On POD-based Deflation Vectors for DPCG applied to porous media problems.

G. B. Diaz Cortes[1], C. Vuik[1] and J. D. Jansen[2]

[1]Department of Applied Mathematics, TU Delft
[2]Department of Geoscience & Engineering, TU Delft

July 26, 2017

Abstract

Simulation of two-phase flow trough highly heterogeneous porous media results in large systems of linear equations for the pressure equation, when using e.g., sequential procedures. During simulation, the most time consuming part is the solution of the resulting linear system. In this work, solution of the elliptic pressure equation is studied with preconditioning and deflation techniques based on information obtained from the system.

For good performance of deflation techniques, it is necessary to find good deflation vectors. If a good selection of these vectors is made, only a a small increase in the required computing time per iteration and an important decrease in the number of iterations is achieved.

In this work, we propose the capture of a series of snapshots, solutions of the system, and the use of them as deflation vectors. These snapshots are used directly as deflation vectors and to construct a POD basis. A POD-reduced basis is also proposed as deflation vectors. We investigate convergence and the properties of the resulting methods. We perform a series of numerical experiments with diverse conditions. We consider compressible two-phase flow in a layered model with variations in the permeability layers up to 10^3 and the SPE 10 benchmark model with a contrast in permeability coefficients of 10^7. We study 2D and 3D reservoirs, taking into account gravity terms for the 3D case. We consider immiscible fluids with and without capillary pressure. We studied water flooding, with injection through the boundaries and through wells.

1 Introduction

1.1 Two phase flow through porous media

When simulating two phase flow through a porous medium, these two phases can be considered as separated, i.e., they are immiscible and there is no mass transfer between them. The contact area between the phases is known as the interface.

When modeling two phases, we usually consider one of the fluids as the wetting phase (w), which is more attracted to the mineral particles than the other phase. The other phase is considered as non-wetting phase (n). In the case of a water-oil system, water is considered the wetting phase.

The saturation of a phase (S_α), is the fraction of void space filled with that phase in the medium, where a zero saturation indicates that the phase is not present. If there are two phases present in the porous medium, these fluids fill completely the empty space, which is expressed by the following relation,

$$S_n + S_w = 1. \tag{1}$$

The surface tension and the curvature of the interface between the fluids causes a difference in pressure between the two phases. This difference is known as the capillary pressure, p_c, which, depends on the saturation:

$$p_c(S_w) = p_n - p_w. \tag{2}$$

The pressure of the non-wetting fluid is higher than the pressure of the wetting fluid, therefore, the capillary pressure is always a positive quantity. The relation between the capillary pressure and the saturation is obtained as an empirical model based on experiments. The capillary curve depends on the difference in pore-size distributions, porosity and permeability of the medium. To normalize the measured data, it is common to use a function called Leverett J-function, which takes the following form:

$$J_L(S_w) = \frac{P_c}{\sigma cos\theta}\sqrt{\frac{K}{\phi}}, \tag{3}$$

where σ is the surface tension, ϕ and K are the permeability and the porosity of the medium and θ the contact angle.

Sometimes, the capillary pressure is expressed as an analytical function of the normalized water saturation ($\hat{S}_w = \frac{S_w - S_w^{min}}{S_w^{max} - S_w^{min}}$). A model for the relationship between the capillary pressure and water saturation was proposed by Brooks and Corey (for water and air):

$$\hat{S}_w = \begin{cases} (p_c/p_e)^{-n_b} & \text{if } p_c > p_e \\ 1 & p_c \leq p_e \end{cases}$$

where p_e is the pressure of air, and n_b is related to the pore-size distribution. Another model was proposed by Genuchten:

$$\hat{S}_w = \left(1 + (\beta_g p_c)_g^n)^{-m_g}\right), \tag{4}$$

where β_g is related to the average size of pores and n_g and m_g are related to the pore size distribution.

Relative permeability

When modeling two phases, the permeability of each phase, α, will be affected by the presence of the other phase, therefore an effective permeability K_α for each phase has to be used instead of the absolute permeability K. Due to interfacial tensions, the sum of all the phase permeabilities is less than one, i.e.,

$$\sum_\alpha K^e_\alpha < K.$$

The saturation dependent relative permeability is defined as:

$$k_{r\alpha}(S_\alpha) = K^e_\alpha / K.$$

The simplest model possible that relates the relative permeabilities with the saturations is the is Corey model:

$$\begin{aligned} k_{rw} &= (\hat{S}_w)^{n_w} k^0_w, \\ k_{ro} &= (1 - \hat{S}_w)^{n_n} k^0_o. \end{aligned} \tag{5}$$

where $n_w > 1$, $n_o > 1$ and k^0_α are fitting parameters.

As in the single-phase case, the governing equations for two-phase flow in a porous medium are the mass conservation and Darcy's law. The mass balance equation for a phase α is given by:

$$\frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{v}_\alpha) = \rho_\alpha q_\alpha, \tag{6}$$

and Darcy's law reads:

$$\mathbf{v}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha} K (\nabla p_\alpha - \rho_\alpha g \nabla z). \tag{7}$$

Where, ρ_α, μ_α, q_α and p_α are the density, viscosity, sources and pressure of each phase, g is the gravity constant, and z is the depth of the reservoir.

To simplify notation, the phase mobilities ($\lambda_\alpha(S_\alpha) = K k_{r\alpha}(S_\alpha)/\mu_\alpha$) are introduced. Combining Darcy's law (7), the mass balance equation (6) and using the phase mobilities, the system reads:

$$\frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} - \nabla \cdot (\rho_\alpha \lambda_\alpha (\nabla p_\alpha - \rho_\alpha g \nabla z)) = \rho_\alpha q_\alpha. \tag{8}$$

Incompressible two-phase flow

In the case of incompressible flow, the porosity ϕ and the densities ρ_α do not depend on time. Therefore, Equation (8) reduces to:

$$\phi \frac{\partial S_\alpha}{\partial t} - \nabla \cdot (\lambda_\alpha (\nabla p_\alpha - \rho_\alpha g \nabla z)) = q_\alpha. \tag{9}$$

A common approach to solve this problem is the fractional flow formulation, where the fractional flow function is defined as:

$$f_{\alpha 1} = \frac{\lambda_{\alpha 1}}{\lambda_{\alpha 1} + \lambda_{\alpha 2}},$$

for a two-phase system with phases $\alpha 1$ and $\alpha 2$.

Fractional flow formulation

When we have a wetting (w) and a non wetting phase (n), the system of equations for an incompressible problem reads:

$$\begin{aligned}\phi\frac{\partial S_w}{\partial t}+\nabla\cdot\mathbf{v}_w=&\phi\frac{\partial S_w}{\partial t}+\nabla\cdot(\lambda_w(\nabla p_w-\rho_w g\nabla z))=q_w,\\ \phi\frac{\partial S_n}{\partial t}+\nabla\cdot\mathbf{v}_n=&\phi\frac{\partial S_n}{\partial t}+\nabla\cdot(\lambda_n(\nabla p_n-\rho_n g\nabla z))=q_n.\end{aligned}\tag{10}$$

To solve this system, we define the total Darcy's velocity as the sum of the velocity in the wetting and non wetting phases:

$$\begin{aligned}\mathbf{v}=\mathbf{v}_w+\mathbf{v}_n&=-\lambda_n\nabla p_n-\lambda_w\nabla p_w+(\lambda_n\rho_n+\lambda_w\rho_w)g\nabla z\\ &=-(\lambda_n+\lambda_w)\nabla p_n+\lambda_w\nabla p_c+(\lambda_n\rho_n+\lambda_w\rho_w)g\nabla z.\end{aligned}\tag{11}$$

If we add the two continuity equations (System (10)) and use the relationship $S_n+S_w=1$, we have:

$$\phi\frac{\partial(S_w+S_n)}{\partial t}+\nabla\cdot(\mathbf{v}_w+\mathbf{v}_n)=\nabla\cdot\mathbf{v}=q,\tag{12}$$

where $q=q_n+q_w$ is the total source. Defining the total mobility as $\lambda=\lambda_n+\lambda_w$, Equation (12) becomes:

$$-\nabla\cdot(\lambda\nabla p_n)=q-\nabla[\lambda_w\nabla p_c+(\lambda_n\rho_n+\lambda_w\rho_w)g\nabla z],\tag{13}$$

which is an equation for the pressure of the non wetting phase. This equation depends on the saturation via the capillarity p_c and the total mobility λ.

Multiplying each phase velocity by the relative mobility of the other phase and subtracting the result, together with Equation (11), we get:

$$\begin{aligned}\lambda_n\mathbf{v}_w-\lambda_w\mathbf{v}_n&=\lambda\mathbf{v}_w-\lambda_w\mathbf{v}\\ &=\lambda_w\lambda_n[\nabla p_c+(\rho_w-\rho_n)g\nabla z].\end{aligned}$$

Therefore, for $\mathbf{v}_w$ we have

$$\mathbf{v}_w=\frac{\lambda_w}{\lambda}\mathbf{v}+\frac{\lambda_w\lambda_n}{\lambda}[\nabla p_c+(\rho_w-\rho_n)g\nabla z].$$

Using the velocity computed above, and the fractional flow function, $f_w=\frac{\lambda_w}{\lambda_w+\lambda_o}$, the transport Equation (6) for the wetting phase reads:

$$\phi\frac{\partial S_w}{\partial t}+\nabla\cdot[f_w(\mathbf{v}+\lambda_n\Delta\rho g\nabla z)]+\nabla\cdot(f_w\lambda_n\nabla p_c)=q_w,\tag{14}$$

where $\Delta\rho=\rho_w-\rho_n$. With this approach, the system is expressed in terms of the pressure of the non wetting phase, Equation (13), and the saturation of the wetting phase, Equation (14). In the pressure equation, the coupling to saturation is present via the phase mobilities and the derivative of the capillary function. For the saturation, we have an indirect coupling with the pressure through the total Darcy velocity.

1.2 Numerical methods

Sequential solution procedures

To solve the system described by Equation (13) and Equation (14), various procedures can be used. For this work, we use the sequential procedure, where, the equations are solved separately in consecutive substeps. An unknown is fixed, e.g., the saturation, and the elliptic equation is solved for the pressure. Once the pressure is computed, it is used to compute the total velocity and to solve the parabolic transport equation. If the capillary pressure is zero, the transport equation becomes hyperbolic.

The sequential solution procedure is sometimes referred as IMPES, implicit pressure, explicit saturation.

