

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

G. B. Diaz Cortes¹, C. Vuik¹ and J. D. Jansen²

¹Department of Applied Mathematics, TU Delft

²Department of Geoscience & Engineering, TU Delft

April 2017

Abstract

We study fast and robust iterative solvers for large systems of linear equations resulting from simulation of flow through strongly heterogeneous porous media. We propose the use of preconditioning and deflation techniques, based on information obtained from the system, to reduce the time spent in the solution of the linear system.

An important question when using deflation techniques is how to find good deflation vectors, which lead to a decrease in the number of iterations and a small increase in the required computing time per iteration. In this paper, we propose the use of deflation vectors based on a POD-reduced set of snapshots. We investigate convergence and the properties of the resulting methods. Finally, we illustrate these theoretical results with numerical experiments. We consider compressible and incompressible single-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 . Using deflation for the incompressible problem, we reduce the number of iterations to 1 or 2 iterations. With deflation, for the compressible problem, we reduce up to $\sim 80\%$ the number of iterations when compared with the only-preconditioned solver.

1 Introduction.

Often, most computational time in the simulation of multi-phase flow through porous media is taken up by the solution of the pressure equation. This involves, primarily, solving large systems of linear equations as part of the iterative solution of the time and space discretized governing nonlinear partial differential equations. The time spent in solving the linear systems depends on the size of the problem and the heterogeneity, i.e. the spatial variations of rock permeability values within the medium (permeability is an inverse measure of the resistance to flow which is related to the porosity and the pore structure of the rock). Solution of problems with extreme contrasts in the permeability values may lead to very large computing times.

Iterative methods are known to be the best option to solve such extreme problems. However, sometimes iterative methods are not sufficient to solve these problems in a reasonable amount of time. As the systems become larger or ill-conditioned, finding a way to accelerate the convergence of these methods becomes necessary. Preconditioning is a way to accelerate convergence, but new preconditioning techniques still need to be developed to improve the performance of iterative methods [? ?]. Reduced Order Models (ROM) have also been studied to improve computational efficiency by reducing the model size without losing essential information [? ? ?]. A potential ROM to reduce the computing time for large-scale problems is Proper Orthogonal Decomposition (POD), a method that has been investigated for flow problems in porous media in [? ? ? ? ? ? ? ? ? ?] among others. The use of a POD-based preconditioner for acceleration of the solution is proposed by Astrid et al. [?] to solve the pressure equation resulting from two-phase reservoir simulation, by Jiang et al. [?] for a similar application and by Pasetto et al. [?] for groundwater flow models.

The POD method requires the computation of a series of 'snapshots' which are solutions of the problem with slightly different parameters or well inputs. Astrid et al. [?] use snapshots in the form of solutions of the pressure equation computed in a small number of short pre-simulations, prior to the actual simulation, with diverse well configurations, reporting promising speed ups with factors between three and five. They note that the overhead required to pre-compute the POD solutions implies that the method will be particularly attractive when many solutions of near-similar simulation models are required. A similar approach is followed by Jiang [?], who concludes that POD-based pressure preconditioning does not appear to be an ideal choice because of its dependence on the differences between the right-hand sides (forcing terms) used in the pre-simulations and the actual simulation. The snapshots computed by Pasetto et al. [?] are solutions of the previous time steps in the full-model. Once the snapshots are computed, the POD method is used to obtain a set of basis vectors that capture the most relevant features of the system, which can be used to speed-up the subsequent simulations.

The method of Pasetto at al. [?] is partly based on the work of Markovinovic and Jansen [?] who use a similar, but more restricted, approach in which the acceleration is achieved by only improving the initial guess.

Problems with high contrast between the permeability coefficients are sometimes ap-

proached through the use of deflation techniques, see, e.g., [?]. These techniques involve the search of good deflation vectors, which are usually problem-dependent. In [?], subdomain based deflation vectors are used for layered problems with a large contrast between permeability coefficients. However, these deflation vectors cannot be used if the distribution of the permeability coefficients is not structured, as is usually the case in reservoir simulation models; see, e.g., the well-known SPE 10 benchmark problem [?].

Algebraic Multigrid (AMG)[?], Multi-level and Domain Decomposition [?] preconditioners have been studied in combination deflation techniques to accelerate the convergence of iterative methods. In [? ?] and [?], after computing a basis from the previously obtained snapshots, the solution is computed in the subspace generated by this basis and then projected back to the original high-dimensional system. Carlberg et al. [?] also use POD to obtain information from the system, in particular, the previous time step solutions. Then, a Krylov-subspace is constructed using the information obtained previously.

Following the ideas of [? ? ? ?], we propose the use of POD of many snapshots to capture the system's behavior and combine this technique with deflation to accelerate the convergence of an iterative Krylov method. In this work, instead of computing the solution in a low dimensional subspace, the basis obtained with POD is proposed as an alternative choice for the deflation vectors to accelerate the convergence of the pressure solution in reservoir simulation.

This work is divided into six sections. Section 2 is devoted to a detailed description of the models used to simulate flow through a porous medium. In Section ??, we present some theory about the linear solvers used in this work and we introduce preconditioning and deflation techniques. In Section ?? we present some theory about POD. We prove two lemmas that will help us in the choice of good deflation vectors for the incompressible case in Section ??.

In Section 3 we present numerical experiments. We describe the problem that is studied, the solver and the preconditioning and deflation techniques used to speed up the solver. The results are also presented in this section. Finally, we end with the conclusions.

2 Flow through porous media

When simulating flow of two phases 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 phases is known as the interface.

While 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 a porous medium. If the phase is not present, we have a zero saturation of that phase. If there are two phases present in the porous medium, these fluids fill completely the empty space, which is expressed by the following relation.

$$S_n + S_w = 1 \quad (1)$$

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

$$p_c(S_w) = p_n - p_w. \quad (2)$$

The pressure in the non-wetting fluid is higher than the pressure in 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's common to use a function called Leverett J-function, which takes the following form:

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

where σ is the surface tension 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:

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

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

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

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 modelling 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.

$$\sum_{\alpha} K_{\alpha}^e < K.$$

The saturation dependent relative permeability is defined as:

$$k_{r\alpha}(S) = K_{\alpha}^e / K.$$

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

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

where $n_w > 1$, $n_o > 1$ and k_{α}^0 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 equations for each phase α are given by:

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

Darcy's law is:

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

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

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

Incompressible two-phase flow

In the case of incompressible flow, the porosity ϕ and the densities ρ_{α} doesn't depend on time. Therefore, Equation (9) reduces to:

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

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

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

Fractional flow formulation

Part of the non linearity of previous formulation can be eliminated if the system is expressed in terms of one phase pressure and one phase saturation. A common choice is to use p_n and S_w which gives the following system

$$\frac{\partial(\phi\rho_w S_w)}{\partial t} + \nabla \cdot (\rho_w \frac{Kk_{rw}}{\mu_w} (\nabla p_n - \nabla P_c(S_w) - \rho_w g \nabla z)) = \rho_w q_w, \quad (10)$$

$$\frac{\partial(\phi\rho_n(1 - S_w))}{\partial t} + \nabla \cdot (\rho_n \frac{Kk_{rn}}{\mu_n} (\nabla p_n - \rho_n g \nabla z)) = \rho_n q_n. \quad (11)$$

Incompressible flow

For incompressible flow, only the Saturation S is a function of time and the fluid densities ρ_α are constant. Therefore, the mass-balance equations are:

$$\phi \frac{\partial(S_\alpha)}{\partial t} + \nabla \cdot (\mathbf{v}_\alpha) = q_\alpha. \quad (12)$$

The total Darcy velocity is defined as the sum of the velocity in the wetting and non wetting phases, it can be expressed in terms of the non-wetting phase as:

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

if we add the two continuity equations and use the relationship $S_n + S_w = 1$ we have:

$$\phi \frac{\partial(S_w + S_n)}{\partial t} + \nabla \cdot (\mathbf{v}_w + \mathbf{v}_n) = q_n + q_w.$$

Using $\lambda = \lambda_n + \lambda_w = \lambda K$ as the total mobility and $q = q_n + q_w$ as the total source, for the total Darcy velocity we have:

$$-\nabla \cdot (\lambda K \nabla p_n) = q - \nabla [\lambda_w \nabla p_c + (\lambda_n \rho_n + \lambda_w \rho_w) g \nabla z].$$

Multiplying each phase velocity by the relative mobility of the other phase and subtracting the result we obtain:

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

Therefore, for \mathbf{v}_w we have

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

Using the velocity computed above, for the wetting phase we have:

$$\phi \frac{\partial(S_w)}{\partial t} + \nabla \cdot [f_w(\mathbf{v} + \lambda_n \Delta \rho g \nabla z)] = q_w - \nabla \cdot (f_w \lambda_n p_c \nabla S_w). \quad (13)$$

with $\Delta\rho = \rho_w - \rho_n$ and the fractional flow function f_w :

$$f_w = \frac{\lambda_w}{\lambda_n + \lambda_w},$$

is the fraction of the total flow that consists of the wetting fluid, and

$$\mathbf{v} = -\lambda(\nabla p_n - f_w \nabla p_w - (f_n \rho_n + f_w \rho_w) g \nabla z)$$

The coupling between the elliptic pressure equation and the parabolic saturation equation is much weaker than the coupling between the two continuity equations in the two-pressure formulation. In the pressure equation, the coupling to saturation appears explicitly in the effective mobility that makes up the variable coefficient in the Poisson problem and on the right-hand side through the phase mobilities and the derivative of the capillary function. In Equation (21), the saturation is only indirectly coupled to the pressure through the total Darcy velocity.

