

Simple Reservoir Model

Integrated Model - PP590

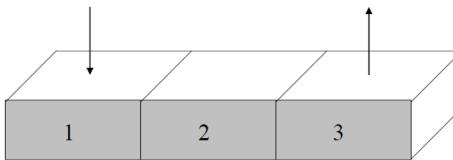
Tiago Amorim, RA: 100.675

- Actual calculations start in **Part 5**.
- Latest version in: https://github.com/TiagoCAAmorim/IntegratedModel/tree/main/docs/Report2_reservoir/SimpleReservoirModel.nb

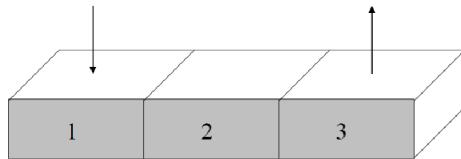
In[1]:= **ClearAll**;

Proposed Problem

- Characteristics:
 - Bi-phase, 3 blocks, horizontal
 - Water injector in block 1 (constant flow rate)
 - Producer in block 3 (well pressure is constant)
 - ϕ, μ, B_o, B_w constants; no capillary pressure
 - Primary variables: $P_{o1}, P_{o2}, P_{o3}, S_{w1}, S_{w2}, S_{w3}$
- The equations must be made available as a function of well flow rates, block dimensions (dx, dy and dz which are also constant), primary variables, block rock and fluid properties
- Make it clear which properties in the equations are functions of the block and which are treated explicitly (using previous time results)
- Assuming that the initial pressure (P_{oi}) and the initial water saturation (S_{wi}) are known, indicate how are the unknown variables



- Characteristics:
 - Bi-phase, 3 blocks, horizontal
 - Water injector in block 1 (constant flow rate)
 - Producer in block 3 (well pressure is constant)
 - ϕ, μ, B_o, B_w constants; no capillary pressure
 - Primary variables: $P_{o1}, P_{o2}, P_{o3}, S_{w1}, S_{w2}, S_{w3}$
- The equations must be made available as a function of well flow rates, block dimensions (dx, dy and dz which are also constant), primary variables, block rock and fluid properties
- Make it clear which properties in the equations are functions of the block and which are treated explicitly (using previous time results)
- Assuming that the initial pressure (P_{oi}) and the initial water saturation (S_{wi}) are known, indicate how are the unknown variables



1) Write the 6 equations as a function of:

- P (pressure)
- S_w (water saturation)
- T_o and T_w (transmissibility)
- V_i (volume from the 3 blocks)
- B_o and B_w (formation volume factor)
- Porosity (ϕ)
- Q_o and Q_w (wells flow rate)
- Use n or $n+1$ (time index representing the variables)

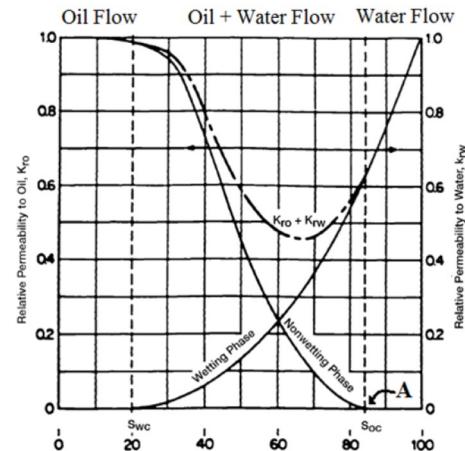


2) Specify

- Q_o and Q_w for the case where the well is operating with constant BHP

3) Forecast the reservoir for 5 years:

- $P_i = 340$ bar
- $S_{wi} = 0.2$ (relative permeability from plot)
- $K = 1$ D and $\phi = 0.3$
- $dx = dy = 200$ m and $dz = 30$ m
- $B_o = 1.01 \text{ m}^3/\text{m}^3$ and $\mu_o = 130 \text{ cP}$
- $B_w = 1 \text{ m}^3/\text{m}^3$ and $\mu_w = 1 \text{ cP}$
- $r_w = 4''$ and $S = 0$
- $Q_{wi} = 350 \text{ m}^3/\text{d}$ and $P^w = 330$ bar
- Relative permeability



Resolution

Part 1 - Formulation

Assuming black-oil formulation and that there is no free gas in the reservoir, we can simplify this problem to two equations:

$$\nabla \cdot \lambda_o (\nabla \Phi_o) = \frac{\partial}{\partial t} \left(\frac{\phi S_o}{B_o} \right) + \tilde{q}_o \quad \text{and} \quad \nabla \cdot \lambda_w (\nabla \Phi_w) = \frac{\partial}{\partial t} \left(\frac{\phi S_w}{B_w} \right) + \tilde{q}_w$$

Applying finite differences, using the implicit form for a 1D problem and multiplying by the block volume (V):

$$\sum_{i=\{i,j,k\}} [T_{p,i+1/2} (\Phi_{p,i+1} - \Phi_{p,i}) + T_{p,i-1/2} (\Phi_{p,i-1} - \Phi_{p,i})]^{n+1} = \frac{V_i}{\Delta t} \left[\left(\frac{\phi S_p}{B_p} \right)_i^{n+1} - \left(\frac{\phi S_p}{B_p} \right)_i^n \right] + q_{p,i}^{\text{well}}$$