Temporal discretization

The saturation equation is time dependent. The time discretization can be performed using two schemes: implicit and explicit. In the explicit scheme, the time derivative is approximated using the solutions obtained for the previous time step, the system to solve, reads:

$$\phi\frac{(S_w^{n+1}-S_w^n)}{\Delta t}+\nabla\cdot[f_w(S_w^n)(\mathbf{v}^n+\lambda_n\Delta\rho g\nabla z)]+\nabla\cdot(f_w(S_w^n)\lambda_n(S_w^n)\nabla p_c(S_w^n))=q_w^{n+1}. \quad (15)$$

For the implicit solution, backward Euler discretization schemes can be used. With this scheme, Equation (14) is:

$$\phi\frac{(S_w^{n+1}-S_w^n)}{\Delta t}+\nabla\cdot[f_w(S_w^{n+1})(\mathbf{v}^n+\lambda_n\Delta\rho g\nabla z)]+\nabla\cdot(f_w(S_w^{n+1})\lambda_n(S_w^{n+1})\nabla p_c(S_w^{n+1}))=q_w^n. \quad (16)$$

That can be rewritten as:

$$\begin{aligned}S_w^{n+1}-S_w^n-\frac{\Delta t}{\phi}\left(q_w-\nabla\cdot[f_w(S_w^{n+1})(\mathbf{v}^n+\lambda_n\Delta\rho g\nabla z)]\right)&\\+\frac{\Delta t}{\phi}\left(\nabla\cdot(f_w(S_w^{n+1})\lambda_n(S_w^{n+1})\nabla p_c(S_w^{n+1}))\right)&=0.\end{aligned} \quad (17)$$

If we use the implicit scheme, the resulting system is nonlinear (Equation (17)) and depends on the saturation at time step n and $n+1$. The nonlinear system can be solved using Newton-Raphson (NR) method. With this method, for the $(k+1)$-th iteration we have:

$$J(S^k)\delta S^{k+1}=-F(S^k,S^n), \qquad S^{k+1}=S^k+\delta S^{k+1},$$

where

$$\begin{aligned}F(S^k,S^n)=S_w^k-S_w^n-\frac{\Delta t}{\phi}\left(q_w-\nabla\cdot[f_w(S_w^k)(\mathbf{v}^n+\lambda_n\Delta\rho g\nabla z)]\right)&\\+\frac{\Delta t}{\phi}\left(\nabla\cdot(f_w(S_w^k)\lambda_n(S_w^k)\nabla p_c(S_w^k))\right),&\end{aligned} \quad (18)$$

$J(S^k) = \frac{\partial F(S^k, S^n)}{\partial S^k}$ is the Jacobian matrix, and δS^{k+1} is the NR update at iteration step $k+1$. Therefore, the linear system to solve is:

$$J(S^k)\delta S^{k+1} = b(S^k). \tag{19}$$

where $b(S^k)$ is the function evaluated at iteration step k, $b(S^k) = -F(S^k, S^n)$.

Spatial discretization

As mentioned before, for the sequential procedure we need to solve the equation for pressure and later solve the transport equation. For the pressure we have to solve Equation (13). The spatial derivatives are approximated using a finite difference scheme with cell central differences. For a 3D model, taking a mesh with a uniform grid size Δx, Δy, Δz where (i, j, l) is the center of the cell in the position i in the x direction, j in the y direction, and l in the z direction (x_i, y_j, z_l), where $p_{i,j,l} = p(x_i, y_j, z_l)$ is the pressure at this point.
For the x direction, we have (see, e.g., [1–3]):

$$\begin{aligned} \nabla \cdot (\lambda \nabla p) &= \frac{\partial}{\partial x}\left(\lambda \frac{\partial p}{\partial x}\right) = \frac{\Delta}{\Delta x}\left(\lambda \frac{\Delta p}{\Delta x}\right) + \mathscr{O}(\Delta x^2) \\ &= \frac{\lambda_{i+\frac{1}{2},j,l}(p_{i+1,j,l} - p_{i,j,l}) - \lambda_{i-\frac{1}{2},j,l}(p_{i,j,l} - p_{i-1,j,l})}{(\Delta x)^2} + \mathscr{O}(\Delta x^2), \end{aligned}$$

where $\lambda_{i-\frac{1}{2},j,l}$ is the harmonic average of the mobility for cells $(i-1, j, l)$ and (i, j, l):

$$\lambda_{i-\frac{1}{2},j,l} = \frac{2}{\frac{1}{\lambda_{i-1,j,l}} + \frac{1}{\lambda_{i,j,l}}}. \tag{20}$$

After discretization, not taking into account capillary pressure and gravity, and defining the *transmissibility* $(T_{i-\frac{1}{2},j,l})$ between grid cells $(i-1, j, l)$ and (i, j, l) as:

$$T_{i-\frac{1}{2},j,l} = \frac{2\Delta y \Delta z}{\mu \Delta x}\lambda_{i-\frac{1}{2},j,l}, \tag{21}$$

we can rewrite Equation (13), together with boundary conditions, as:

$$\mathbf{T}\mathbf{p} = \mathbf{q}, \tag{22}$$

where $\mathbf{T}$ is known as the transmissibility matrix, that is a sparse matrix with 3 non zero diagonals in 1D, 5 in 2D and 7 in 3D. System (22) is a linear system that can be solved with iterative or direct methods. For the solution of this system, it is necessary to define boundary conditions at all boundaries of the domain. These conditions can be prescribed pressures (Dirichlet conditions), flow rates (Neumann conditions) or a combination of these (Robin conditions).

Well model

In reservoir simulation, a fluid is typically injected or extracted through wells. During injection and production, the rate of injection or the bottom hole pressure (bhp) of the well is prescribed.

To model the injection or production through the wells, a linear relationship between the bhp and the flow rate in a well can be used. Taking the cell (i, j, l) that contains a well, the relationship between the pressure inside the well and the pressure of the cell is given by:

$$q_{(i,j,l)} = I_{(i,j,l)}(p_{(i,j,l)} - p_{bh(i,j,l)}), \tag{23}$$

where $I_{(i,j,l)}$ is the productivity or injectivity index of the well, $p_{(i,j,l)}$ is the reservoir pressure in the cell containing the well, and $p_{bh(i,j,l)}$ is a prescribed pressure inside the well.

Incompressible fluid

Using the well model in Equation (22) we obtain:

$$\mathbf{T}\mathbf{p} = \mathbf{I}_w(\mathbf{p} - \mathbf{p}_{bh}), \tag{24}$$

where $\mathbf{I}_w$ is a diagonal matrix containing the productivity or injectivity indices of the wells present in the reservoir. The diagonal elements are zero for cells without wells and have the value of the well index for each cell containing a well.

1.3 Iterative solution methods

When simulating two phase flow through a porous medium, after discretization, using the sequential procedure, we obtain a linear system for the pressure. If the system is large, solving it is time consuming, therefore iterative techniques are prefered over direct methods to solve it. The resulting matrix is SPD (Symmetric Possitive Definite), therefore we choose Conjugate Gradient (CG) as iterative method to solve it. We accelerate the solution with the Incomplete Cholesky preconditioner. In this work, we study a further acceleration with deflation and POD techniques. In this section, we give a brief overview of the methods.

Conjugate Gradient Method

The CG method is a Krylov-subspace method used when the matrix of the linear system is SPD. This method is based in the mininization of the residual in the A-norm.
Given a starting solution $\mathbf{x}^0$ and the residual defined by $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$, we define the Krylov subspace as $\mathcal{K}_k(\mathbf{A}, \mathbf{r}^0) = span\{\mathbf{r}^0, \mathbf{A}\mathbf{r}^0, \ldots, \mathbf{A}^{k-1}\mathbf{r}^0\}$. For this space, $\mathbf{x}^k \in \mathbf{x}^0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}^0)$ has a minimal error measured in the $\mathbf{A}$-norm for all approximations contained in $\mathbf{x}^0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}^0)$. The error of this approximation is bounded by:

$$||\mathbf{x} - \mathbf{x}^{k+1}||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^0||_{\mathbf{A}} \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^{k+1} \text{[1]}. \tag{25}$$

Hence, the convergence of the method depends on the condition number (κ_2) of the matrix. The pseudo code for CG is given in Algorithm 2.

Algorithm 2 Conjugate Gradient (CG) method, solving $\mathbf{A}\mathbf{x} = \mathbf{b}$.

```
Give an initial guess $\mathbf{x}^0$.
Compute $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$ and set $\mathbf{p}^0 = \mathbf{r}^0$.
  for $k = 0, ...,$ until convergence
      $\alpha^k = \frac{(\mathbf{r}^k, \mathbf{r}^k)}{(\mathbf{A}\mathbf{p}^k, \mathbf{p}^k)}$
      $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k$
      $\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha^k \mathbf{A}\mathbf{p}^k$
      $\beta^k = \frac{(\mathbf{r}^{k+1}, \mathbf{r}^{k+1})}{(\mathbf{r}^k, \mathbf{r}^k)}$
      $\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta^k \mathbf{p}^k$
  end
```

Preconditioning

To accelerate the convergence of a Krylov method, the system can be transformed into another one containing an iteration matrix with a better spectrum, i.e, a smaller condition

[1] The condition number $\kappa_2(\mathbf{A})$ is defined as $\kappa_2(\mathbf{A}) = \frac{\sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})}}{\sqrt{\lambda_{min}(\mathbf{A}^T\mathbf{A})}}$. If $\mathbf{A}$ is SPD, $\kappa_2(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})}$.

number. This can be done by multiplying the linear system by a matrix $\mathbf{M}^{-1}$.

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}. \tag{26}$$

The new system has the same solution but can provide a substantial reduction of the condition number. For this preconditioned system, the error is bounded by:

$$||\mathbf{x} - \mathbf{x}^k||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^0||_{\mathbf{A}} \left(\frac{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} + 1} \right)^k. \tag{27}$$

For the CG method, the matrix $\mathbf{M}$ is chosen as an SPD matrix such that $\kappa(\mathbf{M}^{-1}\mathbf{A}) \leq \kappa(\mathbf{A})$, and $\mathbf{M}^{-1}\mathbf{b}$ is cheap to compute.

Deflation

The deflation method is used to annihilate the effect of extreme eigenvalues hampering the convergence of an iterative method [4]. Given an SPD matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, for a given matrix $\mathbf{Z} \in \mathbb{R}^{n\times m}$ the deflation matrix $\mathbf{P}$ is defined as follows ([5, 6]):

$$\mathbf{P} = \mathbf{I} - \mathbf{A}\mathbf{Q}, \qquad \mathbf{P} \in \mathbb{R}^{n\times n}, \qquad \mathbf{Q} \in \mathbb{R}^{n\times n},$$

where

$$\mathbf{Q} = \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T, \qquad \mathbf{Z} \in \mathbb{R}^{n\times m}, \qquad \mathbf{E} \in \mathbb{R}^{m\times m},$$

with

$$\mathbf{E} = \mathbf{Z}^T\mathbf{A}\mathbf{Z}.$$

The matrix $\mathbf{E}$ is known as the *Galerkin* or *coarse* matrix. This matrix has to be invertible, to satisfy this condition $\mathbf{Z}$ has to be a full rank matrix. The full rank matrix $\mathbf{Z}$ is called the $deflation - subspace$ matrix, and it's columns are the $deflation$ vectors or *projection* vectors.