The system of PDEs can then be reformulated so that it consists of an elliptic equation for fluid pressure and one or more transport equations. These transport equations are generally parabolic, but have a strong hyperbolic character. Since the pressure and saturations equations have very different mathematical characteristics, it is natural to solve them in consecutive substeps. Sequential solution procedures and incompressible flow models are popular in academia and for research purposes, but are less used in industry. To the extent simulations are used for practical reservoir engineering, they are mainly based on compressible equations and solution procedures in which flow and transport are solved as a coupled system. Such approaches are very robust and particularly useful for problems with large variations in time constants or strong coupling between different types of flow mechanisms.

Immiscible two-phase flow

For the case of immiscible fluids, the flow equations are:

$$\frac{\partial(\phi\rho_w S_w)}{\partial t} + \nabla \cdot (\rho_w \mathbf{v}_w) = \rho_w q_w, \quad (14)$$

$$\frac{\partial(\phi\rho_n S_n)}{\partial t} + \nabla \cdot (\rho_n \mathbf{v}_n) = \rho_n q_n. \quad (15)$$

Pressure formulation

We can choose the pressures as the primary unknowns, then the saturations are expressed as functions of pressure. Assuming the capillary pressure has a unique inverse function $\hat{S}_w = P_c^{-1}(p_c)$, then we have:

$$S_w = \hat{S}_w(p_n - p_w) \quad S_n = 1 - \hat{S}_w(p_n - p_w),$$

then equations (14) and (15) can be written as:

$$\frac{\partial(\phi\rho_w\hat{S}_w)}{\partial t} + \nabla \cdot (\rho_w \frac{Kk_{rw}(\hat{S}_w)}{\mu_w} (\nabla p_w - \rho_w g \nabla z)) = \rho_w q_w, \quad (16)$$

$$\frac{\partial(\phi\rho_n(1 - \hat{S}_w))}{\partial t} + \nabla \cdot (\rho_n \frac{Kk_{rn}(\hat{S}_w)}{\mu_n} (\nabla p_n - \rho_n g \nabla z)) = \rho_n q_n. \quad (17)$$

Previous system is highly coupled and strongly nonlinear.

Fractional flow formulation

Part of the non linearity of previous formulation can be eliminated if the system is expressed in terms of one phase pressure and one phase saturation. A common choice is to use p_n and S_w which gives the following system

$$\frac{\partial(\phi\rho_w S_w)}{\partial t} + \nabla \cdot (\rho_w \frac{Kk_{rw}}{\mu_w} (\nabla p_n - \nabla P_c(S_w) - \rho_w g \nabla z)) = \rho_w q_w, \quad (18)$$

$$\frac{\partial(\phi\rho_n(1 - S_w))}{\partial t} + \nabla \cdot (\rho_n \frac{Kk_{rn}}{\mu_n} (\nabla p_n - \rho_n g \nabla z)) = \rho_n q_n. \quad (19)$$

Incompressible flow

For incompressible flow, only the Saturation S is a function of time and the fluid densities ρ_α are constant. Therefore, the mass-balance equations are:

$$\phi \frac{\partial(S_\alpha)}{\partial t} + \nabla \cdot (\mathbf{v}_\alpha) = q_\alpha. \quad (20)$$

The total Darcy velocity is defined as the sum of the velocity in the wetting and non wetting phases, it can be expressed in terms of the non-wetting phase as:

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

if we add the two continuity equations and use the relationship $S_n + S_w = 1$ we have:

$$\phi \frac{\partial(S_w + S_n)}{\partial t} + \nabla \cdot (\mathbf{v}_w + \mathbf{v}_n) = q_n + q_w.$$

Using $\lambda = \lambda_n + \lambda_w = \lambda K$ as the total mobility and $q = q_n + q_w$ as the total source, for the total Darcy velocity we have:

$$-\nabla \cdot (\lambda K \nabla p_n) = q - \nabla[\lambda_w \nabla p_c + (\lambda_n \rho_n + \lambda_w \rho_w) g \nabla z].$$

Multiplying each phase velocity by the relative mobility of the other phase and subtracting the result we obtain:

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

Therefore, for \mathbf{v}_w we have

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

Using the velocity computed above, for the wetting phase we have:

$$\phi \frac{\partial(S_w)}{\partial t} + \nabla \cdot [f_w(\mathbf{v} + \lambda_n \Delta \rho g \nabla z)] = q_w - \nabla \cdot (f_w \lambda_n p_c \nabla S_w). \quad (21)$$

with $\Delta \rho = \rho_w - \rho_n$ and the fractional flow function f_w :

$$f_w = \frac{\lambda_w}{\lambda_n + \lambda_w},$$

is the fraction of the total flow that consists of the wetting fluid, and

$$\mathbf{v} = -\lambda (\nabla p_n - f_w \nabla p_w - (f_n \rho_n + f_w \rho_w) g \nabla z)$$

The coupling between the elliptic pressure equation and the parabolic saturation equation is much weaker than the coupling between the two continuity equations in the two-pressure formulation. In the pressure equation, the coupling to saturation appears explicitly in the effective mobility that makes up the variable coefficient in the Poisson problem and on the right-hand side through the phase mobilities and the derivative of the capillary function. In Equation (21), the saturation is only indirectly coupled to the pressure through the total Darcy velocity.

The system of PDEs can then be reformulated so that it consists of an elliptic equation for fluid pressure and one or more transport equations. These transport equations are generally parabolic, but have a strong hyperbolic character. Since the pressure and saturations equations have very different mathematical characteristics, it is natural to solve them in consecutive substeps. Sequential solution procedures and incompressible flow models are popular in academia and for research purposes, but are less used in industry. To the extent simulations are used for practical reservoir engineering, they are mainly based on compressible equations and solution procedures in which flow and transport are solved as a coupled system. Such approaches are very robust and particularly useful for problems with large variations in time constants or strong coupling between different types of flow mechanisms.

Sequential solution procedures

The two-phase, incompressible model will be solved using the fractional-flow formulation. This fractional flow model consists of an elliptic pressure equation

$$\nabla \cdot \mathbf{v} = q, \quad \mathbf{v} = -\lambda (\nabla p_n - f_w \nabla p_c - (f_n \rho_n + f_w \rho_w) g \nabla z) \quad (22)$$

and a parabolic transport equation (21)

$$\phi \frac{\partial(S_w)}{\partial t} + \nabla \cdot [f_w(\mathbf{v} + \lambda_n(\Delta \rho g \nabla z) + \nabla P_c)] = q_w. \quad (23)$$

Where the capillary pressure $p_c = p_w - p_n$ is assumed to be a known function P_c of the wetting saturation S_w . The transport equation becomes hyperbolic whenever P_c is zero.

In the standard sequential solution procedure, the system above, is evolved in time using a set of discrete time steps Δt_i . We assume that p , \mathbf{v} , and S_w are all known at time t and that we want to evolve the solution to time $t + \Delta t$.

At the beginning of the time step, we first assume that the saturation S_w is fixed. This means that the parameters λ , f_w , and f_n are functions of the spatial variable \mathbf{x} , and hence we can use this equation only to update the pressure p_n and the Darcy velocity \mathbf{v} . Then, \mathbf{v} and p_n are held fixed while equation 23 is evolved a time step Δt to define an updated saturation $S_w(\mathbf{x}, t + \Delta t)$. This saturation is then held fixed when we update p_n and \mathbf{v} in the next time step, and so on. Some authors refer to this solution procedure as an operator splitting method since the solution procedure effectively splits the overall solution operator of the flow model into two parts that are evolved in consecutive substeps. Likewise, some authors refer to the sequential solution procedure as IMPES, which is short-hand for implicit pressure, explicit saturation. Using the name IMPES is strictly speaking only correct if the saturation evolution is approximated by a single time step of an explicit transport solver.

Pressure solvers

The pressure Equation (22) for incompressible, multiphase flow is time dependent. This time dependence comes as the result of three factors:

1. K/μ being replaced by the total mobility $\lambda(S_w)$, which depends on time through the saturation $S_w(\mathbf{x}, t + \Delta t)$,
2. the constant density ρ being replaced by a saturation-dependent quantity $\rho_w f_w(S_w) + \rho_n f_n(S_w)$, and
3. the source term q being replaced by a saturation-dependent source term $q - \nabla \lambda_w(S_w) \nabla P_c(S_w)$.

However, once S_w is held fixed in time, all three quantities become functions of \mathbf{x} only, and we hence end up again with an elliptic Poisson-type equation having the same spatial variation.

Saturation solvers

The saturation equation depends on the time, using backward Euler discretization for the time derivative in Equation 23, we have:

$$\phi \frac{(S_w^{n+1} - S_w^n)}{\Delta t} + \nabla \cdot [f_w(S_w)(\mathbf{v} + \lambda_n(S_w)(\Delta \rho g \nabla z + \nabla P_c(S_w)))] = q_w, \quad (24)$$

or

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

which can be computed explicitly:

$$S_w^{n+1} = S_w^n - \mathcal{F}(S_w^n, S_w^n),$$

or implicitly:

$$S_w^{n+1} = S_w^n - \mathcal{F}(S_w^{n+1}, S_w^n).$$

If we use the implicit scheme, the system is nonlinear and depends on the saturation at time step n and $n + 1$.