Where:

p is either oil or water

q^{well} is in standard conditions

i is one of the main flow directions (i,j,k)

The transmissibility in the x -direction is calculated between blocks (equivalent expressions for the other directions):

$$T_{p,i+1/2} = \left(\lambda_p \frac{\Delta x \Delta y \Delta z}{\Delta x^2} \right)_{i+1/2} = \left(k \frac{k r_p}{B_p \mu_p} \frac{\Delta x \Delta y \Delta z}{\Delta x^2} \right)_{i+1/2} = \left(\frac{k A}{\Delta x} \right)_{i+1/2} \left(\frac{1}{B_p \mu_p} \right)_{i+1/2} k r_{p,i+1/2}$$

$$\left(\frac{k_i A}{\Delta x} \right)_{i+1/2} = \frac{A_{i+1/2}}{\frac{x_{i+1}-x_{i+1/2}}{k_{i,i+1}} + \frac{x_{i+1/2}-x_i}{k_{i,i}}}$$

The first part is related to the reservoir characteristics:

$$\left(\frac{k_i A}{\Delta x} \right)_{i+1/2} = \frac{A_{i+1/2}}{\frac{x_{i+1}-x_{i+1/2}}{k_{i,i+1}} + \frac{x_{i+1/2}-x_i}{k_{i,i}}}$$

Where:

$x_{i+1/2} - x_i$ is the distance in i direction between the cell center of cells i and the interface between cells $i + i+1$

$A_{i+1/2}$ is the flow area in common between cells i and $i+1$

The second part is a function of pressure:

$$\left(\frac{1}{B_p \mu_p} \right)_{i+1/2} = \frac{\left(\frac{1}{B_p \mu_p} \right)_i + \left(\frac{1}{B_p \mu_p} \right)_{i+1}}{2}$$

The third part is a function of saturation and equal to the value in the upstream block (larger pressure):

$$kr_{p,i+1/2} = kr_p, \text{"upstream"}$$

Given that for the proposed problem:

$$\Phi_{p,i} = P_{p,i} \text{ (no vertical displacement)}$$

$$P_o = P_w = P \text{ (no capillary pressure)}$$

$$S_o = 1 - S_w \text{ (no free gas)}$$

$$B_o, B_w, \mu_o, \mu_w, k, \Delta x, \Delta y, \Delta z, V \text{ are constants}$$

1-D problem, in the x direction

Flow is from the injector ($i=1$) to the producer ($i=3$)

The transmissibility becomes:

$$T_{p,i+1/2} = \frac{k \Delta y \Delta z}{\Delta x} \frac{1}{B_p \mu_p} kr_{p,i}$$

The formulation reduces to:

$$[T_{p,i+1/2} (P_{i+1} - P_i) + T_{p,i-1/2} (P_{i-1} - P_i)]^{n+1} = \frac{\Delta x \Delta y \Delta z}{\Delta t} \frac{\phi}{B_p} [S_{p,i}^{n+1} - S_{p,i}^n] + q_{p,i}^{\text{well}}$$

$$\frac{k \Delta y \Delta z}{\Delta x} \frac{1}{B_p \mu_p} [kr_{p,i} (P_{i+1} - P_i) + kr_{p,i-1} (P_{i-1} - P_i)]^{n+1} = \frac{\Delta x \Delta y \Delta z}{\Delta t} \frac{\phi}{B_p} [S_{p,i}^{n+1} - S_{p,i}^n] + q_{p,i}^{\text{well}}$$

Which can be write in as a $f(x)=0$ problem:

$$\begin{aligned} & \frac{k \Delta y \Delta z}{\Delta x} \frac{1}{B_p \mu_p} [kr_{p,i} P_{i+1}]^{n+1} - \frac{k \Delta y \Delta z}{\Delta x} \frac{1}{B_p \mu_p} [(kr_{p,i} + kr_{p,i-1}) P_i]^{n+1} + \frac{k \Delta y \Delta z}{\Delta x} \frac{1}{B_p \mu_p} [kr_{p,i-1} P_{i-1}]^{n+1} \\ & - \frac{\Delta x \Delta y \Delta z}{\Delta t} \frac{\phi}{B_p} S_{p,i}^{n+1} + \frac{\Delta x \Delta y \Delta z}{\Delta t} \frac{\phi}{B_p} S_{p,i}^n - q_{p,i}^{\text{well}} = 0 \end{aligned}$$

Where:

$$S_{p,i}^0 \text{ is known}$$

$$P_i^0 \text{ is known}$$

Problem variables:

$S_{p,i}^n$ and P_i^n , with $n = \{1, 2, \dots, n_{t\text{end}}\}$

Important to remember that:

$$kr_p = f(S_p)$$

Part 2 - Wells

For cell $i=1$ there is an injector with constant water injection rate:

$$q_{w,1}^{\text{well}} = Q_w$$

$$q_{o,1}^{\text{well}} = 0$$

Cell $i=2$ has no wells:

$$q_{w,2}^{\text{well}} = 0$$

$$q_{o,2}^{\text{well}} = 0$$

Cell $i=3$ has a producer with constant bottom-hole pressure (p_{wf}):

$$q_{w,3}^{\text{well}} = WI \frac{kr_w}{B_w \mu_w} (P_{wf} - P_3)$$

$$q_{o,3}^{\text{well}} = WI \frac{kr_o}{B_o \mu_o} (P_{wf} - P_3)$$

With $\Delta x = \Delta y$ and $k_x = k_y = k$:

$$WI = \frac{2 \pi k \Delta z}{\ln \left(\frac{r_e}{r_w} \right) + S} = \frac{2 \pi k \Delta z}{\ln \left(\frac{0.208 \Delta x}{r_w} \right)}$$