Deflated PCG Method

In this work, the solution of the linear system (24) is performed with the deflated CG method. With this method, insted of solving the original system, we solve the deflated system (see Appendix D):

$$\mathbf{P}\mathbf{A}\hat{\mathbf{x}} = \mathbf{P}\mathbf{b}, \tag{28}$$

for the deflated solution $\hat{\mathbf{x}}$. This deflated the solution is related to the solution $\mathbf{x}$ of the original system as (see Appendix D):

$$\mathbf{x} = \mathbf{Q}\mathbf{b} + \mathbf{P}^T\hat{\mathbf{x}}. \tag{29}$$

The deflated linear system can also be preconditioned by an SPD matrix $\mathbf{M}$. After preconditioning, the deflated preconditioned system to solve with CG is [6]:

$$\tilde{\mathbf{P}}\tilde{\mathbf{A}}\hat{\tilde{\mathbf{x}}} = \tilde{\mathbf{P}}\tilde{\mathbf{b}},$$

where:

$$\tilde{\mathbf{A}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{M}^{-\frac{1}{2}}, \qquad \hat{\hat{\mathbf{x}}} = \mathbf{M}^{\frac{1}{2}}\hat{\mathbf{x}}, \qquad \tilde{\mathbf{b}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{b}$$

This method is called the Deflated Preconditioned Conjugate Gradient $DPCG$ method. In practice $\mathbf{M}^{-1}\mathbf{PAx} = \mathbf{M}^{-1}\mathbf{Pb}$ is computed and the error is bounded by:

$$||\mathbf{x} - \mathbf{x}^{i+1}||_{\mathbf{A}} \leq 2||\mathbf{x} - \mathbf{x}^{0}||_{\mathbf{A}} \left(\frac{\sqrt{\kappa_{eff}(\mathbf{M}^{-1}\mathbf{PA})} - 1}{\sqrt{\kappa_{eff}(\mathbf{M}^{-1}\mathbf{PA})} + 1} \right)^{i+1},$$

were $\kappa_{eff} = \frac{\lambda_{max}(M^{-1}PA)}{\lambda_{min}(M^{-1}PA)}$ is the effective condition number and $\lambda_{min}(M^{-1}PA)$ is the smallest non-zero eigenvalue of $M^{-1}PA$.

Choices of Deflation Vectors

The deflation method is used to remove the effect of the most unfavorable eigenvalues of $\mathbf{A}$. If the matrix $\mathbf{Z}$ contains eigenvectors corresponding to the unfavorable eigenvalues, the convergence of the iterative method is achieved faster. However, to obtain and to apply the eigenvectors is costly in view of memory and CPU time. Therefore, a good choice of the matrix $\mathbf{Z}$ that efficiently approximates the eigenvectors is essential for the applicability of the method.

A good choice of the deflation vectors is usually problem-dependent. Available information on the system is, in general, used to obtain these vectors. Most of the techniques used to choose deflation vectors are based on approximating eigenvectors, recycling [7], subdomain deflation vectors [8] or multigrid and multilevel based deflation techniques [6, 9]. A summary of these techniques is given below.

Recycling Deflation. A set of search vectors previously used is reused to build the deflation-subspace matrix [7]. The vectors could be, for example, $q - 1$ solution vectors of the linear system with different right-hand sides or of different time steps. The matrix $\mathbf{Z}$ containing these solutions is:

$$\mathbf{Z} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(q-1)}].$$

Subdomain Deflation. The domain is divided into several subdomains, using domain decomposition techniques or taking into account the properties of the problem. For each subdomain, there is a deflation vector that contains ones for cells in the subdomain and zeros for cells outside [8].

Multi Grid and Multilevel Deflation. For the multigrid and multilevel methods, the prolongation and restriction matrices are used to pass from one level or grid to another. These matrices can be used as the deflation-subspace matrices $\mathbf{Z}$ [6].

1.4 Proper Orthogonal Decomposition (POD)

As mentioned before, in this work we want to combine deflation techniques and Proper Orthogonal Decomposition (POD) to reduce the number of iterations necessary to solve the linear system obtained from reservoir simulation in a cheap and automatic way. In this section, we give a brief overview of the POD method.

The POD method is a Model Order Reduction (MOR) method, where a high-order model is projected onto a space spanned by a small set of orthonormal basis vectors. The high dimensional variable $\mathbf{x} \in \mathbb{R}^n$ is approximated by a linear combination of l orthonormal basis vectors [10]:

$$\mathbf{x} \approx \sum_{i=1}^{l} c_i \psi_i, \tag{30}$$

where $\psi_i \in \mathbb{R}^n$ are the basis vectors and c_i are their corresponding coefficients. In matrix notation, equation (30) is rewritten as :

$$\mathbf{x} \approx \Psi \mathbf{c},$$

where $\Psi = [\psi_1 \ \psi_2 \ .. \ \psi_l]$, $\Psi \in \mathbb{R}^{n \times l}$ is the matrix containing the basis vectors, and $\mathbf{c} \in \mathbb{R}^l$ is the vector containing the coefficients of the basis vectors.

The basis vectors ψ_i are computed from a set of 'snapshots' $\{\mathbf{x_i}\}_{i=1,..,m}$, obtained by simulation or experiments [11]. In POD, the basis vectors $\{\psi_j\}_{j=1}^{l}$, are l eigenvectors corresponding to the largest eigenvalues $\{\sigma_j\}_{j=1}^{l}$ of the data snapshot correlation matrix $\mathbf{R}$.

$$\mathbf{R} := \frac{1}{m}\mathbf{X}\mathbf{X}^T \equiv \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^T, \qquad \mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m], \tag{31}$$

where $\mathbf{X} \in \mathbb{R}^{n \times m}$ is an SPSD matrix containing the previously obtained snapshots. The l eigenvectors should contain almost all the variability of the snapshots. Usually, they are chosen as the eigenvectors of the maximal number (l) of eigenvalues satisfying [11]:

$$\frac{\sum_{j=1}^{l} \sigma_j}{\sum_{j=1}^{m} \sigma_j} \leq \alpha, \qquad 0 < \alpha \leq 1, \tag{32}$$

with α close to 1. The eigenvalues σ_j are ordered from large to small with σ_1 the largest eigenvalue of $\mathbf{R}$. It is not necessary to compute the eigenvalues from $\mathbf{X}\mathbf{X}^T$, but instead, it is possible to compute the eigenvalues of the much smaller matrix $\mathbf{X}^T\mathbf{X}$ (see Appendix C). In this study, we normalize the snapshots, so $||\mathbf{x}_i||_2 = 1$.

2 Numerical experiments

2.1 Model problems

We model water flooding into a reservoir initially filled with oil. Therefore, the initial saturation is set as one for the oil and zero for the water (see Table 6). The solution of the systems of linear equations, resulting from the discretization of the partial differential equations describing this process, is studied in this work. Sequential solution procedures are used in this work. The transport equation is solved with the Matlab Reservoir Simulation Toolbox (MRST) [12] using implicit schemes. In this work, we focus on the solution of the pressure equation. We propose the use of the Deflated Conjugate Gradient method preconditioned with Incomplete Cholesky (DICCG) method together with POD to solve this problem. We investigate the use of snapshots and snapshots-based POD vectors as deflation vectors for DICCG.

Pressure solver

The solution of the pressure equation is approximated with the ICCG and DICCG methods. For the DICCG method, it is necessary to select a set of deflation vectors. Snapshots are taken during the first 10 time steps using the ICCG method. Once the snapshots are obtained, they are used as deflation vectors for the DICCG$_{10}$ (snapshots based deflation vectors). These snapshots are also used to compute a POD basis. For the first set of experiments, 10 POD basis vectors are used (DICCG$_{POD_{10}}$), whereas 5 POD basis vectors (DICCG$_{POD_5}$) are used for the second set of experiments (snapshots-based POD deflation vectors). The matrices corresponding to the linear systems $\mathbf{A}$ and right-hand sides $\mathbf{b}$ are obtained with MRST [12].

As tolerance or stopping criterium, we use the relative residual, defined as the 2-norm of the residual of the k^{th} iteration divided by the 2-norm of the right-hand side of the preconditioned system:

$$\frac{||\mathbf{M}^{-1}r^k||_2}{||\mathbf{M}^{-1}b||_2} \leq \epsilon.$$

The stopping criterium of the linear solvers is $\epsilon = 10^{-7}$.

In the present section, we give a general overview of the experiments that we perform, but the specifications are presented below for each problem separately.

Heterogeneous permeability layers

The experiments simulate flow through a porous medium with a constant porosity field of 0.2. For the first set of experiments, we study a layered system (see Figure 1). We use 8 layers of the same size, 4 layers with one value of permeability σ_1, followed by a layer with a different permeability value σ_2. The permeability of one set of layers is set to $\sigma_1 = 1mD$, the permeability of the other set σ_2 is varied. Therefore, the contrast in permeability between the layers ($\frac{\sigma_2}{\sigma_1} = \sigma_2$), depends on the value of σ_2. The permeability σ_2 varies from $\sigma_2 = 1mD$ to $\sigma_2 = 10^{-3}mD$. The domain consists of a Cartesian grid of 64 x 64 cells with a length of one meter each cell. For the relative permeability of the fluids, the Corey model is used. The properties of the fluids are presented in Table 1. No gravity terms and no capillary pressure are taken into account in the first set of experiments. For the second set, capillary pressure is taken into account. The capillary relationship is linear, $P_c = C(1 - S)$. The curve is presented in Figure 3. Finally, we perform 3D experiments with 64 x 64 x 10 grid cells and gravity terms included.

Property	Water	Oil	Units
μ	1	10	cp
ρ	1000	700	kg/m^3
k_r	$(S_w)^2$	$(1 - S_w)^2$	
C_p	$10 * (1 - S)$		bars

Table 1: Fluids properties.

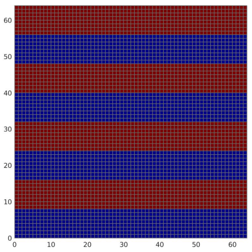

Figure 1: Rock permeability

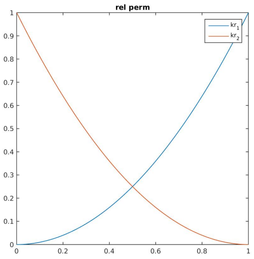

Figure 2: Fluid relative permeability

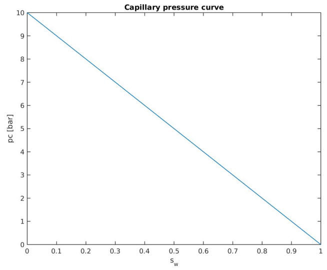

Figure 3: Capillary pressure

Injection through the boundary

For the first set of experiments, water is injected from one boundary at a rate of 0.4 m^3/day and pressure is set as zero at the right boundary and 100 bars inside the reservoir. In Case 1 the water is injected from the left boundary, and in Case 2, water is injected from the bottom boundary. The simulation is run during 2400 days with 120 time steps of 40 days (See Table 7).

Case 1: Injection through the left boundary, no capillary pressure.