The discretization is implemented with the following residual form for each cell Ω_i ,

$$F_i(s, r) = s_i - r_i + \frac{\Delta t}{\phi_i |\Omega_i|} [H_i(s) - \max(q_i, 0) - \min(q_i, 0) f(S_i)], \quad (25)$$

where s and r are cell-averaged quantities and subscript i refers to the cell the average is evaluated in. The sum of the interface fluxes for cell i

$$H_i(s) = \sum_k \frac{\lambda_w^u(s_i, s_k)}{\lambda_w^u(s_i, s_k) + \lambda_n^u(s_i, s_k)} [v_{ik} + \lambda_n^u(s_i, s_k)(g_{ik} + P_{ik})], \quad (26)$$

is computed using the single-point upstream mobility-weighting, whereas the fractional flow function f in the source term is evaluated from the cell average of S in cell Ω_i . The explicit scheme is given as $S^{n+1} = S^n - F(S^n, S^n)$ and the implicit solution is obtained solving $F(S^{n+1}, S^n) = 0$. The residual equations (25) for all cells in vector form

$$\mathbf{F}(\mathbf{s}) = \mathbf{s} - \mathbf{S} + \frac{\Delta t}{\phi_i |\Omega|} [\mathbf{H}(\mathbf{s}) - \mathbf{Q}^+ - \mathbf{Q}^- \mathbf{f}(\mathbf{S})] = \mathbf{0}. \quad (27)$$

Here, \mathbf{s} is the unknown state at time t_f and \mathbf{S} is the known state at the start of the time step. The nonlinear system can be solved with NR method, where, for the $(k + 1)$ -th iteration we have:

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

where $\mathbf{J}(\mathbf{S}^k) = \frac{\partial \mathbf{F}(\mathbf{S}^k; \mathbf{S}^n)}{\partial \mathbf{S}^k}$ is the Jacobian matrix, and $\delta \mathbf{S}^{k+1}$ is the NR update at iteration step $k + 1$.

Therefore, the linear system to solve is:

$$\mathbf{J}(\mathbf{pS}^k) \delta \mathbf{S}^{k+1} = \mathbf{b}(\mathbf{S}^k). \quad (28)$$

with $\mathbf{b}(\mathbf{S}^k)$ being the function evaluated at iteration step k , $\mathbf{b}(\mathbf{S}^k) = -\mathbf{F}(\mathbf{S}^k; \mathbf{S}^n)$.

3 Numerical experiments

3.1 Model problems

We model water flooding into a reservoir initially filled with oil. Therefore, the initial saturation is set as 1 for the oil and 0 for the water (see Figure 3, and Table 2). The solution of the systems of linear equations resulting from the discretization of partial differential equations for the description of this process is studied in this work. The solution of the system is performed with the Deflated Conjugate Gradient method preconditioned with Incomplete Cholesky (DICCG). We propose the use of snapshots and the snapshots-based POD vectors as deflation vectors for the above-mentioned method.

The matrices corresponding to the linear systems \mathbf{A} and right-hand sides \mathbf{b} are obtained with the Matlab Reservoir Simulation Toolbox (MRST) [?].

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 $\sigma_2 = 10^{-1}mD$ 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. And finally, we perform 3D experiments with 64 x 64 x 10 grid cells and gravity terms are included.

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

Table 1: Fluids properties.

Injection through the boundary

For the first set of experiments, water is injected from one boundary at a rate of $0.1 m^3/day$ 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 4800 days with a time step of 40 days (See Table 3).

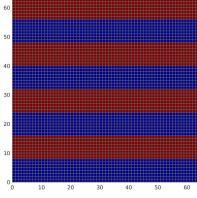


Figure 1: Rock permeability

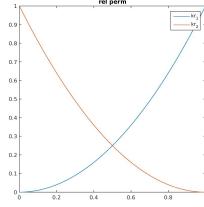


Figure 2: Fluid relative permeability

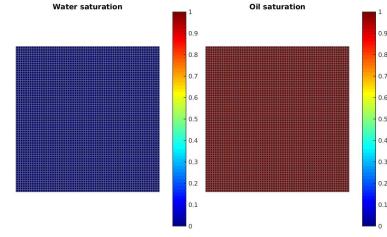


Figure 3: Initial saturation

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}	4800	days
dT	T/120	days
Case 1		
$Q_{x=0}$	0.1	m^3/day
$P_{0,x \neq 0, L_x}$	100	bars
$P_{x=L_x}$	0	bars
Case 2		
$Q_{y=0}$	0.1	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.

The solver

The solution of the system is approximated with the ICCG and DICCG methods.

For the DICCG method, it is necessary to select a set of deflation vectors. In the first series of experiments, the deflation vectors are the solutions of the system in the previous 10 time steps ($DICCG_{10}$). The snapshots of the first 10 time steps are obtained with ICCG. For the second set of experiments POD is used ($DICCG_{POD_{10}}$). In the first case 10 ($DICCG_{POD_{10}}$), and in the second 5 POD vectors ($DICCG_{POD_5}$) are used as deflation vectors. 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 10^{-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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 4 and 5 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	13109	DICCG ₁₀	950	1653	2603	20
10^1	13109	DICCG _{POD₁₀}	950	1655	2605	20
10^1	13109	DICCG _{POD₅}	950	1770	2720	21
10^2	13429	DICCG ₁₀	902	1475	2377	18
10^2	13429	DICCG _{POD₁₀}	902	1475	2377	18
10^2	13429	DICCG _{POD₅}	902	1499	2401	18
10^3	10612	DICCG ₁₀	602	1142	1744	16
10^3	10612	DICCG _{POD₁₀}	602	1142	1744	16
10^3	10612	DICCG _{POD₅}	602	1130	1732	16

Table 4: Comparison between the ICCG 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.