Part 3 - System of equations

Some additional constants to simplify the notation:

$$\frac{k \Delta y \Delta z}{\Delta x} \frac{1}{B_p \mu_p} = \alpha_p$$

$$\frac{\phi}{B_p} = \beta_p$$

$$WI \frac{1}{B_p \mu_p} = \tau_p$$

Thus, the main formula becomes:

$$\alpha_p [kr_{p,i} P_{i+1}]^{n+1} - \alpha_p [(kr_{p,i} + kr_{p,i-1}) P_i]^{n+1} + \alpha_p [kr_{p,i-1} P_{i-1}]^{n+1} - \beta_p S_{p,i}^{n+1} + \beta_p S_{p,i}^n - q_{p,i}^{\text{well}} = 0$$

Plugging everything together:

$$\alpha_o [kr_{o,1} P_2]^{n+1} - \alpha_o [kr_{o,1} P_1]^{n+1} + \frac{\beta_o}{\Delta t} S_{w,1}^{n+1} - \frac{\beta_o}{\Delta t} S_{w,1}^n = 0$$

$$\alpha_w [kr_{w,1} P_2]^{n+1} - \alpha_w [kr_{w,1} P_1]^{n+1} - \frac{\beta_w}{\Delta t} S_{w,1}^{n+1} + \frac{\beta_w}{\Delta t} S_{w,1}^n - Q_w = 0$$

$$\alpha_o [kr_{o,2} P_3]^{n+1} - \alpha_o [(kr_{o,2} + kr_{o,1}) P_2]^{n+1} + \alpha_o [kr_{o,1} P_1]^{n+1} + \frac{\beta_o}{\Delta t} S_{w,2}^{n+1} - \frac{\beta_o}{\Delta t} S_{w,2}^n = 0$$

$$\alpha_w [kr_{w,2} P_3]^{n+1} - \alpha_w [(kr_{w,2} + kr_{w,1}) P_2]^{n+1} + \alpha_w [kr_{w,1} P_1]^{n+1} - \frac{\beta_w}{\Delta t} S_{w,2}^{n+1} + \frac{\beta_w}{\Delta t} S_{w,2}^n = 0$$

$$-\alpha_o [kr_{o,2} P_3]^{n+1} + \alpha_o [kr_{o,2} P_2]^{n+1} + \frac{\beta_o}{\Delta t} S_{w,3}^{n+1} - \frac{\beta_o}{\Delta t} S_{w,3}^n + \tau_o [kr_{o,3}]^{n+1} P_{wf} - \tau_o [kr_{o,3} P_3]^{n+1} = 0$$

$$-\alpha_w [kr_{w,2} P_3]^{n+1} + \alpha_w [kr_{w,2} P_2]^{n+1} - \frac{\beta_w}{\Delta t} S_{w,3}^{n+1} + \frac{\beta_w}{\Delta t} S_{w,3}^n + \tau_w [kr_{w,3}]^{n+1} P_{wf} - \tau_w [kr_{w,3} P_3]^{n+1} = 0$$

Part 4 - Newton-Raphson

Since it is not possible to directly solve the system of equations, Newton-Raphson will be employed to search for the answer in each time-step.
Rewriting the problem in matrix form ($Kx = f$):

$$\begin{pmatrix}
 -\alpha_o [kr_{o,1}]^{n+1} \frac{\beta_o}{\Delta t} & \alpha_o [kr_{o,1}]^{n+1} & 0 & 0 & 0 \\
 -\alpha_w [kr_{w,1}]^{n+1} \frac{-\beta_w}{\Delta t} & \alpha_w [kr_{w,1}]^{n+1} & 0 & 0 & 0 \\
 \alpha_o [kr_{o,1}]^{n+1} & 0 & -\alpha_o [kr_{o,2} + kr_{o,1}]^{n+1} \frac{\beta_o}{\Delta t} & \alpha_o [kr_{o,2}]^{n+1} & 0 \\
 \alpha_w [kr_{w,1}]^{n+1} & 0 & -\alpha_w [kr_{w,2} + kr_{w,1}]^{n+1} \frac{-\beta_w}{\Delta t} & \alpha_w [kr_{w,2}]^{n+1} & 0 \\
 0 & 0 & \alpha_o [kr_{o,2}]^{n+1} & 0 & -\alpha_o [kr_{o,2}]^{n+1} - \tau_o [kr_{o,3}]^{n+1} \frac{\beta_o}{\Delta t} \\
 0 & 0 & \alpha_w [kr_{w,2}]^{n+1} & 0 & -\alpha_w [kr_{w,2}]^{n+1} - \tau_w [kr_{w,3}]^{n+1} \frac{-\beta_w}{\Delta t}
 \end{pmatrix} =
 \begin{pmatrix}
 \frac{\beta_o}{\Delta t} S_{w,1}^n \\
 \frac{-\beta_w}{\Delta t} S_{w,1}^n + Q_w \\
 \frac{\beta_o}{\Delta t} S_{w,2}^n \\
 \frac{-\beta_w}{\Delta t} S_{w,2}^n \\
 \frac{\beta_o}{\Delta t} S_{w,3}^n - \tau_o [kr_{o,3}]^{n+1} P_{wf} \\
 \frac{-\beta_w}{\Delta t} S_{w,3}^n - \tau_w [kr_{w,3}]^{n+1} P_{wf}
 \end{pmatrix}$$