In Table 4, the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selection of deflation vectors. The number of iterations necessary to achieve convergence for the ICCG method is presented in the second column (Total ICCG). The number of iterations necessary to compute the first 10 snapshots with the ICCG method are presented in the 4th column (ICCG Snapshots). In the 5th column, we present the total number of iterations taking into account the first 10 snapshots computed with ICCG and the rest of the iterations computed with DICCG. In the last column, the percentage of the total number

Case 1		
	Water	Oil
$S_{0,x\neq 0,L_x}$	0	1
$S_{x=0}$	1	0
$S_{x=L_x}$	0	1
Case 2		
	Water	Oil
$S_{0,y\neq 0,L_y}$	0	1
$S_{y=0}$	1	0
$S_{y=L_y}$	0	1

Table 2: Saturations.

Property	Value	Units
T_{total}	2400	days
T_{steps}	120	
dT	40	days
Case 1		
$Q_{x=0}$	0.4	m^3/day
$P_{0,x\neq 0,L_x}$	100	$bars$
$P_{x=L_x}$	0	$bars$
Case 2		
$Q_{y=0}$	0.4	m^3/day
$P_{0,y\neq 0,L_y}$	100	$bars$
$P_{y=L_y}$	0	$bars$

Table 3: Initial values of the system.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^0	9918	DICCG_{10}	798	872	1670	17
10^0	9918	$\text{DICCG}_{POD_{10}}$	798	872	1670	17
10^0	9918	DICCG_{POD_5}	798	710	1508	15
10^1	12115	DICCG_{10}	929	821	1750	14
10^1	12115	$\text{DICCG}_{POD_{10}}$	929	821	1750	14
10^1	12115	DICCG_{POD_5}	929	952	1881	16
10^2	11633	DICCG_{10}	920	839	1759	15
10^2	11633	$\text{DICCG}_{POD_{10}}$	920	839	1759	15
10^2	11633	DICCG_{POD_5}	920	961	1881	16
10^3	8971	DICCG_{10}	692	699	1391	16
10^3	8971	$\text{DICCG}_{POD_{10}}$	692	699	1391	16
10^3	8971	DICCG_{POD_5}	692	740	1432	16

Table 4: Comparison between the ICCC and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the left boundary, domain 64 x 64 cells.

of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the water saturation are presented in Figures 4 for various times.

We observe in Table 4 that when we use deflation methods, the number of linear iterations is reduced up to $\approx 17\%$ of the total number of iterations computed with ICCG. We also note that the number of iterations necessary to compute the 10 snapshots is similar to the number of iterations necessary to compute the rest of the timesteps with DICCG (110). In this table, we also observe that the number of iterations does not change when we vary the contrast between permeability layers. Finally, we note that the results are similar for the three deflation approaches, 10 snapshots and 10 and 5 POD basis vectors as deflation vectors.

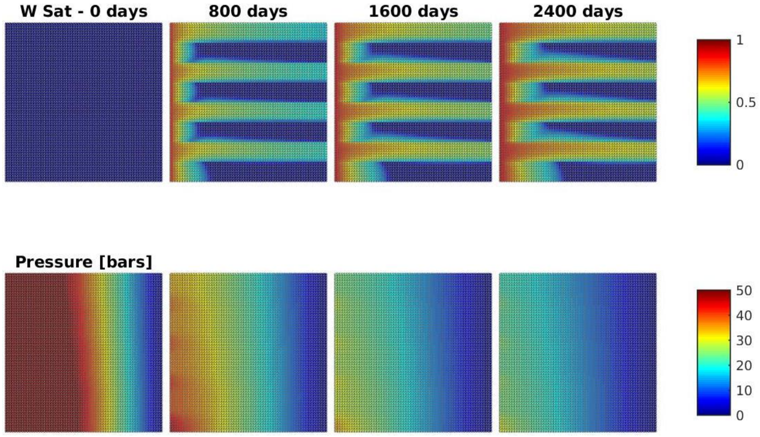

Figure 4: Pressure field for various times for a contrast between permeability values of 10^1, 64 x 64 grid cells.

Case 1 A: Injection through the left boundary, capillary pressure included.

In Table 5, the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selection of deflation vectors. The pressure field and the water saturation are presented in Figure 5 for various times. In Table 5, we observe that the behaviour of the deflation solver is similar to the case when we do not take capillary pressure into account. The percentage of ICCG iterations is slightly larger than the previous case, but it does not depend on the contrast between permeability layers.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^0	10059	DICCG$_{10}$	806	1053	1859	18
10^0	10059	DICCG$_{POD_{10}}$	806	1053	1859	18
10^0	10059	DICCG$_{POD_5}$	806	1022	1828	18
10^1	12774	DICCG$_{10}$	916	1105	2021	16
10^1	12774	DICCG$_{POD_{10}}$	916	1105	2021	16
10^1	12774	DICCG$_{POD_5}$	916	1214	2130	17
10^2	11115	DICCG$_{10}$	890	986	1876	17
10^2	11115	DICCG$_{POD_{10}}$	890	986	1876	17
10^2	11115	DICCG$_{POD_5}$	890	1100	1990	18
10^3	8633	DICCG$_{10}$	680	887	1567	18
10^3	8633	DICCG$_{POD_{10}}$	680	887	1567	18
10^3	8633	DICCG$_{POD_5}$	680	908	1588	18

Table 5: Comparison between the ICCC and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the left boundary, domain 64 x 64 cells, capillary pressure included.

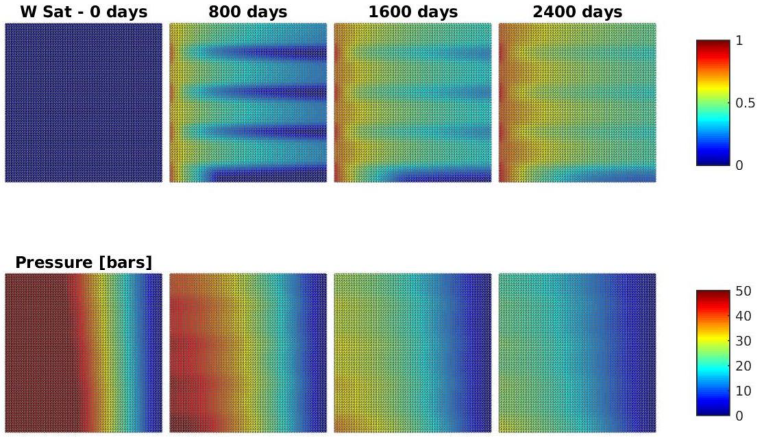

Figure 5: Pressure field for various times for a contrast between permeability values of 10^1, 64 x 64 grid cells, capillary pressure included.

Case 1 B: Injection through the left boundary, no capillary pressure, gravity included (3D).

Case 1		
	Water	Oil
$S_{0,x\neq 0,L_x}$	0	1
$S_{x=0}$	1	0
$S_{x=L_x}$	0	1
Case 2		
	Water	Oil
$S_{0,y\neq 0,L_y}$	0	1
$S_{y=0}$	1	0
$S_{y=L_y}$	0	1

Table 6: Saturations.

Property	Value	Units
T_{total}	4800	days
T_{steps}	120	
dT	80	days
Case 1		
$Q_{x=0}$	0.4	m^3/day
$P_{0,x\neq 0,L_x}$	100	$bars$
$P_{x=L_x}$	0	$bars$
Case 2		
$Q_{y=0}$	0.4	m^3/day
$P_{0,y\neq 0,L_y}$	100	$bars$
$P_{y=L_y}$	0	$bars$

Table 7: Initial values of the system.

Fr this case, we repeat the experiments performed in the 2D case, but for a 3D problem with 10 cells in the z direction. As in the previous case, each cell is one meter long. In Table 8, the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method. The pressure field and the water saturation are presented in Figure 6 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG	% of total ICCG
10^0	12503	DICCG$_{10}$	1025	910	1935	15
10^0	12503	DICCG$_{POD_{10}}$	1025	910	1935	15
10^0	12503	DICCG$_{POD_5}$	1025	920	1945	16
10^1	16458	DICCG$_{10}$	1174	1216	2390	15
10^1	16458	DICCG$_{POD_{10}}$	1174	1216	2390	15
10^1	16458	DICCG$_{POD_5}$	1174	1338	2512	15
10^2	16496	DICCG$_{10}$	1154	1471	2625	16
10^2	16496	DICCG$_{POD_{10}}$	1154	1471	2625	16
10^2	16496	DICCG$_{POD_5}$	1154	1551	2705	16
10^3	14261	DICCG$_{10}$	819	1235	2054	14
10^3	14261	DICCG$_{POD_{10}}$	819	1235	2054	14
10^3	14261	DICCG$_{POD_5}$	819	1339	2158	15

Table 8: Comparison between the ICCC and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the left boundary, domain 64 x 64 x 10 cells, no capillary pressure, gravity included.

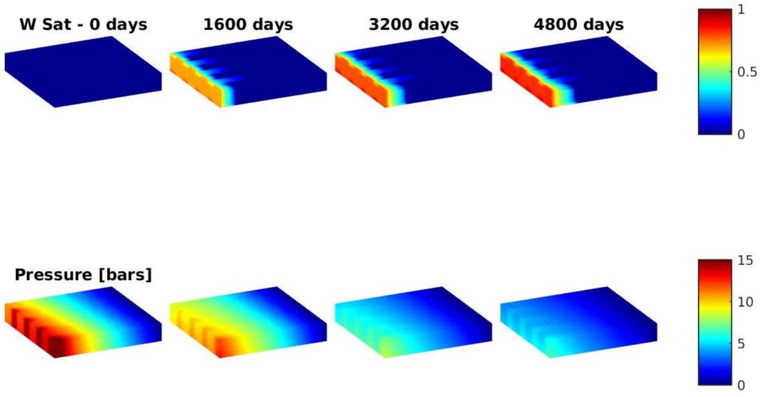

Figure 6: Pressure field for various times for a contrast between permeability values of 10^1, 64 x 64 x 10 grid cells, gravity included, no capillary pressure.

Case 1 C: Injection through the left boundary, capillary pressure and gravity included (3D).

In Table 9 the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selection of deflation vectors. The pressure field and the water saturation are presented in Figures 7 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^0	12347	DICCG$_{10}$	1017	733	1750	14
10^0	12347	DICCG$_{POD_{10}}$	1017	733	1750	14
10^0	12347	DICCG$_{POD_5}$	1017	746	1763	14
10^1	15454	DICCG$_{10}$	1176	941	2117	14
10^1	15454	DICCG$_{POD_{10}}$	1176	941	2117	14
10^1	15454	DICCG$_{POD_5}$	1176	976	2152	14
10^2	16334	DICCG$_{10}$	1146	1142	2288	14
10^2	16334	DICCG$_{POD_{10}}$	1146	1142	2288	14
10^2	16334	DICCG$_{POD_5}$	1146	1144	2290	14
10^3	13518	DICCG$_{10}$	807	1089	1896	14
10^3	13518	DICCG$_{POD_{10}}$	807	1089	1896	14
10^3	13518	DICCG$_{POD_5}$	807	953	1760	13

Table 9: Comparison between the ICCC and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the left boundary, domain 64 x 64 x 10 cells, capillary pressure and gravity included.