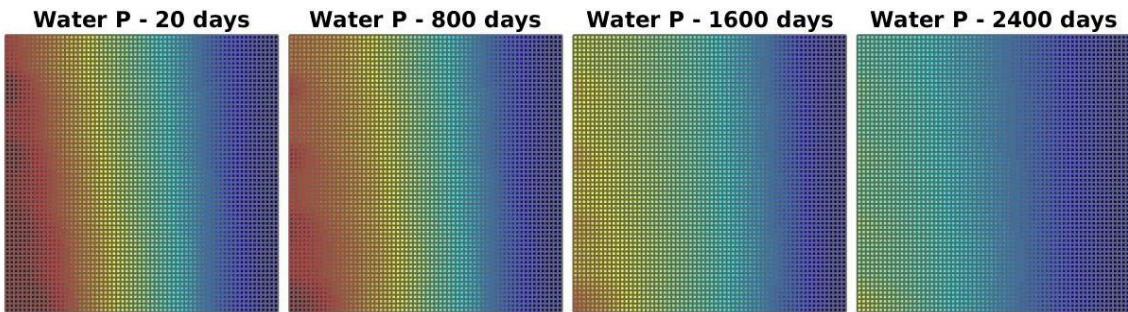


Figure 4: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

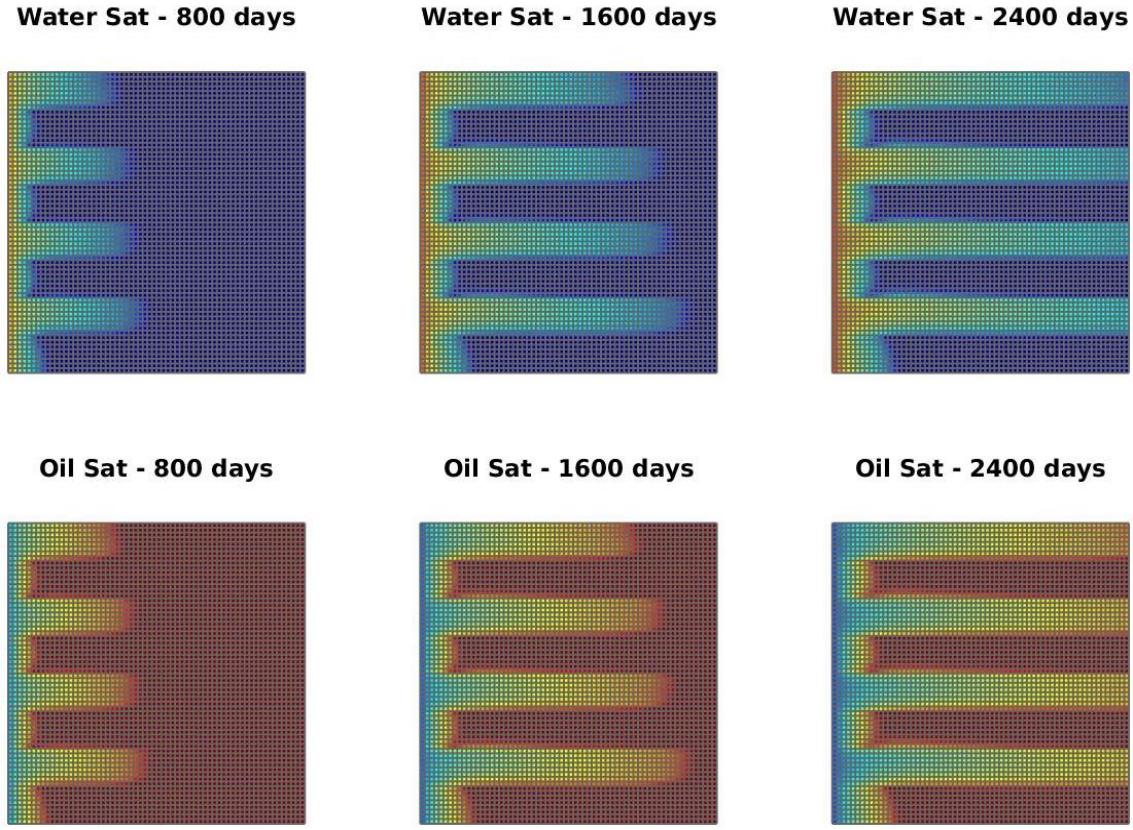


Figure 5: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

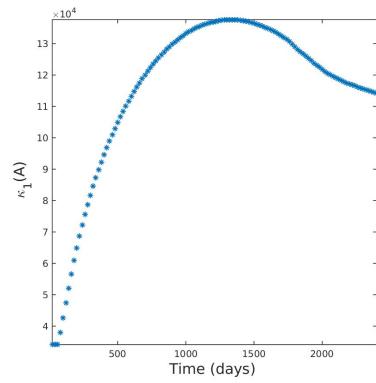


Figure 6: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

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 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 7 and 8 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	12712	DICCG ₁₀	949	1065	2014	16
10^1	12712	DICCG _{POD₁₀}	949	1065	2014	16
10^1	12712	DICCG _{POD₅}	949	1125	2074	16
10^2	12450	DICCG ₁₀	894	1049	1943	16
10^2	12450	DICCG _{POD₁₀}	894	1049	1943	16
10^2	12450	DICCG _{POD₅}	894	1059	1953	16
10^3	9790	DICCG ₁₀	600	893	1493	15
10^3	9790	DICCG _{POD₁₀}	600	893	1493	15
10^3	9790	DICCG _{POD₅}	600	839	1439	15

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.

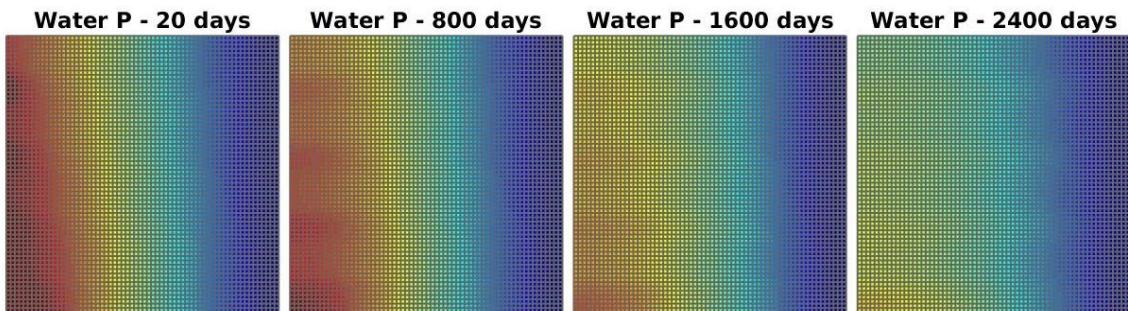


Figure 7: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells, capillary pressure included.

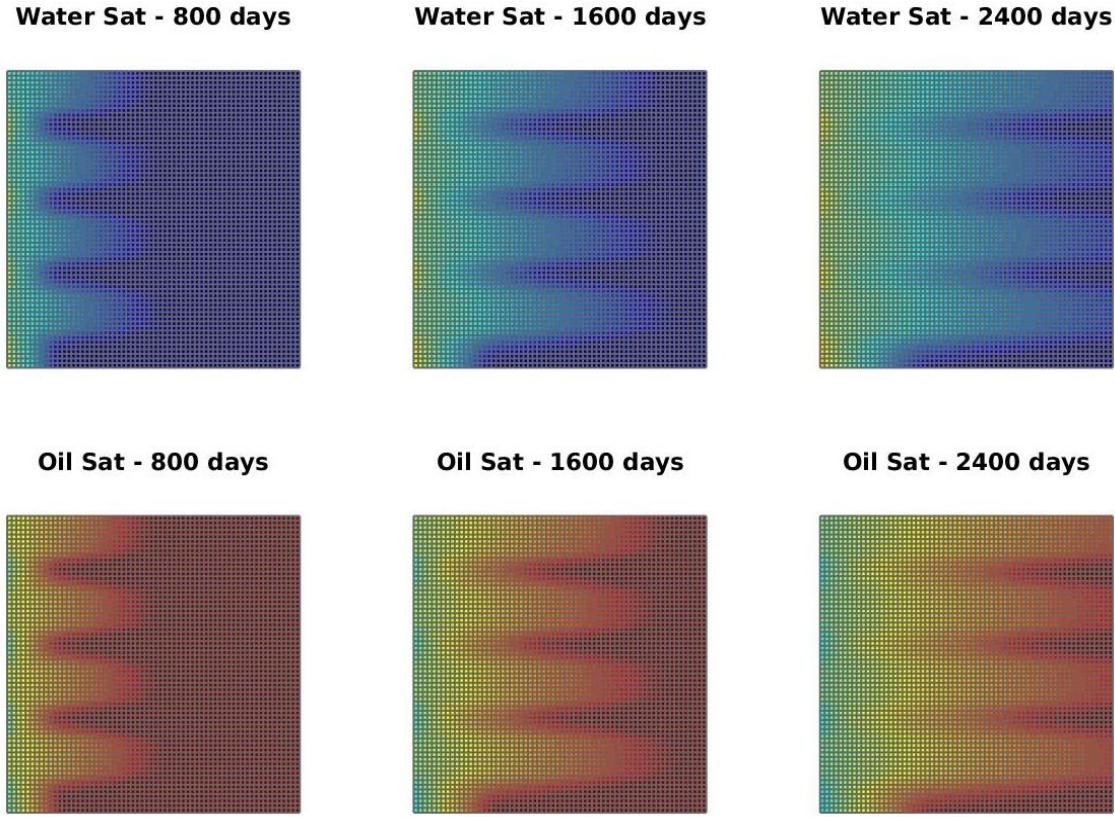


Figure 8: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells, capillary pressure included.

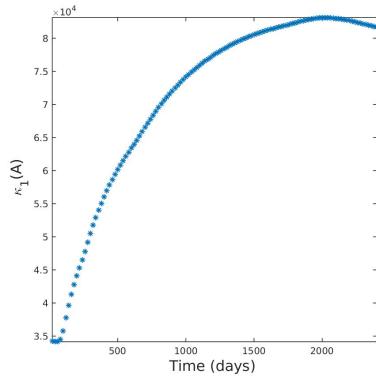


Figure 9: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

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

In Table 6 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 10 and 11 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	30662	DICCG ₁₀	1210	616	1826	6
10^1	30662	DICCG _{POD₁₀}	1210	482	1692	6
10^1	30662	DICCG _{POD₅}	1210	590	1800	6
10^2	33843	DICCG ₁₀	1195	669	1864	6
10^2	33843	DICCG _{POD₁₀}	1195	682	1877	6
10^2	33843	DICCG _{POD₅}	1195	795	1990	6
10^3	27144	DICCG ₁₀	695	856	1551	6
10^3	27144	DICCG _{POD₁₀}	695	897	1592	6
10^3	27144	DICCG _{POD₅}	695	1008	1703	6

Table 6: 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.

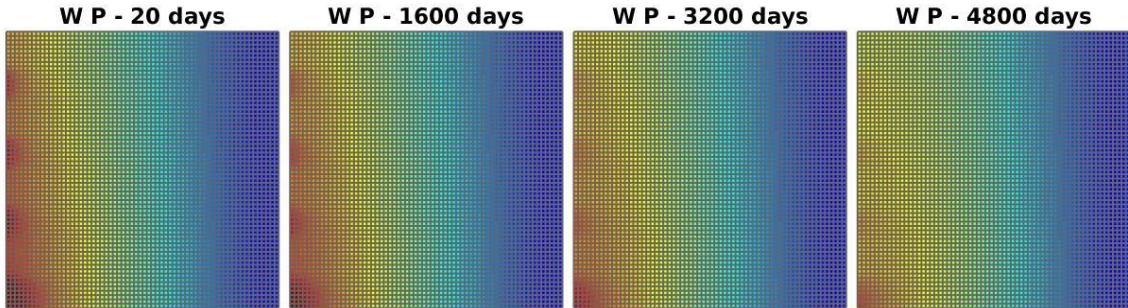


Figure 10: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 x 10 grid cells, no capillary pressure, gravity included.

