

Future work/plans

The work on implementing extra polymer functionality into OPM was postponed due to other tasks (e.g., report writing). The current ambition is to continue this work.

Similarly, the work on pH developments in sandstones was discontinued, mainly due to difficulties in interpreting the data. If time allows, more work might be done in the future.

However, it is likely that attention will be shifted towards the modelling of experiments in chalk, where the mineralogy is less complicated. In particular, the plan is to look at how one can model the separation of ions and tracers with surface complexation and diffuse layer models; possibly in combination with allowing variable diffusion speeds for different species.

Dissemination of results

So far, results have been presented in project reports for the National IOR Centre of Norway.

If the work on implementing polymer models in OPM is successful, the plan is to summarize it in a journal paper. To this end, it will be necessary with much more testing, as well as discussions with other implementers of the code base.

Another goal is to publish one or more articles on Smart Water simulations.

Tables

Table 1: Reservoir and fluid properties used in Egg model case study (adapted from table 1 of Jansen et al. 2014)

Symbol	Variable	Value	Unit
h	Grid-block height	4	m
Δx, Δy	Grid-block length/width	8	m
ϕ	Porosity	0.2	-
c_o	Oil compressibility	1e-5	1/bar
c_w	Water compressibility	5e-5	1/bar
c_r	Rock compressibility	4.934e-5	1/bar
μ_w	Water (solvent) viscosity	1.07	mPas
μ_o	Oil viscosity	20	mPas
ρ_w	Water density at 200 bar	1000	kg/m^3
ρ_o	Oil density at 200 bar	900	kg/m^3
k_{rw}^o	End-point relative permeability, water	0.4	-
k_{ro}^o	End-point relative permeability, oil	1	-
n_w	Corey exponent, water	2.5	
n_o	Corey exponent, oil	3.2	
S_{or}	Residual oil saturation	0.2	
S_{wi}	Initial water saturation	0.25	-
p_c	Capillary pressure	0	bar
T	Temperature	20	°C
r_w	Well radius	0.1	m
p_i	Initial reservoir pressure	200	bar
S_{wi}	Initial water saturation	0.25	-

References

Aksulu, H., D. Håmsø, S. Strand, T. Puntervold, and T. Austad. 2012. "Evaluation of low-salinity enhanced oil recovery effects in sandstone: Effects of the temperature and pH gradient." *Energy & Fuels* 26 (6): 3497-3503.

Appelo, C. A. J., og D. Postma. 2004. *Geochemistry, groundwater and pollution.* CRC press.

Archie, G. E. 1942. «The electrical resistivity log as an aid in determining some reservoir characteristics.» *Transactions of the AIME* 146 (01): 54-62.

Austad, T., A. RezaeiDoust, and T. Puntervold. 2010. "Chemical mechanism of low salinity water flooding in sandstone reservoirs." *SPE improved oil recovery symposium.* OnePetro. doi:https://doi.org/10.2118/129767-MS.

Chauveteau, G. 1981. «Molecular interpretation of several different properties of flow of coiled polymer solutions through porous media in oil recovery conditions.» *In SPE Annual Technical Conference and Exhibition.* OnePetro.

Gaines Jr, G. L., og H. C. Thomas. 1953. «Adsorption studies on clay minerals. II. A formulation of the thermodynamics of exchange adsorption.» *The Journal of Chemical Physics* 21 (4): 714-718.

Huggins, M. L. 1942. «The viscosity of dilute solutions of long-chain molecules. IV. Dependence on concentration.» *Journal of the American Chemical Society* 64 (11): 2716-2718.

Jansen, J. D., R. M. Fonseca, S. Kahrobaei, M. M. Siraj, G. M Van Essen, og P. M. J. Van den Hof. 2014. «The Egg model – A geological ensemble for reservoir simulation.» *Geoscience Data Journal* 1 (2): 192-195.

Lantz, R. B. 1971. «Quantitative evaluation of numerical diffusion (truncation error).» *Society of Petroleum Engineers Journal, 11(03)* 315-320.

Li, Z., and M. Delshad. 2014. "Development of an analytical injectivity model for non-Newtonian polymer solutions." *SPE Journal* 19 (03): 381-389.

Li, Z., Fortenberry, Luo, H. R., and M. Delshad. 2017. "An examination of the concept of apparent skin factor in modeling injectivity of non-Newtonian polymer solutions." *Journal of Petroleum Science and Engineering* 158: 160-174.

Lohne, A., O. Nødland, A. Stavland, og A. Hiorth. 2017. «A model for non-Newtonian flow in porous media at different flow regimes.» *Computational Geosciences* 21.5: 1289–1312. doi:https://doi.org/10.1007/s10596-017-9692-6.

Meter, Donald M, and R Byron Bird. 1964. "Tube flow of non-Newtonian polymer solutions: PART I. Laminar flow and rheological models." *AIChE Journal* 10 (6): 878-881.

Muskat, M., and M. W. Meres. 1936. "The flow of heterogeneous fluids through porous media." *Physics, 7(9)* 346-363.

Nødland, O., A. Lohne, A. Stavland, og A. Hiorth. 2019. «An investigation of polymer mechanical degradation in radial well geometry.» *Transport in Porous Media* 128 (1): 1-27.

Nødland, Oddbjørn. 2021. *Project report 1.1.15.* Stavanger, Norway: The National IOR Centre of Norway.

Nødland, Oddbjørn. 2021. *Project report 1.1.15.* Stavanger, Norway: The National IOR Centre of Norway.

Peaceman, D. W. 1983. «Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability.» *Society of Petroleum Engineers Journal* 23 (03): 531-543.

Rasmussen, A. F., T. H. Sandve, K. Bao, A. Lauser, J. Hove, B. Skaflestad, og A. Thune. 2021. «The open porous media flow reservoir simulator.» *Computers & Mathematics with Applications* 81: 159-185.

RezaeiDoust, A., T. Puntervold, og T. Austad. 2011. «Chemical verification of the EOR mechanism by using low saline/smart water in sandstone.» *Energy & Fuels* 25 (5): 2151-2162.

Sorbie, K.S. 1991. *Polymer-improved oil recovery.* Springer Science & Business Media.

Stavland, A., S. M. Åsen, A. Mebratu, og F. Gathier. 2021. «Scaling of Mechanical Degradation of EOR Polymers: From Field-Scale Chokes to Capillary Tubes.» *SPE Production & Operations* 36 (01): 43-56.

Thompson, L. G., og A. C. Reynolds. 1997. «Well testing for radially heterogeneous reservoirs under single and multiphase flow conditions.» *SPE Formation Evaluation* 12 (01): 57-64.

Todd, M. R., and W. J. Longstaff. 1972. “The development, testing, and application of a numerical simulator for predicting miscible flood performance.” *Journal of Petroleum Technology, 24(07)* 874-882.

Welge, H. J. 1952. «A simplified method for computing oil recovery by gas or water drive.» *Journal of Petroleum Technology* 4 (04): 91-98.

Wieland, E., H. Wanner, Y. Albinsson, P. Wersin, og O. Karnland. 1994. *A surface chemical model of the bentonite-water interface and its implications for modelling the near field chemistry in a repository for spent fuel.* Swedish Nuclear Fuel and Waste Management Co.

Yasuda, K Y, R C Armstrong, and R E Cohen. 1981. “Shear flow properties of concentrated solutions of linear and star branched polystyrenes.” *Rheologica Acta* 20.2: 163-178.