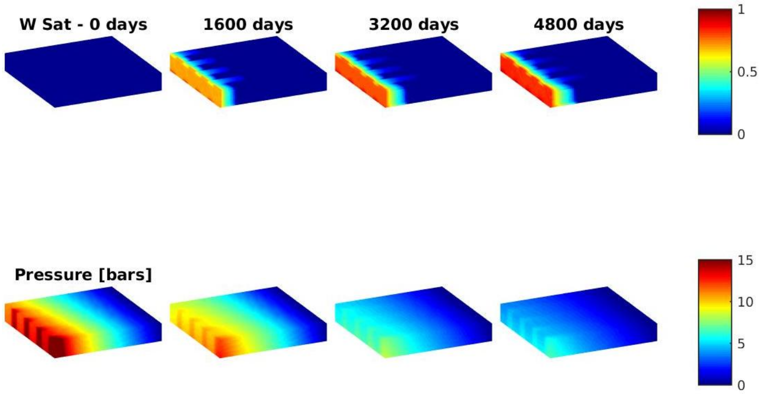

Figure 7: Pressure field for various times for a contrast between permeability values of 10^1, 64 x 64 x 10 grid cells, gravity and capillary pressure included.

Wells.

The experiments simulate flow through a porous medium with a constant porosity field of 0.25. For the first set of experiments, two wells are placed in the reservoir, for the second, five wells are placed. The pressure in the wells is presented for each experiment. On the boundaries, we imposed Neumann boundary conditions (no flux). We study a problem with layerd permeability values and the SPE 10 problem with different sizes. The characteristics of the fluid are presented in Table 10. For the relative permeability of the fluids, the Corey model is used. The curves of this model are presented in Figure 8. No gravity terms and no capillary pressure are taken into account in the first set of experiments.

	Water	Oil	Units
μ	1	10	cp
ρ	1000	850	kg/m^3
k_r	$(S_w)^2$	$(1-S_w)^2$	

Table 10: Fluids properties.

Well	Water Sat	Oil Sat	Pressure
I	0	1	100 bars
P	0	1	0 bars

Table 11: Wells properties.

Heterogeneous permeability layers, 2 wells, bhp controlled

Layers with different permeability, 35 x 35 cells.

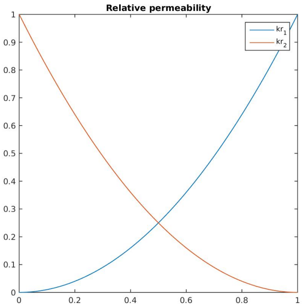

Figure 8: Fluid relative permeability

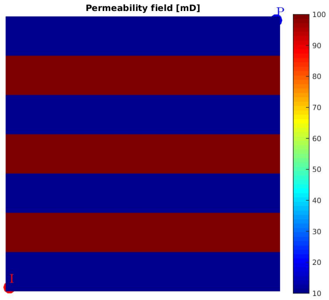

Figure 9: Rock perm.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG	% of total ICCG
10^0	30965	DICCG$_{10}$	395	35878	36273	117
10^0	30965	DICCG$_{POD_{10}}$	395	2998	3393	11
10^0	30965	DICCG$_{POD_5}$	395	4139	4534	15
10^1	45283	DICCG$_{10}$	570	45991	46561	103
10^1	45283	DICCG$_{POD_{10}}$	570	3406	3976	9
10^1	45283	DICCG$_{POD_5}$	570	4244	4814	11
10^2	50111	DICCG$_{10}$	629	50236	50865	102
10^2	50111	DICCG$_{POD_{10}}$	629	3587	4216	8
10^2	50111	DICCG$_{POD_5}$	629	4379	5008	10
10^3	53476	DICCG$_{10}$	670	53615	54285	102
10^3	53476	DICCG$_{POD_{10}}$	670	3826	4496	8
10^3	53476	DICCG$_{POD_5}$	670	4335	5005	9

Table 12: Comparison between the ICCC and DICCG methods of the average number of linear iterations for the second NR iteration for various contrast between permeability layers.

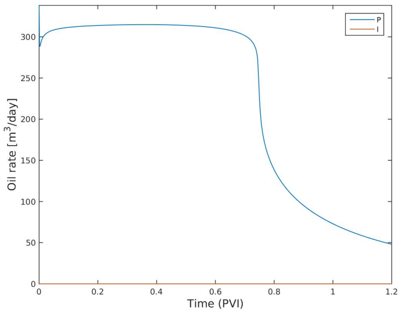

Figure 10: Oil Rate.

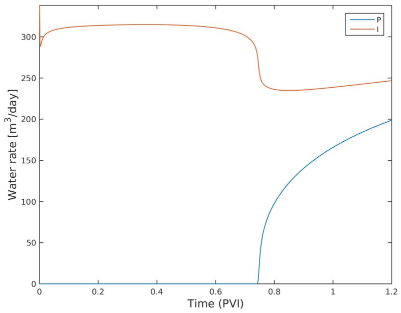

Figure 11: Water Rate.

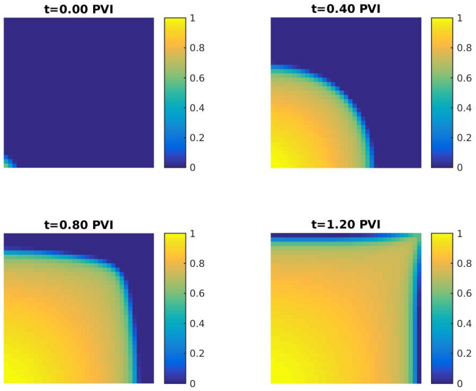

Figure 12: Water saturation.

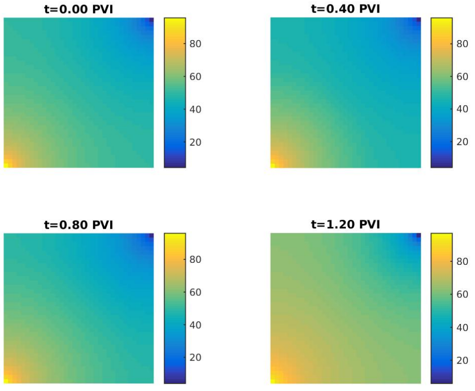

Figure 13: Pressure.

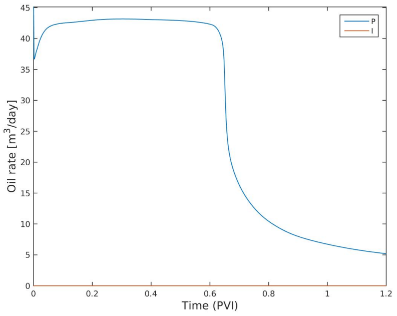

Figure 14: Oil Rate (perm 10-1).

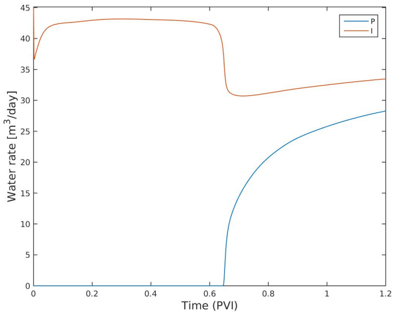

Figure 15: Water Rate (perm 10-1).

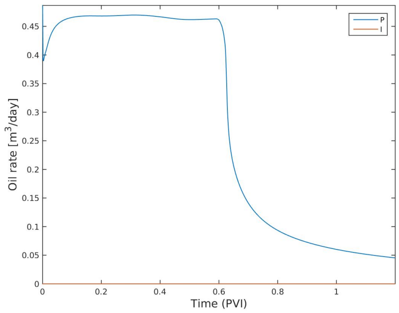

Figure 16: Oil Rate (perm 10-3).

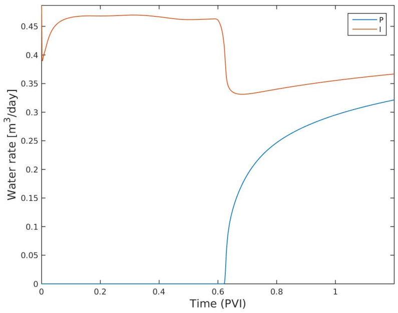

Figure 17: Water Rate (perm 10 -3).

Five wells, bhp

Heterogeneous permeability layers

For the first set of experiments, we study a layered system (see Figure 1). We use 6 layers of the same size, 3 layers with one value of permeability σ_1, followed by a layer with a different permeability value σ_2. The permeability of one set of layers is set to $\sigma_1 = 100mD$, the permeability of the other set σ_2 is varied. Therefore, the contrast in permeability between the layers ($\frac{\sigma_2}{\sigma_1} = \sigma_2$), depends on the value of σ_2. The permeability σ_2 varies from $\sigma_2 = 100mD$ to $\sigma_2 = 10^{-3}mD$. The domain consists of a Cartesian grid of 30 x 30 cells with a length of ten meters each cell.

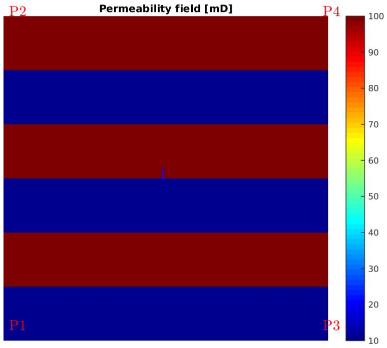

Figure 18: Rock permeability

One injector is placed in the center and four on the corners of the reservoir (see Figure 18). The wells are controled via the botomm hole pressure (bhp). Water is injected into a reservoir initially filled with oil. The values of the wells are presented in Table 24. The first simulation has constant permeability through all the reservoir. The simulation is runned until we inject 1.2 times the pore volumen of the reservoir, this process takes 345 days. We used this time, as the total time of the rest of the experiments. We use 200 time steps.

The number of iterations necessary to achieve convergence are presented in Table 15.

Well	Water Sat	Oil Sat	Pressure
P_1	0	1	50 bars
P_2	0	1	50 bars
P_3	0	1	50 bars
P_4	0	1	50 bars
I	1	0	200 bars

Table 13: Wells properties.

Property	Value
T_{total}	1.2 PV
T_{steps}	1.2/200 PV

Table 14: Initial values of the system.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG + DICCG	% of total ICCG
10^0	5445	DICCG$_{POD_{10}}$	280	632	912	17
10^0	5445	DICCG$_{POD_5}$	280	767	1047	19
10^1	8031	DICCG$_{POD_{10}}$	400	781	1181	15
10^1	8031	DICCG$_{POD_5}$	400	1011	1411	18
10^2	9788	DICCG$_{POD_{10}}$	483	210	693	7
10^2	9788	DICCG$_{POD_5}$	483	274	757	8
10^3	10779	DICCG$_{POD_{10}}$	531	437	968	9
10^3	10779	DICCG$_{POD_5}$	531	484	1015	9

Table 15: Comparison between the ICCC and DICCG methods.