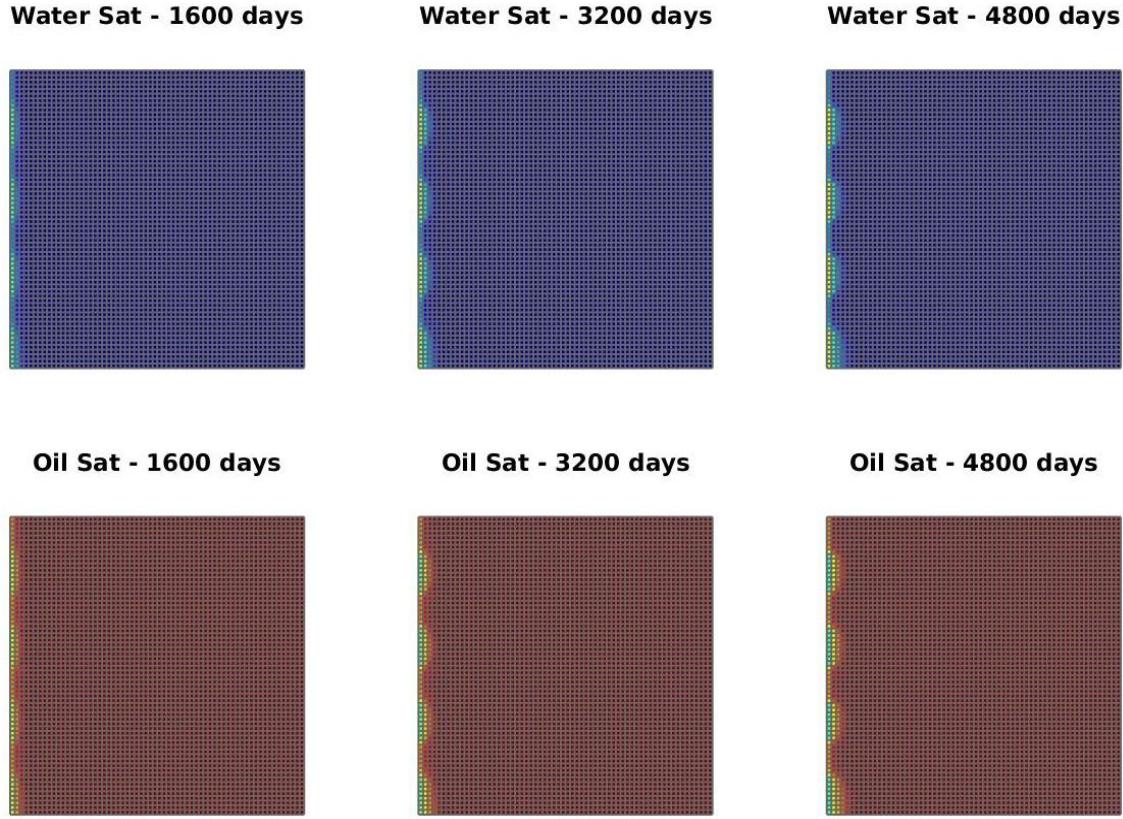


Figure 11: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, no capillary pressure, gravity included.

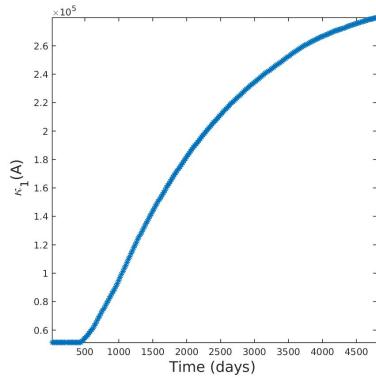


Figure 12: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells.

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

In Table 7 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 13 and 14 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	29667	DICCG ₁₀	1210	324	1534	5
10^1	29667	DICCG _{POD₁₀}	1210	328	1538	5
10^1	29667	DICCG _{POD₅}	1210	378	1588	5
10^2	29452	DICCG ₁₀	1196	511	1707	6
10^2	29452	DICCG _{POD₁₀}	1196	512	1708	6
10^2	29452	DICCG _{POD₅}	1196	597	1793	6
10^3	24285	DICCG ₁₀	719	831	1550	6
10^3	24285	DICCG _{POD₁₀}	719	830	1549	6
10^3	24285	DICCG _{POD₅}	719	858	1577	6

Table 7: 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.

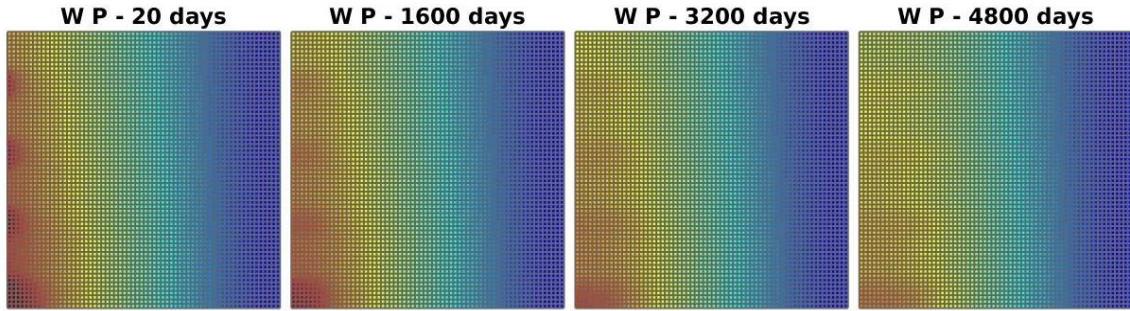


Figure 13: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 x 10 grid cells, capillary pressure and gravity included.

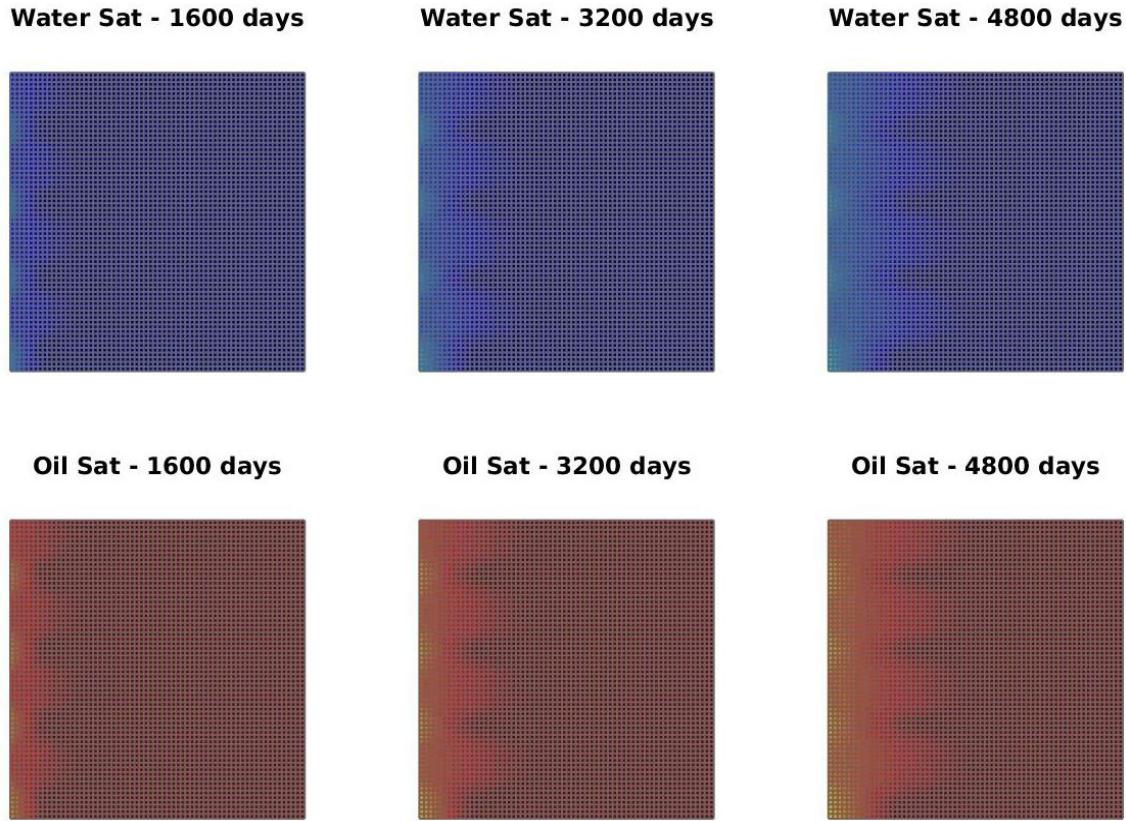


Figure 14: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, capillary pressure and gravity included.

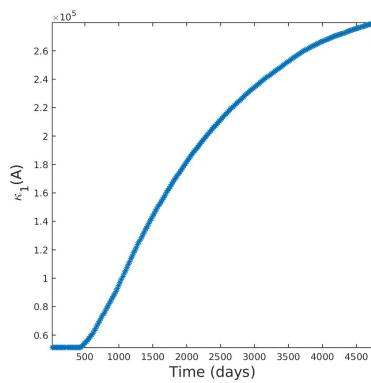


Figure 15: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, capillary pressure and gravity included.

Case 2: Injection through the bottom boundary, no capillary pressure.

In Table 8 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 16 and 17 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	10209	DICCG ₁₀	818	1098	1916	19
10^1	10209	DICCG _{POD₁₀}	818	1072	1890	19
10^1	10209	DICCG _{POD₅}	818	1090	1908	19
10^2	13481	DICCG ₁₀	1092	1175	2267	17
10^2	13481	DICCG _{POD₁₀}	1092	1192	2284	17
10^2	13481	DICCG _{POD₅}	1092	1122	2214	16
10^3	14693	DICCG ₁₀	1182	1260	2442	17
10^3	14693	DICCG _{POD₁₀}	1182	1315	2497	17
10^3	14693	DICCG _{POD₅}	1182	1041	2223	15

Table 8: Comparison between the ICCG and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the bottom boundary, domain 64 x 64 cells.