To use Newton-Raphson it is needed the Jacobian of $R = (Kx - f)$:

$$J = K_+ =
 \begin{pmatrix}
 0 & \alpha_o (P_2^{n+1} - P_1^{n+1}) \frac{\partial kr_{o,1}}{\partial S_{w,1}} & 0 & 0 & 0 & 0 \\
 0 & \alpha_w (P_2^{n+1} - P_1^{n+1}) \frac{\partial kr_{w,1}}{\partial S_{w,1}} & 0 & 0 & 0 & 0 \\
 0 & -\alpha_o (P_2^{n+1} - P_1^{n+1}) \frac{\partial kr_{o,1}}{\partial S_{w,1}} & 0 & \alpha_o (P_3^{n+1} - P_2^{n+1}) \frac{\partial kr_{o,2}}{\partial S_{w,2}} & 0 & 0 \\
 0 & -\alpha_w (P_2^{n+1} - P_1^{n+1}) \frac{\partial kr_{w,1}}{\partial S_{w,1}} & 0 & \alpha_w (P_3^{n+1} - P_2^{n+1}) \frac{\partial kr_{w,2}}{\partial S_{w,2}} & 0 & 0 \\
 0 & 0 & 0 & -\alpha_o (P_3^{n+1} - P_2^{n+1}) \frac{\partial kr_{o,2}}{\partial S_{w,2}} & 0 & -\tau_o [P_3 - P_{wf}]^{n+1} \frac{\partial kr_{o,3}}{\partial S_{w,3}} \\
 0 & 0 & 0 & -\alpha_w (P_3^{n+1} - P_2^{n+1}) \frac{\partial kr_{w,2}}{\partial S_{w,2}} & 0 & -\tau_w [P_3 - P_{wf}]^{n+1} \frac{\partial kr_{w,3}}{\partial S_{w,3}}
 \end{pmatrix}$$

Newton-Raphson algorithm (with simplified convergence controls):

Define initial Δt

Start with the initial state (x^0)

For $i = 1, \dots, n$

Make $x_0^i = x^{i-1}$

Repeat

Calculate f , K and J

$$x_{j+1}^i = x_j^i - J^{-1}(Kx - f)$$

Until convergence : $\|Kx - f\| < \epsilon$ or $j = j_{max}$

if $\Delta_t P > \Delta_t P_{max}$ or $\Delta_t S > \Delta_t S_{max}$ then

```

 $\Delta t^{\text{new}} = \Delta t / 2$ 
Repeat time-step ( $j--$ )
else
 $\Delta t^{\text{new}} = 1.2 \Delta t$ 
 $x^i = x_{j+1}^i$ 

```

Part 5 - Relative Permeability

- Actual calculations start in the next cells.

The provided plot was manually digitalized, and the curves were approximated with known Kr correlations. The major endpoints are:

```

In[2]:= Swc = 0.10;
Swcr = 0.20;
Sor = 0.16;
Krwmax = 0.63;
Kromax = 1.00;

```

The water curve was adjusted to the Corey formulation:

```

In[7]:= nw = 2;
Swd = (Sw - Swcr) / (1 - Swcr - Sor);
krw = Piecewise[{
  {0, Sw < Swc},
  {0, Sw < Swcr},
  {Krwmax * Power[Swd, nw], Sw <= 1 - Sor},
  {Krwmax + (1 - Krwmax) * (Sw - 1 + Sor) / Sor, Sw <= 1},
  {1, Sw > 1}
}];
dkrw = D[krw, Sw];