Homogeneous permeability

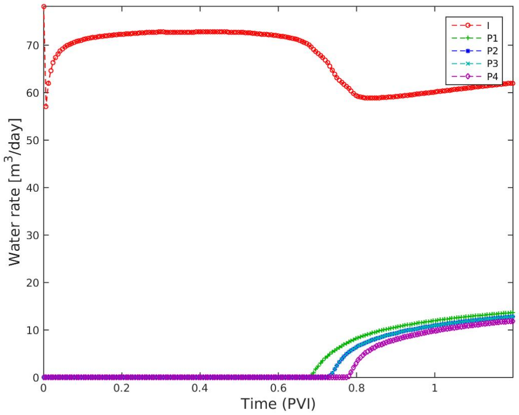

Figure 19: Water Rate, homogeneous permeability.

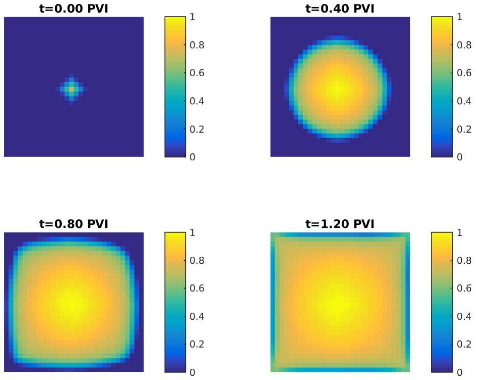

Figure 20: Water Saturation, homogeneous permeability.

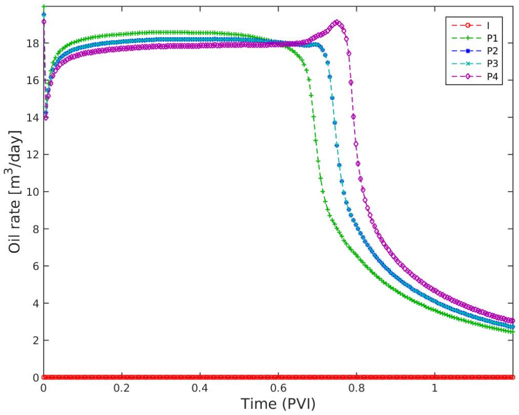

Figure 21: Oil rate, homogeneous permeability.

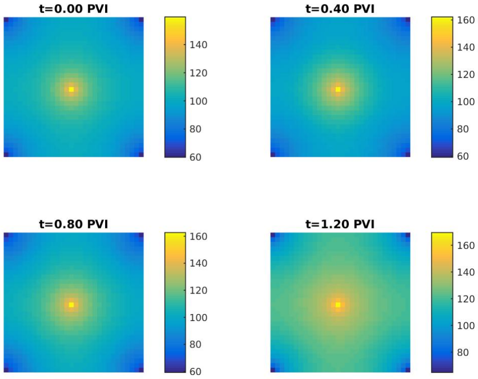

Figure 22: Pressure, homogeneous permeability.

Results heterogeneous permeability (contrast 10^1)

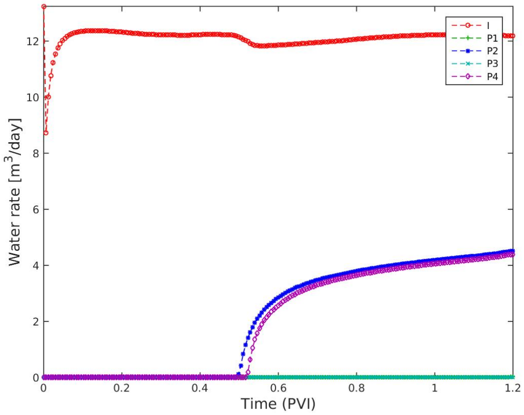

Figure 23: Water Rate, heterogeneous permeability (contrast 10^1).

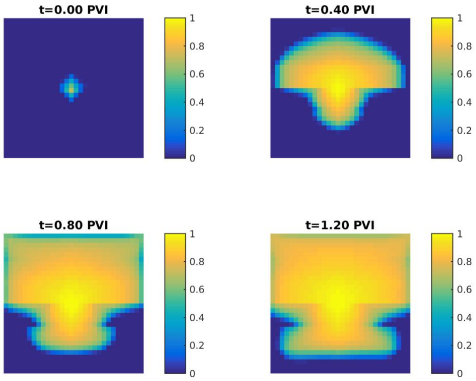

Figure 24: Water Saturation, heterogeneous permeability (contrast 10^1).

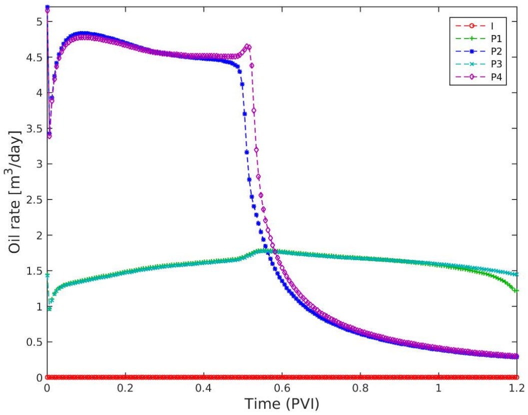

Figure 25: Oil rate, heterogeneous permeability (contrast 10^1).

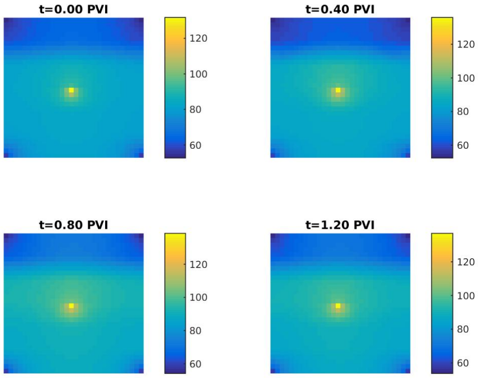

Figure 26: Pressure, heterogeneous permeability (contrast 10^1).

Results heterogeneous permeability (contrast 10^3)

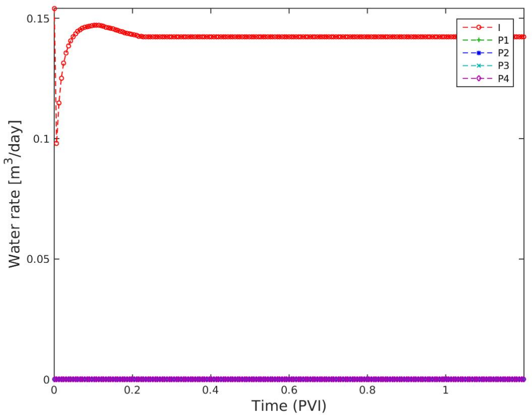

Figure 27: Water Rate, heterogeneous permeability (contrast 10^3).

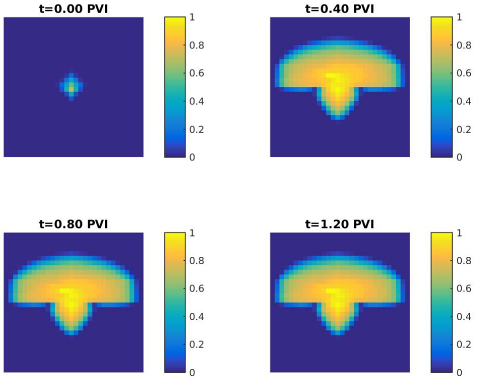

Figure 28: Water Saturation, heterogeneous permeability (contrast 10^3).

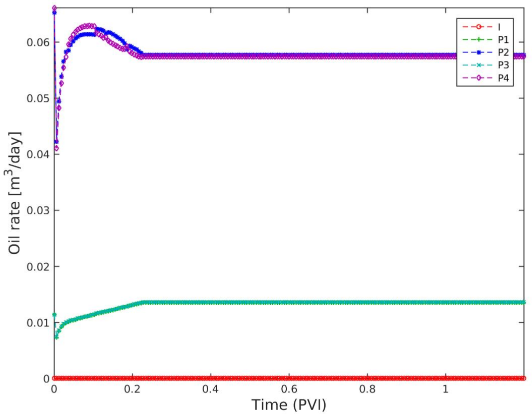

Figure 29: Oil rate, heterogeneous permeability (contrast 10^3).

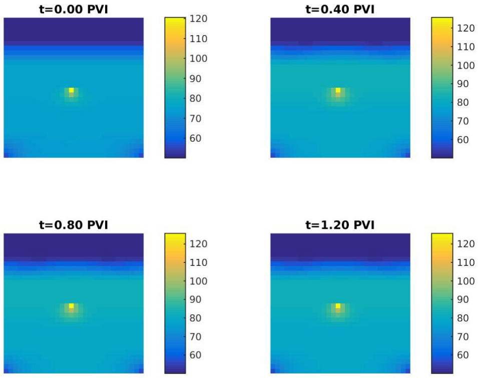

Figure 30: Pressure, heterogeneous permeability (contrast 10^3).

2.2 SPE 10

We study the SPE 10 model with 5 wells, 4 producers and one injector (see Figure **??**). The original model contains 60 x 220 x 85 cells, in these experiments the grid sized is 16 x 56 x 1. Only one layer is studied. The permeability and porosity fields are upscaled averaging the value in each grid cell using the function *sampleFromBox* from MRST. The permeability contrast for this problem is 1.05×10^7.

Well	Water Sat	Oil Sat	Pressure
P_1	0	1	275 bars
P_2	0	1	275 bars
P_3	0	1	275 bars
P_4	0	1	275 bars
I	1	0	1100 bars

Table 16: Wells properties.

Property	Value
T_{total}	1.2 PV
T_{steps}	1.2/200 PV

Table 17: Initial values of the system.

Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG + DICCG	% of total ICCG
11313	DICCG$_{POD_{10}}$	567	1615	2182	19
11313	DICCG$_{POD_5}$	567	2235	2802	25

Table 18: Comparison between the ICCC and DICCG methods of the average number of linear iterations for the SPE 10 problem, 16x56 grid cells.

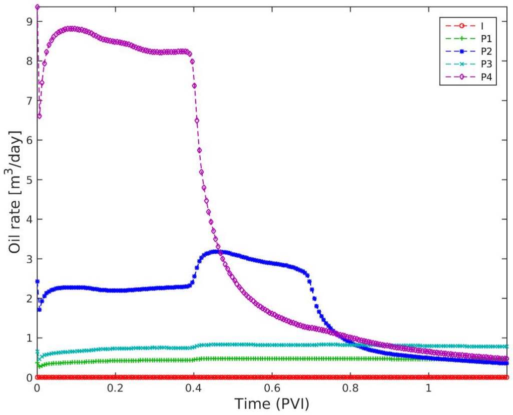

Figure 31: Oil rate, SPE 10, 16x56 grid cells.