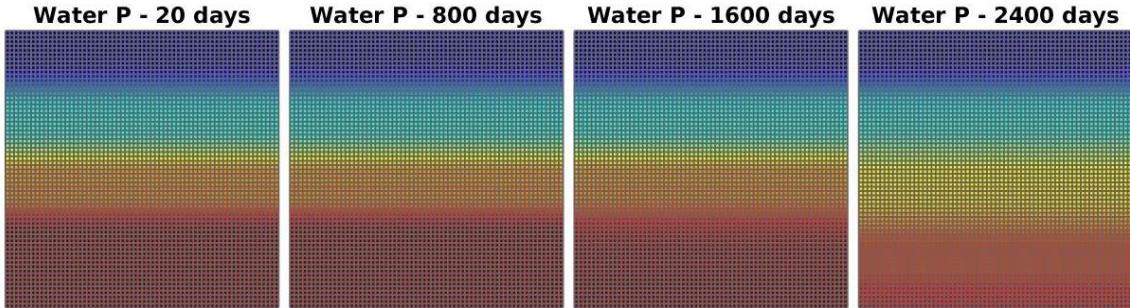


Figure 16: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

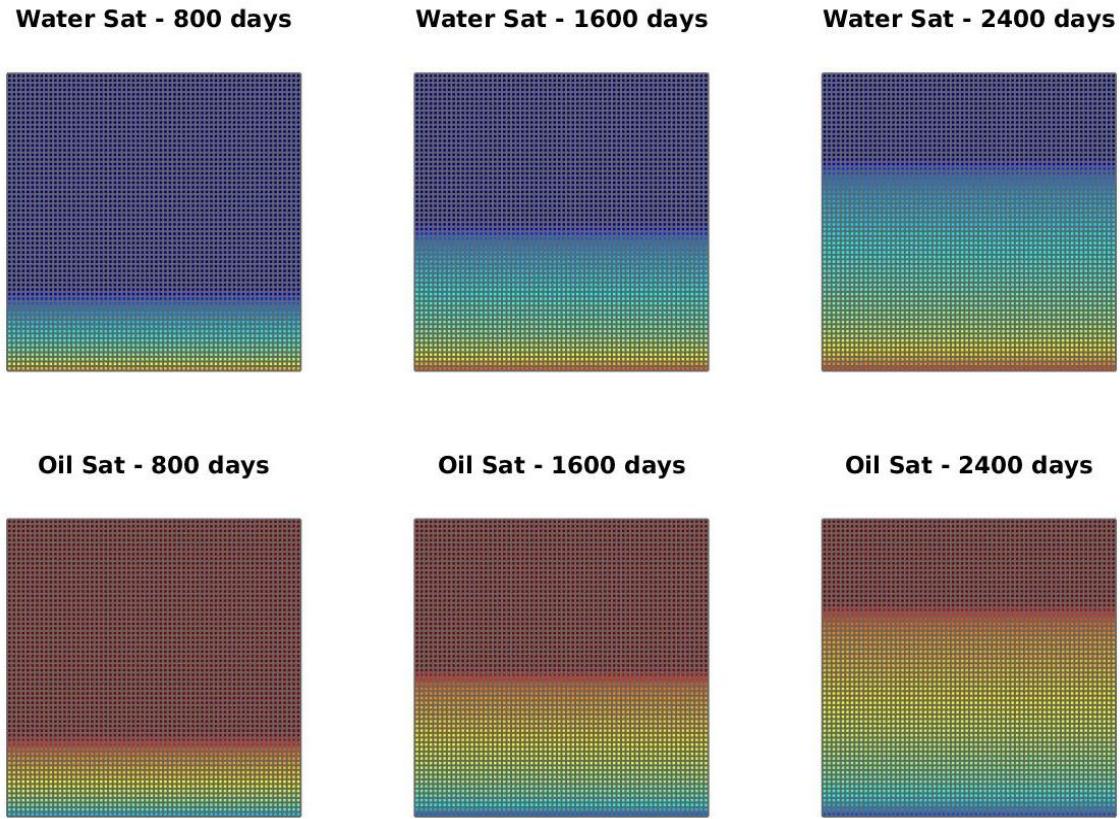


Figure 17: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

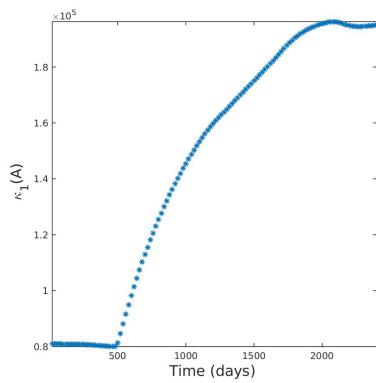


Figure 18: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

Case 2 A: Injection through the bottom boundary, capillary pressure included.

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 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 19 and 20 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	10710	DICCG ₁₀	824	1132	1956	18
10^1	10710	DICCG _{POD₁₀}	824	1157	1981	18
10^1	10710	DICCG _{POD₅}	824	1022	1846	17
10^2	13459	DICCG ₁₀	1092	1076	2168	16
10^2	13459	DICCG _{POD₁₀}	1092	1087	2179	16
10^2	13459	DICCG _{POD₅}	1092	1004	2096	16
10^3	14678	DICCG ₁₀	1182	1254	2436	17
10^3	14678	DICCG _{POD₁₀}	1182	1236	2418	16
10^3	14678	DICCG _{POD₅}	1182	943	2125	14

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

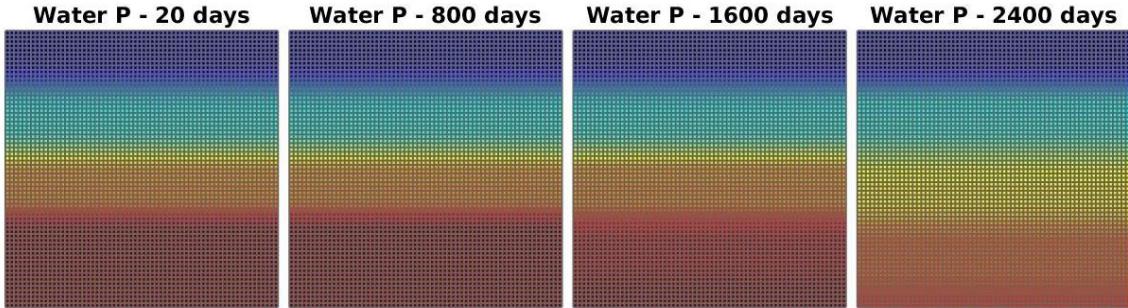


Figure 19: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells, capillary pressure included.

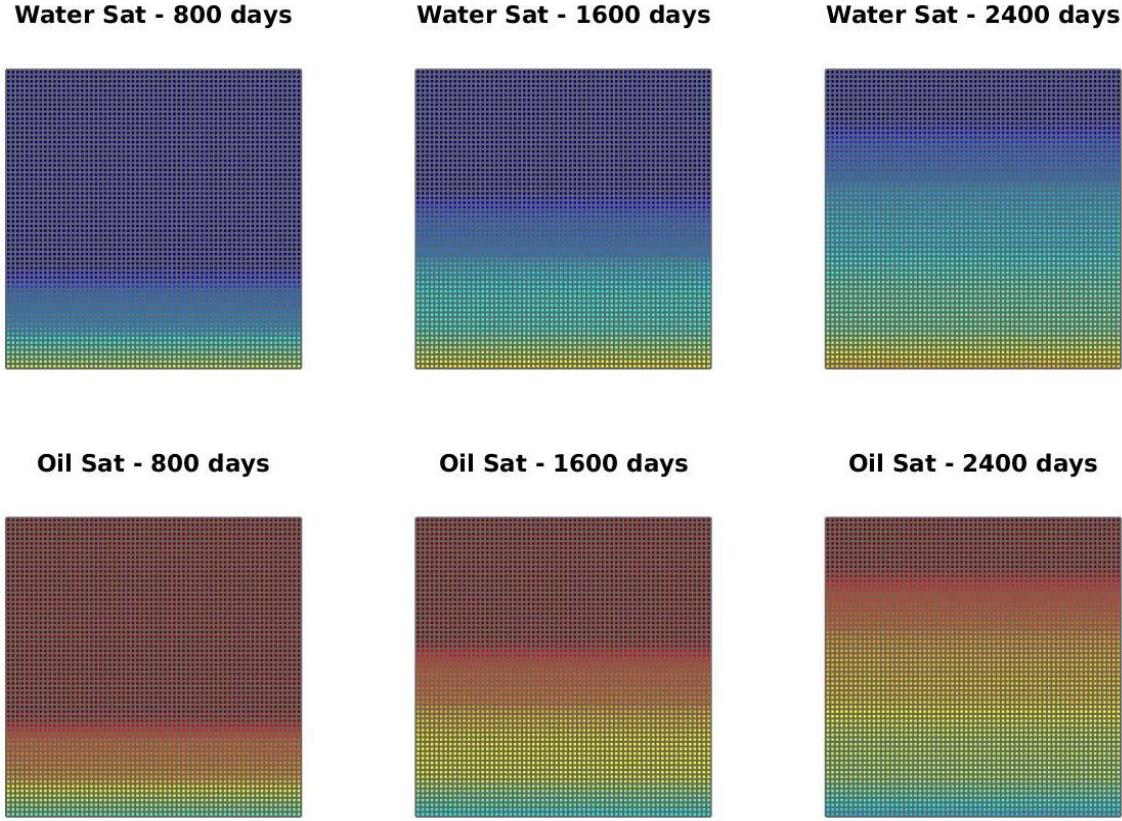


Figure 20: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells, capillary pressure included.