```

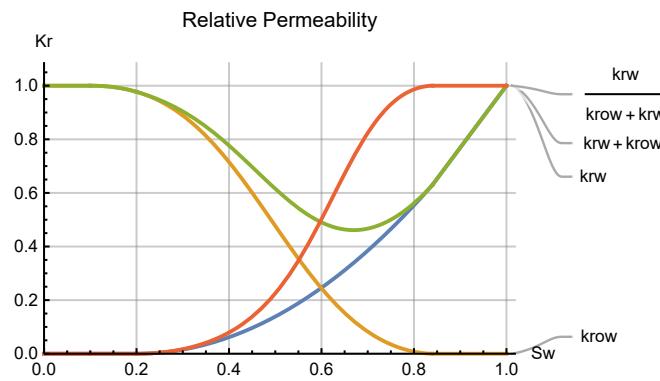
The oil curve was adjusted to the LET formulation :

```
In[11]:= Lo = 1.7;
Eo = 1.;
To = 2.;

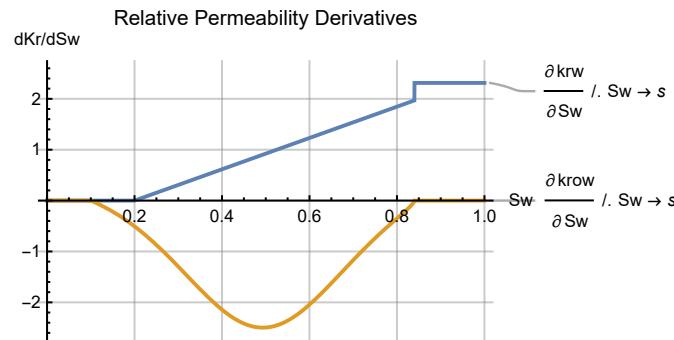
Sod = 1 - (Sw - Swc) / (1 - Swc - Sor);
krow = Piecewise[{
  {Kromax, Sw < Swc},
  {Kromax * Power[Sod, Lo] / (Power[Sod, Lo] + Eo * Power[1 - Sod, To]), Sw \leq 1 - Sor},
  {0, Sw < 1},
  {0, Sw > 1}
}];
dkrow = D[krow, Sw];

In[17]:= Plot[{krw, krow, krw + krow, krw / (krow + krw)}, {Sw, 0, 1},
  AxesLabel \rightarrow {"Sw", "Kr"},
  PlotLabel \rightarrow "Relative Permeability",
  PlotLabels \rightarrow "Expressions",
  GridLines \rightarrow Automatic,
  PlotRange \rightarrow {{0, 1}, {0, 1}}]
Plot[{D[krw, Sw] /. Sw \rightarrow s, D[krow, Sw] /. Sw \rightarrow s}, {s, 0, 1},
  AxesLabel \rightarrow {"Sw", "dKr/dSw"},
  PlotLabel \rightarrow "Relative Permeability Derivatives",
  PlotLabels \rightarrow "Expressions",
  GridLines \rightarrow Automatic,
  PlotRange \rightarrow Full]
```

Out[17]=



Out[18]=



Since S_{wi} is defined as 0.20 in the proposed problem, it will be modified in the relative permeability parameters as well.

```
In[19]:= Swc = 0.20;
Swcr = 0.20;
Sor = 0.16;
Krwmax = 0.63;
Kromax = 1.00;
nw = 2;
Swd = (Sw - Swcr) / (1 - Swcr - Sor);
krw = Piecewise[{{
    {0, Sw < Swc},
    {0, Sw < Swcr},
    {Krwmax * Power[Swd, nw], Sw \leq 1 - Sor},
    {Krwmax + (1 - Krwmax) * (Sw - 1 + Sor) / Sor, Sw \leq 1},
    {1, Sw > 1}
  }];
Lo = 1.7;
Eo = 1.;
To = 2.;
Sod = 1 - (Sw - Swc) / (1 - Swc - Sor);
krow = Piecewise[{{
    {Kromax, Sw < Swc},
    {Kromax * Power[Sod, Lo] / (Power[Sod, Lo] + Eo * Power[1 - Sod, To]), Sw \leq 1 - Sor},
    {0, Sw < 1},
    {0, Sw > 1}
  }];
dkrw = D[krw, Sw];
dkrow = D[krow, Sw];
```

Part 6 - Numerical Example

3) Forecast the reservoir for 5 years:

- $P_i = 340$ bar
- $S_{wi} = 0.2$ (relative permeability from plot)
- $K = 1$ D and $\phi = 0.3$
- $dx = dy = 200$ m and $dz = 30$ m
- $B_o = 1.01 \text{ m}^3/\text{m}^3$ and $\mu_o = 130$ cP
- $B_w = 1 \text{ m}^3/\text{m}^3$ and $\mu_w = 1$ cP
- $r_w = 4"$ and $S = 0$
- $Q_{wi} = 350 \text{ m}^3/\text{d}$ and $P^w = 330$ bar

3) Forecast the reservoir for 5 years:

- $P_i = 340$ bar
- $S_{wi} = 0.2$ (relative permeability from plot)
- $K = 1$ D and $\phi = 0.3$
- $dx = dy = 200$ m and $dz = 30$ m
- $B_o = 1.01 \text{ m}^3/\text{m}^3$ and $\mu_o = 130$ cP
- $B_w = 1 \text{ m}^3/\text{m}^3$ and $\mu_w = 1$ cP
- $r_w = 4"$ and $S = 0$
- $Q_{wi} = 350 \text{ m}^3/\text{d}$ and $P^w = 330$ bar

Problem description

```
In[34]:= Pinit = 340.; (* bar *)
Swi = 0.2;
k = 1000; (* mD *)
ϕ = 0.3;
Δx = Δy = 200.; (* m *)
Δz = 30.; (* m *)
Bo = 1.01;
μo = 130.; (* cP *)
Bw = 1.;
μw = 1.; (* cP *)
rw = 4.* $\frac{2.54}{100}$ ; (* m *)
S = 0.;
Qwi = -350.; (* m³/d *)
Pwf = 330.; (* bar *)
tEnd = 5*365.; (* days*)
```