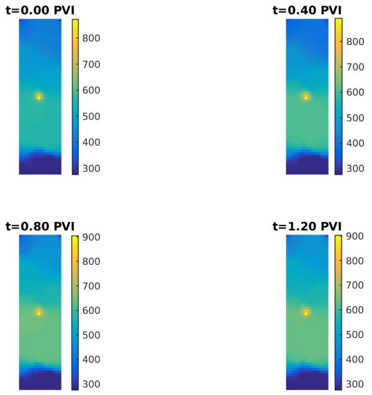

Figure 32: Pressure, SPE 10, 16x56 grid cells.

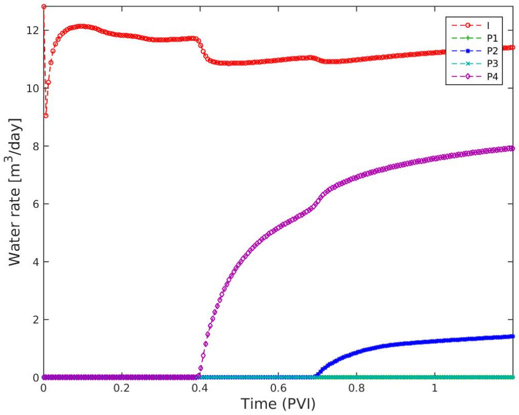

Figure 33: Water rate, SPE 10, 16x56 grid cells.

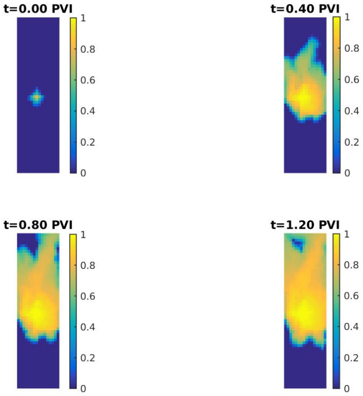

Figure 34: Water Saturation, SPE 10, 16x56 grid cells.

2.3 SPE 10, training simulation

We study the SPE 10 model with 5 wells, 4 producers and one injector (see Figure **??**). The original model contains 60 x 220 x 85 cells, in these experiments the grid sized is 16 x 56 x 1. Only one layer is studied. The permeability and porosity fields are upscaled averaging the value in each grid cell using the function *sampleFromBox* from MRST. The permeability contrast for this problem is 1.05×10^7.

Well	Water Sat	Oil Sat	Pressure
P_1	0	1	$rand(0-275)$ bars
P_2	0	1	$rand(0-275)$ bars
P_3	0	1	$275 - P_1$ bars
P_4	0	1	$275 - P_2$ bars
I	1	0	1100 bars

Table 19: Wells properties.

Property	Value
T_{total}	2 PV
T_{steps}	2/200 PV

Table 20: Initial values of the system.

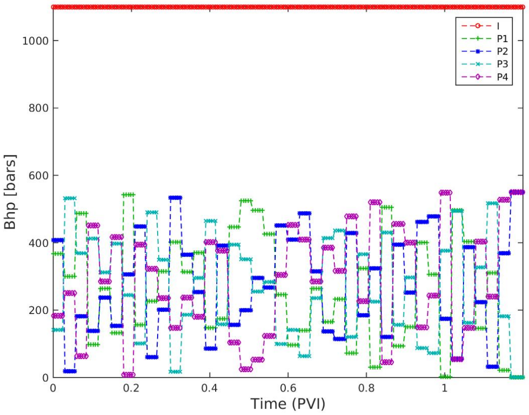

Figure 35: Bhp, SPE 10, 16x56 grid cells. tring simulation.

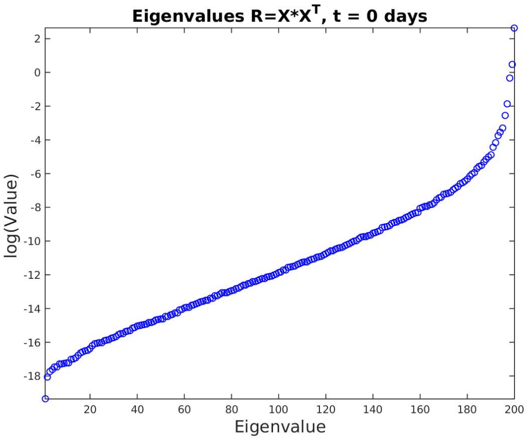

Figure 36: Eigenvalues training simulation, 200 timesteps.

SPE10

Well	Water Sat	Oil Sat	Pressure
P_1	0	1	275 bars
P_2	0	1	275 bars
P_3	0	1	275 bars
P_4	0	1	275 bars
I	1	0	1100 bars

Table 21: Wells properties.

Property	Value
T_{total}	1.2 PV
T_{steps}	1.2/200 PV

Table 22: Initial values of the system.

Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG + DICCG	% of total ICCG
11362	DICCG$_{POD_{30}}$	106	1734	1840	16
11362	DICCG$_{POD_{10}}$	151	2498	2649	23

Table 23: Comparison between the ICCC and DICCG methods of the average number of linear iterations for the SPE 10 problem, 16x56 grid cells.

.

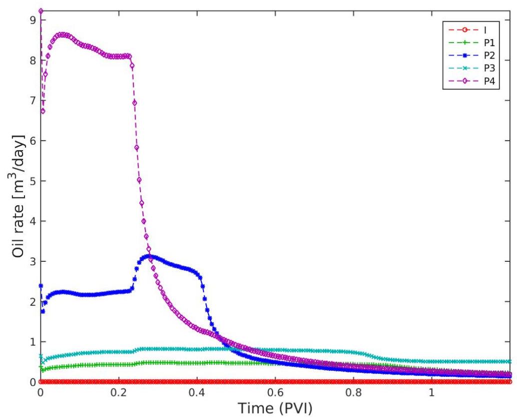

Figure 37: Oil rate, SPE 10, 16x56 grid cells.

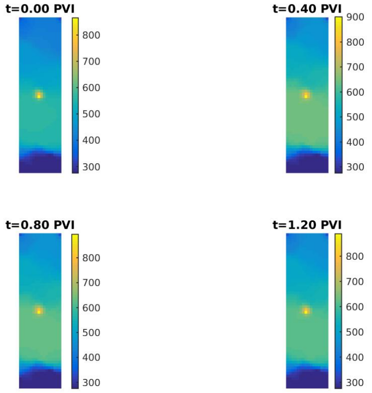

Figure 38: Pressure, SPE 10, 16x56 grid cells.

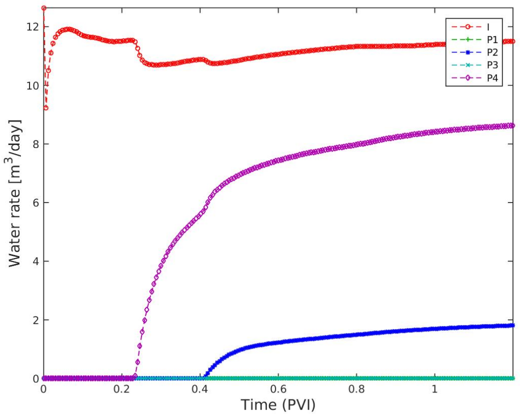

Figure 39: Water rate, SPE 10, 16x56 grid cells.

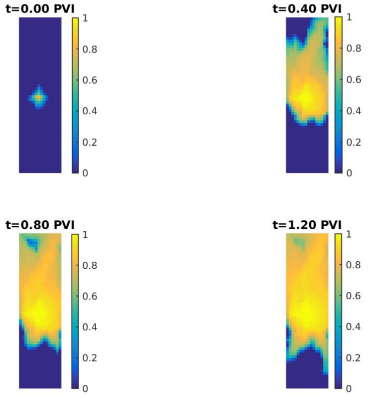

Figure 40: Water Saturation, SPE 10, 16x56 grid cells.

Heterogeneous permeability layers, 2 wells, rate controlled

Layers with different permeability, 35 x 35 cells.

Well	Water Sat	Oil Sat	Pressure
I	0	1	$300m^3/day$
P	0	1	0 bars

Table 24: Wells properties.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG	% of total ICCG
10^0	37089	DICCG$_{10}$	462	41878	42340	114
10^0	37089	DICCG$_{POD_{10}}$	462	31830	32292	87
10^0	37089	DICCG$_{POD_5}$	462	32355	32817	88

Table 25: Comparison between the ICCC and DICCG methods of the average number of linear iterations for the second NR iteration for various contrast between permeability layers.

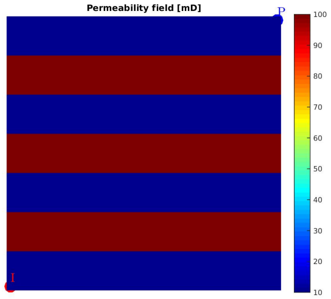

Figure 41: Rock perm.

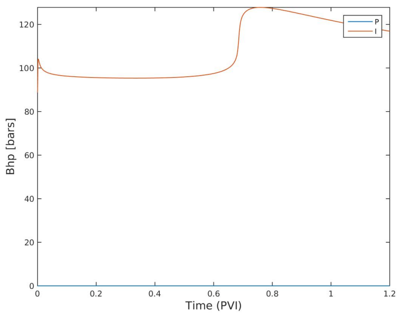

Figure 42: Bhp, homogeneous perm.

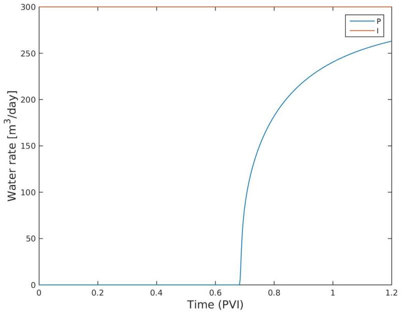

Figure 43: Water Rate.

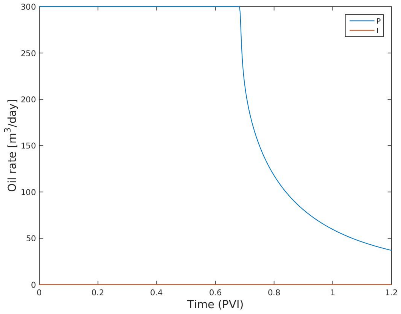

Figure 44: Oil rate.

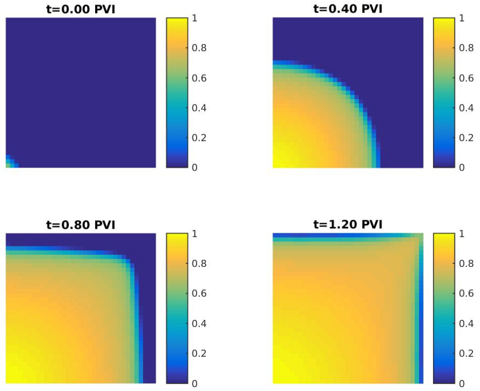

Figure 45: Water saturation.