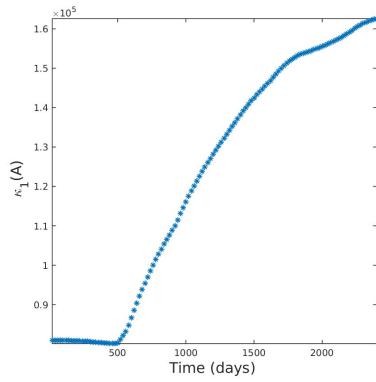


Figure 21: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

Case 2 B: Injection through the bottom boundary, no capillary pressure, gravity included (3D).

In Table 10 the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selec-

tion 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 22 and 23 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	27558	DICCG ₁₀	1140	1034	2174	8
10^1	27558	DICCG _{POD₁₀}	1140	331	1471	5
10^1	27558	DICCG _{POD₅}	1140	385	1525	6
10^2	35477	DICCG ₁₀	1454	413	1867	5
10^2	35477	DICCG _{POD₁₀}	1454	430	1884	5
10^2	35477	DICCG _{POD₅}	1454	398	1852	5
10^3	38736	DICCG ₁₀	1522	543	2065	5
10^3	38736	DICCG _{POD₁₀}	1522	496	2018	5
10^3	38736	DICCG _{POD₅}	1522	477	1999	5

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

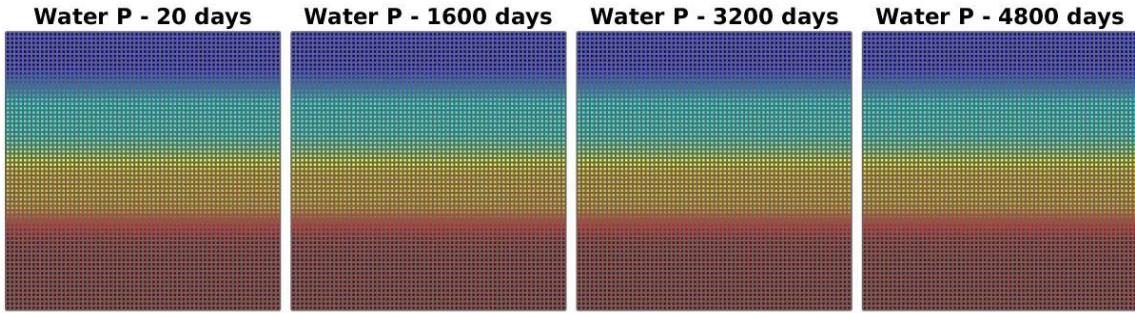


Figure 22: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 x 10 grid cells, no capillary pressure, gravity included.

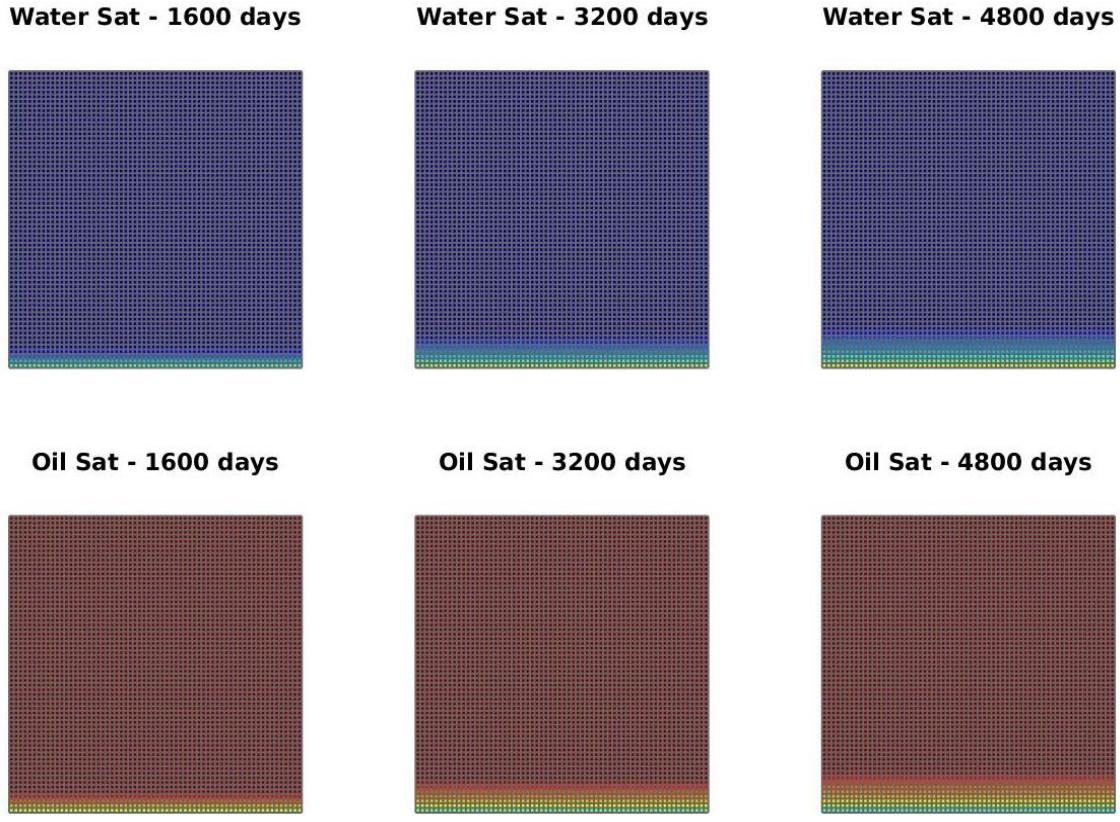


Figure 23: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, no capillary pressure, gravity included.

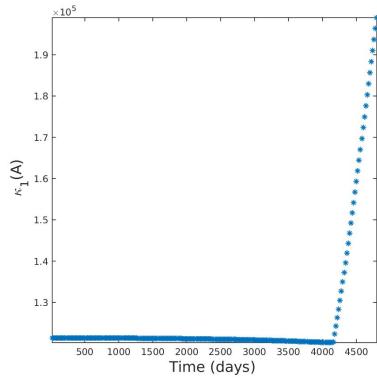


Figure 24: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells.

Case 2 C: Injection through the bottom boundary, capillary pressure and gravity included (3D).

In Table 11 the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selec-

tion 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 25 and 26 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	27518	DICCG ₁₀	1140	806	1946	7
10^1	27518	DICCG _{POD₁₀}	1140	403	1543	6
10^1	27518	DICCG _{POD₅}	1140	486	1626	6
10^2	35740	DICCG ₁₀	1451	563	2014	6
10^2	35740	DICCG _{POD₁₀}	1451	580	2031	6
10^2	35740	DICCG _{POD₅}	1451	580	2031	6
10^3	38706	DICCG ₁₀	1527	506	2033	5
10^3	38706	DICCG _{POD₁₀}	1527	517	2044	5
10^3	38706	DICCG _{POD₅}	1527	576	2103	5

Table 11: Comparison between the ICCC and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the bottom boundary, domain 64 x 64 x 10 cells, capillary pressure and gravity included.

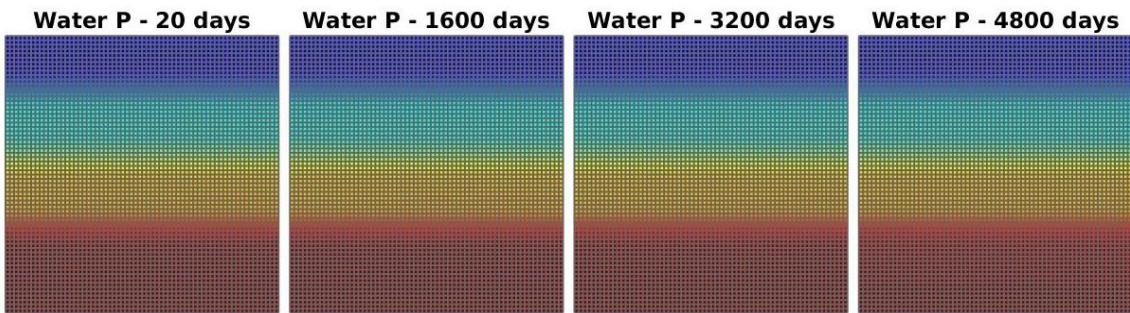


Figure 25: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells, capillary pressure and gravity included.