Constants (with proper unit conversion)

$$\text{In[49]:= } \alpha_o = \frac{(k 9.869233 \times 10^{-16}) \Delta y \Delta z}{\Delta x} \frac{1}{Bo (\mu o 10^{-3})} 10^5 (24 \times 60 \times 60);$$

$$\alpha_w = \frac{(k 9.869233 \times 10^{-16}) \Delta y \Delta z}{\Delta x} \frac{1}{Bw (\mu w 10^{-3})} 10^5 (24 \times 60 \times 60);$$

$$\text{In[51]:= } \beta_o = \Delta x \Delta y \Delta z \frac{\phi}{Bo};$$

$$\beta_w = \Delta x \Delta y \Delta z \frac{\phi}{Bw};$$

$$\text{In[53]:= WI} = \frac{2\pi \left(k 9.869233 \times 10^{-16} \right) \Delta z}{\text{Log}\left[\frac{0.208 \Delta x}{r_w} \right] + S};$$

$$\text{In[54]:= } \tau_o = WI \frac{1}{Bo (\mu o 10^{-3})} 10^5 (24 \times 60 \times 60);$$

$$\tau_w = WI \frac{1}{Bw (\mu w 10^{-3})} 10^5 (24 \times 60 \times 60);$$

Some Quantities

```
In[56]:= Print["Pore volume: ", Δx Δy Δz φ, " m³"]
```

Pore volume: 360000. m³

```
In[57]:= Print["Time do inject 1% of pore volume: ", -0.01 * Δx Δy Δz φ / Qwi, " days"]
```

Time do inject 1% of pore volume: 10.2857 days

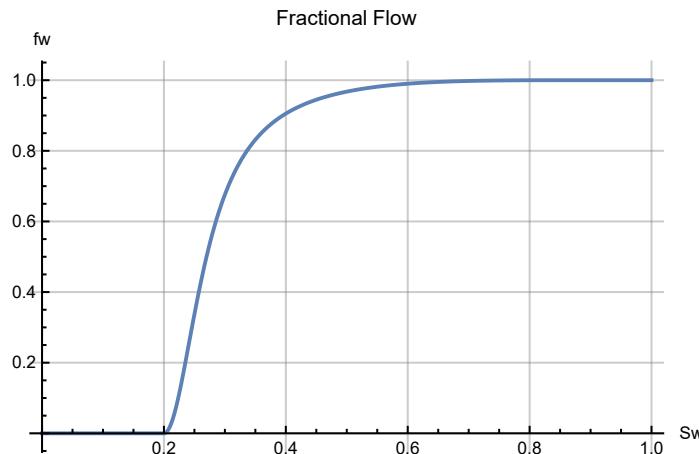
```
In[58]:= Print["Initial potential producer flow rate: ", τo (Pinit - Pwf) krow /. Sw → Swi, " m³/d"]
```

Initial potential producer flow rate: 20.3522 m³/d

```
In[59]:= fw = Piecewise[{
  {0, Sw < Swi},
  {1 / (1 + (krow / (Bo muo)) / (krw)), True}
}];

Plot[{fw /. Sw -> s}, {s, 0, 1},
 AxesLabel -> {"Sw", "fw"},
 PlotLabel -> "Fractional Flow",
 GridLines -> Automatic,
 PlotRange -> Full]
```