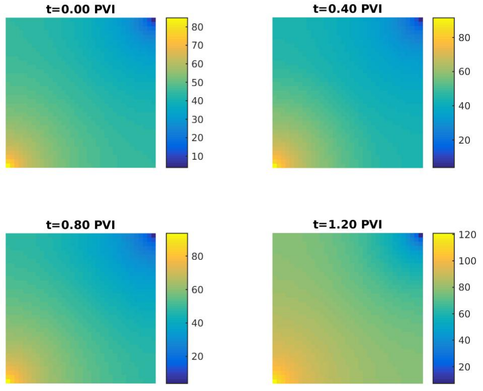

Figure 46: Pressure.

References

[1] K. Aziz and A. Settari. *Petroleum reservoir simulation.* Chapman & Hall, 1979.

[2] Z. Chen, G. Huan and Y. Ma. *Computational methods for multiphase flows in porous media.* SIAM, 2006.

[3] J.D. Jansen. *A systems description of flow through porous media.* Springer, 2013.

[4] C. Vuik, A. Segal and J. A. Meijerink. An Efficient Preconditioned CG Method for the Solution of a Class of Layered Problems with Extreme Contrasts in the Coefficients. *Journal of Computational Physics*, 152:385, 1999.

[5] J. Tang. *Two-Level Preconditioned Conjugate Gradient Methods with Applications to Bubbly Flow Problems.* PhD thesis, Delft University of Technology, 2008.

[6] J.M. Tang, R. Nabben, C. Vuik and Y. Erlangga. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *Journal of scientific computing*, 39(3):340–370, 2009.

[7] M. Clemens, M. Wilke, R. Schuhmann and T. Weiland. Subspace projection extrapolation scheme for transient field simulations. *IEEE Transactions on Magnetics*, 40(2):934–937, 2004.

[8] C. Vuik, A. Segal, L. Yaakoubi and E. Dufour. A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients. *Applied Numerical Mathematics*, 41(1):219–233, 2002.

[9] B. Smith, P. Bjorstad and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press New York, 1996.

[10] P. Astrid, G. Papaioannou, J.C. Vink and J.D. Jansen. Pressure Preconditioning Using Proper Orthogonal Decomposition. In *2011 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA*, January 2011.

[11] R. Markovinović and J.D. Jansen. Accelerating iterative solution methods using reduced-order models as solution predictors. *International journal for numerical methods in engineering*, 68(5):525–541, 2006.

[12] K.A. Lie. *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST).* SINTEF ICT, 2013.

[13] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics Philadelphia, PA, USA. 2nd edition, 2003.

A List of notation

Symbol	Quantity	Unit
ϕ	Rock porosity	
$\mathbf{K}$	Rock permeability	$Darcy$ (D)
c_r	Rock compressibility	Pa^{-1}
$\mathbf{v}$	Darcy's velocity	m/d
ρ	Fluid density	kg/m^3
μ	Fluid viscosity	$Pa \cdot s$
p	Pressure	Pa
g	Gravity	m/s^2
c_f	Fluid compressibility	Pa^{-1}
q	Sources	

Table 26: Notation

B Stopping criteria

When we use an iterative method, we always want that our approximation is close enough to the exact solution. In other words, we want that the error [13, pag. 42]:

$$||\mathbf{e}^k||_2 = ||\mathbf{x} - \mathbf{x}^k||_2,$$

or the relative error:

$$\frac{||\mathbf{x} - \mathbf{x}^k||_2}{||\mathbf{x}||_2},$$

is small.
When we want to chose a stopping criteria, we could think that the relative error is a good candidate, but is has the disadvantage that we need to know the exact solution to compute it. What we have instead is the residual

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k,$$

that is actually computed in each iteration of the CG method. There is a relationship between the error and the residual that can help us with the choice of the stopping criteria.

$$\frac{||\mathbf{x} - \mathbf{x}^k||_2}{||\mathbf{x}||_2} \leq \kappa_2(A)\frac{||\mathbf{r}^k||_2}{||\mathbf{b}||_2}.$$

With this relationship in mind, we can choose the stopping criteria as an ϵ for which

$$\frac{||\mathbf{r}^k||_2}{||\mathbf{b}||_2} \leq \epsilon.$$

But we should keep to have in mind the condition number of the matrix $\mathbf{A}$, because the relative error will be bounded by:

$$\frac{||\mathbf{x} - \mathbf{x}^k||_2}{||\mathbf{x}||_2} \leq \kappa_2(A)\epsilon.$$

C Singular Value Decomposition for POD

If we perform SVD in $\mathbf{X}$, we have
$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T, \qquad \mathbf{U} \in \mathbb{R}^{n\times n}, \qquad \mathbf{\Sigma} \in \mathbb{R}^{n\times m}, \qquad \mathbf{V} \in \mathbb{R}^{m\times m}.$
Then we have

$$\begin{aligned}\mathbf{R} &= \mathbf{X}\mathbf{X}^T \\ &= \mathbf{U}\Sigma\mathbf{V}^T(\mathbf{U}\Sigma\mathbf{V}^T)^T \\ &= \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma^T\mathbf{U}^T,\ \mathbf{V}^T\mathbf{V} = \mathbf{I} \\ &= \mathbf{U}\Lambda\mathbf{U}^T,\ \Lambda = \Sigma\Sigma^T \in \mathbb{R}^{n\times n}\end{aligned} \qquad \begin{aligned}\mathbf{R}^T &= \mathbf{X}^T\mathbf{X} \\ &= (\mathbf{U}\Sigma\mathbf{V}^T)^T\mathbf{U}\Sigma\mathbf{V}^T \\ &= \mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T,\ \mathbf{U}^T\mathbf{U} = \mathbf{I} \\ &= \mathbf{V}\Lambda^T\mathbf{V}^T,\ \Lambda^T = \Sigma^T\Sigma \in \mathbb{R}^{m\times m}.\end{aligned}$$

$$\begin{aligned}\mathbf{X} &= \mathbf{U}\Sigma\mathbf{V}^T \\ \mathbf{U} &= \mathbf{X}\mathbf{V}\Sigma^{-1} \\ \mathbf{U} &= \mathbf{X}\mathbf{V}\Lambda^{-\frac{1}{2}}\end{aligned}$$

If we compute Λ^T, we can compute U as follows:

$$\mathbf{U} = \mathbf{X}\mathbf{V}(\Lambda^T)^{-\frac{T}{2}} = \mathbf{X}\mathbf{V}(\Lambda^T)^{\frac{1}{2}}$$

D Deflation method

In this appendix, we explain how to obtain the solution of the linear system (**??**) with deflation. Some properties of the matrices used for deflation that will help us to find the solution of system (**??**) are [6]:

a) $\mathbf{P}^2 = \mathbf{P}$.

b) $\mathbf{A}\mathbf{P}^T = \mathbf{P}\mathbf{A}$.

c) $(\mathbf{I} - \mathbf{P}^T)\mathbf{x} = \mathbf{Q}\mathbf{b}$.

d) $\mathbf{P}\mathbf{A}\mathbf{Q} = \mathbf{0}^{n\times n}$.

e) $\mathbf{P}\mathbf{A}\mathbf{Z} = \mathbf{0}^{n\times l}$.

To obtain the solution of the linear system (**??**), we start with the splitting:

$$\mathbf{x} = \mathbf{x} - \mathbf{P}^T\mathbf{x} + \mathbf{P}^T\mathbf{x} = (\mathbf{I} - \mathbf{P}^T)\mathbf{x} + \mathbf{P}^T\mathbf{x}. \tag{33}$$

Multiplying expression (33) by $\mathbf{A}$, using the properties of the deflation matrices, we have:

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{A}(\mathbf{I} - \mathbf{P}^T)\mathbf{x} + \mathbf{A}\mathbf{P}^T\mathbf{x}, & \qquad\qquad \mathit{Property}: \\ \mathbf{A}\mathbf{x} &= \mathbf{A}\mathbf{Q}\mathbf{b} + \mathbf{A}\mathbf{P}^T\mathbf{x}, & c) \\ \mathbf{b} &= \mathbf{A}\mathbf{Q}\mathbf{b} + \mathbf{P}\mathbf{A}\mathbf{x}, & b), \end{aligned}$$

multiplying by $\mathbf{P}$ and using the properties $\mathbf{P}\mathbf{A}\mathbf{Q} = \mathbf{0}^{n\times n}$ and $\mathbf{P}^2 = \mathbf{P}$, properties *d*) and *a*), we have:

$$\begin{aligned} \mathbf{P}\mathbf{A}\mathbf{Q}\mathbf{b} + \mathbf{P}^2\mathbf{A}\mathbf{x} &= \mathbf{P}\mathbf{b}, \\ \mathbf{P}\mathbf{A}\mathbf{x} &= \mathbf{P}\mathbf{b}, \end{aligned}$$

where $\mathbf{P}\mathbf{A}\mathbf{x} = \mathbf{P}\mathbf{b}$ is the deflated system. Since $\mathbf{P}\mathbf{A}$ is singular, the solution of Equation (34) can contain components of the null space of $\mathbf{P}\mathbf{A}$, $(\mathcal{N}(\mathbf{P}\mathbf{A}))$. A solution of this system, called the deflated solution, is denoted by $\hat{\mathbf{x}}$. Then, the linear system to solve is:

$$\mathbf{P}\mathbf{A}\hat{\mathbf{x}} = \mathbf{P}\mathbf{b}. \tag{34}$$

As the solution of Equation (34) can contain components of $\mathcal{N}(\mathbf{P}\mathbf{A})$, $\hat{\mathbf{x}}$ can be decomposed as:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{y}, \tag{35}$$

with $\mathbf{y} \in \mathcal{R}(\mathbf{Z}) \subset \mathcal{N}(\mathbf{P}\mathbf{A})$, and $\mathbf{x}$ the solution of Equation (**??**).
Note: If $\mathbf{y} \in \mathcal{R}(\mathbf{Z})$, then

$$\mathbf{y} = \sum_{i=1}^{m} \alpha_i \mathbf{z}_i,$$

$$\mathbf{PAy} = \mathbf{PA}(\mathbf{z}_1\alpha_1 + ... + \mathbf{z}_m\alpha_m) = \mathbf{PAZ}\alpha,$$

from property e) we have:

$$\mathbf{PAy} = \mathbf{0}.$$

Therefore $\mathcal{R}(\mathbf{Z}) \subset \mathcal{N}(\mathbf{PA})$, and using property b) we have:

$$\mathbf{PAy} = \mathbf{AP}^T\mathbf{y} = \mathbf{0}.$$

As $\mathbf{A}$ is invertible, we have:

$$\mathbf{P}^T\mathbf{y} = \mathbf{0}. \tag{36}$$

Multiplying Equation (35) by $\mathbf{P}^T$ we obtain:

$$\mathbf{P}^T\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x} + \mathbf{P}^T\mathbf{y}.$$

substituting Equation (36) we arrive to:

$$\mathbf{P}^T\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x}. \tag{37}$$

Substitution of Equation (37) and property c) in Equation (33) leads to:

$$\mathbf{x} = \mathbf{Qb} + \mathbf{P}^T\hat{\mathbf{x}}, \tag{38}$$

which gives us the relation between $\hat{\mathbf{x}}$ and $\mathbf{x}$.