Out[60]=



Main functions

```
In[61]:= buildK[x_, Δt_] := Block[{Sw, kro1, kro2, kro3, krw1, krw2, krw3, K},
  kro1 = krw /. Sw → x[[2]];
  krw1 = krw /. Sw → x[[2]];
  kro2 = krw /. Sw → x[[4]];
  krw2 = krw /. Sw → x[[4]];
  kro3 = krw /. Sw → x[[6]];
  krw3 = krw /. Sw → x[[6]];
  K = {{{-αo kro1, βo/Δt, αo kro1, 0, 0, 0}}};
  K = ArrayFlatten[{{K}, {{{-αw krw1, -βw/Δt, αw krw1, 0, 0, 0}}}}];
  K = ArrayFlatten[{{K}, {{{{αo kro1, 0, -αo (kro2 + kro1), βo/Δt, αo kro2, 0}}}}}];
  K = ArrayFlatten[{{K}, {{{{αw krw1, 0, -αw (krw2 + krw1), -βw/Δt, αw krw2, 0}}}}];
  K = ArrayFlatten[{{K}, {{{{0, 0, αo kro2, 0, -αo kro2 - τo kro3, βo/Δt}}}}];
  K = ArrayFlatten[{{K}, {{{{0, 0, αw krw2, 0, -αw krw2 - τw krw3, -βw/Δt}}}}];
  K]
```

```
In[62]:= buildF[xNew_, xOld_, Δt_] := Block[{Sw, F, kro3, krw3},  
  kro3 = krow /. Sw → xNew[[6]];  
  krw3 = krw /. Sw → xNew[[6]];  
  F = {0, 0, 0, 0, 0, 0};  
  F[[1]] =  $\frac{\beta_o}{\Delta t}$  xOld[[2]];  
  F[[2]] = -  $\frac{\beta_w}{\Delta t}$  xOld[[2]] + Qwi;  
  F[[3]] =  $\frac{\beta_o}{\Delta t}$  xOld[[4]];  
  F[[4]] = -  $\frac{\beta_w}{\Delta t}$  xOld[[4]];  
  F[[5]] =  $\frac{\beta_o}{\Delta t}$  xOld[[6]] - τo kro3 Pwf;  
  F[[6]] = -  $\frac{\beta_w}{\Delta t}$  xOld[[6]] - τw krw3 Pwf;  
  F]
```

```
In[63]:= buildJ[x_, Δt_] := Block[{Sw, dkro1, dkro2, dkro3, dkrw1, dkrw2, dkrw3, J},  
dkro1 = dkrw /. Sw → x[[2]];  
dkrw1 = dkrw /. Sw → x[[2]];  
dkro2 = dkrw /. Sw → x[[4]];  
dkrw2 = dkrw /. Sw → x[[4]];  
dkro3 = dkrw /. Sw → x[[6]];  
dkrw3 = dkrw /. Sw → x[[6]];  
J = buildK[x, Δt];  
  
J[[1]][[2]] += αo (x[[3]] - x[[1]]) dkro1;  
J[[2]][[2]] += αw (x[[3]] - x[[1]]) dkrw1;  
  
J[[3]][[2]] += -αo (x[[3]] - x[[1]]) dkro1;  
J[[4]][[2]] += -αw (x[[3]] - x[[1]]) dkrw1;  
J[[3]][[4]] += αo (x[[5]] - x[[3]]) dkro2;  
J[[4]][[4]] += αw (x[[5]] - x[[3]]) dkrw2;  
  
J[[5]][[4]] += -αo (x[[5]] - x[[3]]) dkro2;  
J[[6]][[4]] += -αw (x[[5]] - x[[3]]) dkrw2;  
J[[5]][[6]] += -τo (x[[5]] - Pwf) dkro3;  
J[[6]][[6]] += -τw (x[[5]] - Pwf) dkrw3;  
J]
```

```
In[64]:= buildR[x_, xOld_, Δt_, verbose_] := Block[{K, F, R},
  K = buildK[x, Δt];
  F = buildF[x, xOld, Δt];
  R = Dot[K, x] - F;
  If[verbose,
    Print["X=", MatrixForm[x]];
    Print["K=", MatrixForm[K]];
    Print["F=", MatrixForm[F]];
    Print["R=", MatrixForm[R]];
  ];
  R]

In[65]:= solveNextX[xOld_, Δt_, verbose_] := Block[{K, F, J, R, x, dx, MaxDeltaP, MaxDeltaS},
  x = xOld;
  n = 1;
  If[False, Print["Initial value"]];
  R = buildR[x, xOld, Δt, False];
  While[n ≤ 10 && Norm[R] > 10-6,
    J = buildJ[x, Δt];
    dx = LinearSolve[J, -R];
    x += dx;
    If[False, Print["Iteration ", n]];
    R = buildR[x, xOld, Δt, False];
    n++];
  dx = x - xOld;
  MaxDeltaP = Max[Abs[dx[[1]]], Abs[dx[[3]]], Abs[dx[[5]]]];
  MaxDeltaS = Max[Abs[dx[[2]]], Abs[dx[[4]]], Abs[dx[[6]]]];
  If[verbose, Print["  iterations: ", n, "; ||R||=", Norm[R], " MaxΔP=", MaxDeltaP, " MaxΔS=", MaxDeltaS]];
  {x, dx}
]
```

```

In[66]:= checkConvergence[ $\Delta P_{max}$ ,  $\Delta S_{max}$ ,  $dx_$ ] := Block[{MaxDeltaP, MaxDeltaS, ok},
  MaxDeltaP = Max[Abs[dx[1]], Abs[dx[3]], Abs[dx[5]]];
  MaxDeltaS = Max[Abs[dx[2]], Abs[dx[4]], Abs[dx[6]]];
  ok = (MaxDeltaP <  $\Delta P_{max}$ ) && (MaxDeltaS <  $\Delta S_{max}$ );
  ok
]

In[67]:= runSim[xInit_,  $\Delta P_{max}$ ,  $\Delta S_{max}$ ,  $\Delta t_{max}$ ,  $\Delta t_{Init}$ , verbose_] := Block[{x,  $\Delta t$ , t, xLast, xNew, dx, dxNew, i},
   $\Delta t$  =  $\Delta t_{Init}$ ;
  t = {0};
  x = {xInit};
  dx = {{0, 0, 0, 0, 0, 0}};
  xLast = xInit;
  For[i = 0, t[[i + 1]] < tEnd, i++,
    If[verbose, Print["Time-step: ", i + 1, ", Time: ", t[[i + 1]] +  $\Delta t$ , ",  $\Delta t$ : ",  $\Delta t$ , " days"]];
    {xNew, dxNew} = solveNextX[xLast,  $\Delta t$ , verbose];
    (* In this incompressible system the first dP is very high, and so it will be ignored in the 1st time-step. *)
    If[checkConvergence[If[Length[t] > 1,  $\Delta P_{max}$ ,  $10^6$ ],  $\Delta S_{max}$ , dxNew],
      xLast = xNew;
      AppendTo[t, t[[i + 1]] +  $\Delta t$ ];
      AppendTo[x, xNew];
      AppendTo[dx, dxNew];
       $\Delta t$  = Min[1.2  $\Delta t$ ,  $\Delta t_{max}$ , tEnd - t[[i + 2]]];
      ,
       $\Delta t$  = 0.5  $\Delta t$ ;
      If[verbose, Print["Convergence criteria failed. New  $\Delta t$ : ",  $\Delta t$ , " days"]];
      i--;
    ];
  ];
  {t, x, dx}
]

In[68]:= postProcess[t_, x_, dx_] := Block[{i, n, P1, P2, P3, S1, S2, S3, t2, dP1, dP2, dP3, dS1, dS2, dS3, kro3, krw3, Qop, Qwp, Qlp, Qwiv, Wcut},

```

```

Print[ListLinePlot[{Transpose[{t, Join[{0}, Differences[t]]}]},
  AxesLabel -> {"Time [days]", "Δt [days]"},
  PlotLabel -> "Time-Step Size",
  GridLines -> Automatic,
  PlotRange -> Full]];

n = Length[t];
P1 = Table[x[[i]][1], {i, n}];
P2 = Table[x[[i]][3], {i, n}];
P3 = Table[x[[i]][5], {i, n}];
S1 = Table[x[[i]][2], {i, n}];
S2 = Table[x[[i]][4], {i, n}];
S3 = Table[x[[i]][6], {i, n}];

Print[ListLinePlot[{Transpose[{t, P1}], Transpose[{t, P2}], Transpose[{t, P3}]},
  AxesLabel -> {"Time [days]", "Pressure [bar]"},
  PlotLabel -> "Block Pressures",
  PlotLabels -> {"P1", "P2", "P3"},
  GridLines -> Automatic,
  PlotRange -> Full]];

Print[ListLinePlot[{Transpose[{t, S1 * 100}], Transpose[{t, S2 * 100}], Transpose[{t, S3 * 100}]},
  AxesLabel -> {"Time [days]", "Saturation [%]"},
  PlotLabel -> "Block Saturarions",
  PlotLabels -> {"S1", "S2", "S3"},
  GridLines -> Automatic,
  PlotRange -> Full]];

(* In this incompressible system the first dP is very high, an so it will be ignored in the 1st time-step. *)
t2 = Take[t, -(n - 2)];
dP1 = Table[dx[[i + 2]][1], {i, n - 2}];
dP2 = Table[dx[[i + 2]][3], {i, n - 2}];

```

```

dP3 = Table[dx[[i + 2]][5], {i, n - 2}];

dS1 = Table[dx[[i + 2]][2], {i, n - 2}];
dS2 = Table[dx[[i + 2]][4], {i, n - 2}];
dS3 = Table[dx[[i + 2]][6], {i, n - 2}];

Print[ListLinePlot[{Transpose[{t2, dP1}], Transpose[{t2, dP2}], Transpose[{t2, dP3}]},
  AxesLabel → {"Time [days]", "ΔPressure [bar]"}, 
  PlotLabel → "Block Delta Pressures",
  PlotLabels → {"ΔP1", "ΔP2", "ΔP3"}, 
  GridLines → Automatic,
  PlotRange → Full]];

Print[ListLinePlot[{Transpose[{t2, dS1 * 100}], Transpose[{t2, dS2 * 100}], Transpose[{t2, dS3 * 100}]},
  AxesLabel → {"Time [days]", "ΔSaturation [%]"}, 
  PlotLabel → "Block Delta Saturarions",
  PlotLabels → {"ΔS1", "ΔS2", "ΔS3"}, 
  GridLines → Automatic,
  PlotRange → Full]];

kro3 = Table[krow /. Sw → S3[[i]], {i, n}];
krw3 = Table[krw /. Sw → S3[[i]], {i, n}];
Qop = Table[τo kro3[[i]] (P3[[i]] - Pwf), {i, n}];
Qwp = Table[τw krw3[[i]] (P3[[i]] - Pwf), {i, n}];
Qlp = Qop + Qwp;
Qwiv = Table[-Qwi, {i, n}];
Wcut = Qwp / (Qop + Qwp);

colors = {Green, Blue, Black, Purple};
Print[ListLinePlot[{Transpose[{t, Qop}], Transpose[{t, Qwp}], Transpose[{t, Qlp}], Transpose[{t, Qwiv}]},
  AxesLabel → {"Time [days]", "Liquid Flow [m³/d]"}, 
  PlotLabel → "Well Rates",
  PlotLabels → {"Qop", "Qwp", "Qlp", "Qwi"}, 
  PlotRange → All]];

```

```
PlotStyle -> colors,
GridLines -> Automatic,
PlotRange -> Full]];

Print[ListLinePlot[{Transpose[{t, Wcut * 100}]},
AxesLabel -> {"Time [days]", "Water-cut [%]"},
PlotLabel -> "Water Cut",
GridLines -> Automatic,
PlotRange -> Full]];
]
```

Simulation

```
In[69]:= xInit = {Pinit, Swi, Pinit, Swi, Pinit, Swi};

In[70]:= {t, x, dx} = runSim[xInit, 1, 0.005, 10, 1, False];

In[71]:= postProcess[t, x, dx]
```