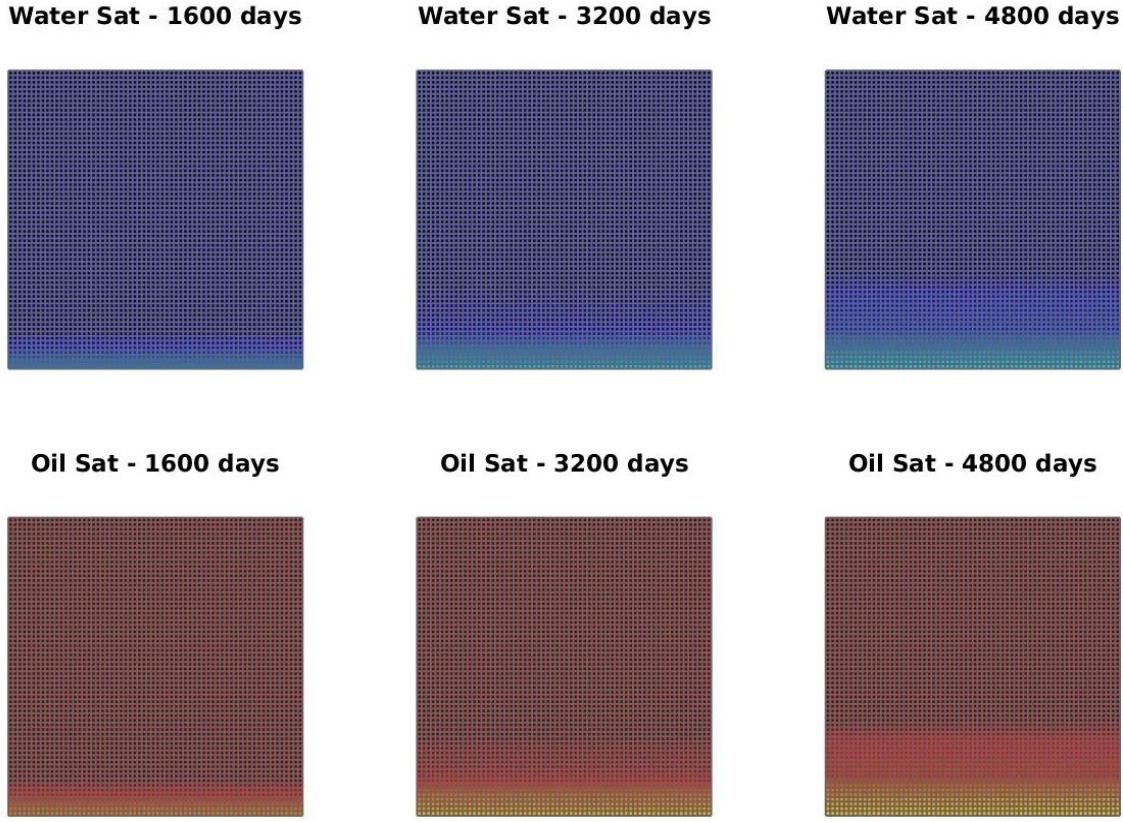


Figure 26: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, capillary pressure and gravity included.

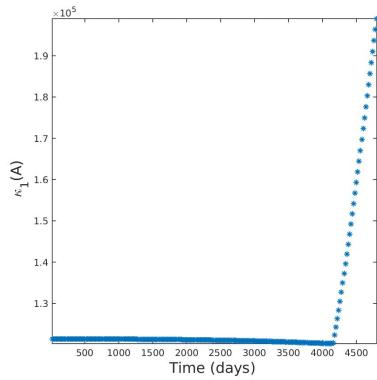


Figure 27: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, capillary pressure and gravity included.

Case 3: Two wells with a pre-set pressure, no capillary pressure, no gravity included.

In this experiments we set Newman boundary conditions in all the boundaries, and we add two wells with prescribed pressure in two opposite corners of the reservoir. The values of

the pressure are presented in Table 12.

Well	Water Sat	Oil Sat	Pressure
W_1	1	0	100 bars
W_2	0	1	0 bars

Table 12: Wells properties.

Property	Value	Units
T_{total}	450	days
dT	T/60	days

Table 13: Initial values of the system.

In Table 14 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 28 and 29 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	5118	DICCG ₁₀	866	95	961	19
10^1	5118	DICCG _{POD₁₀}	866	93	959	19
10^1	5118	DICCG _{POD₅}	866	116	982	19
10^2	7392	DICCG ₁₀	1240	57	1297	18
10^2	7392	DICCG _{POD₁₀}	1240	57	1297	18
10^2	7392	DICCG _{POD₅}	1240	67	1307	18
10^3	7920	DICCG ₁₀	1320	52	1372	17
10^3	7920	DICCG _{POD₁₀}	1320	52	1372	17
10^3	7920	DICCG _{POD₅}	1320	53	1373	17

Table 14: Comparison between the ICCG and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the bottom boundary, domain 64 x 64 cells.

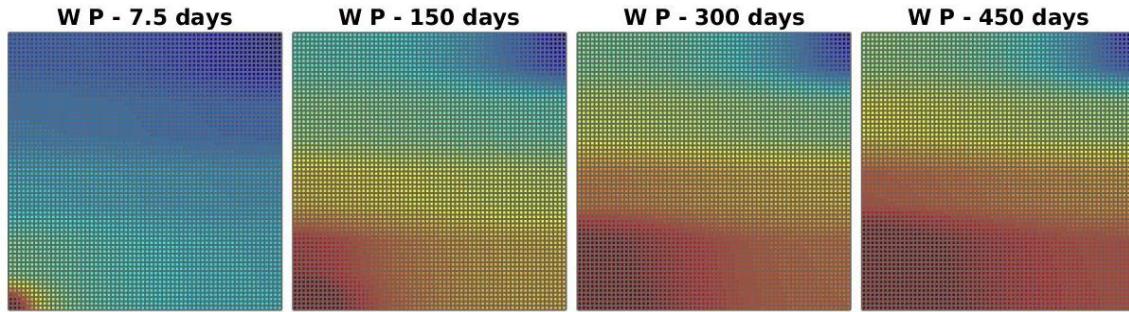


Figure 28: Pressure field for various times for a contrast between permeability values of 10^1 , 64×64 grid cells.

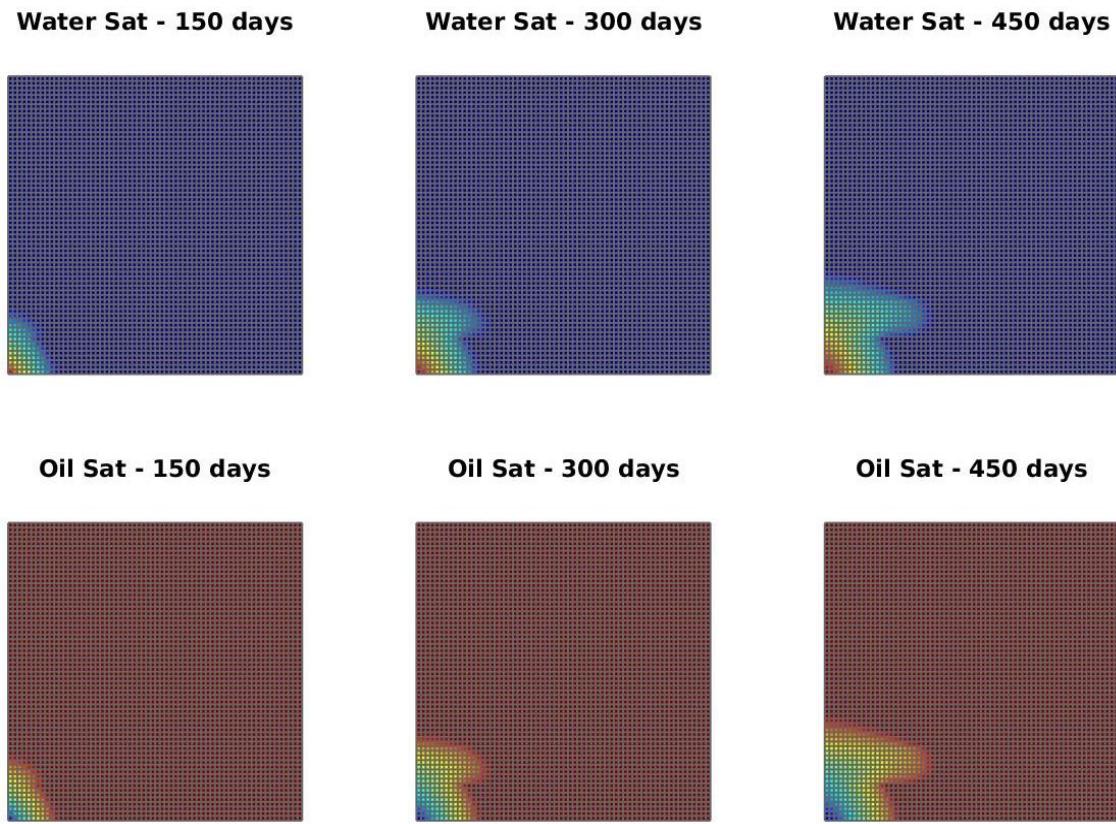


Figure 29: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , 64×64 grid cells.

Case 3 A: Two wells with a pre-set pressure, capillary pressure included.

In Table 15 the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selec-

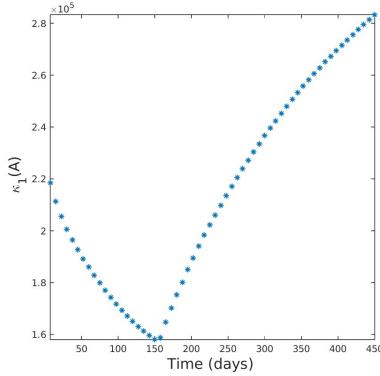


Figure 30: Estimated condition number of the matrix \mathbf{A} for a contrast between permeability values of 10^1 , 64×64 grid cells.

tion 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 31 and 32 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	5176	DICCG ₁₀	875	94	969	19
10^1	5176	DICCG _{POD₁₀}	875	95	970	19
10^1	5176	DICCG _{POD₅}	875	100	975	19
10^2	7620	DICCG ₁₀	1270	74	1344	18
10^2	7620	DICCG _{POD₁₀}	1270	74	1344	18
10^2	7620	DICCG _{POD₅}	1270	71	1341	18
10^3	8345	DICCG ₁₀	1357	64	1421	17
10^3	8345	DICCG _{POD₁₀}	1357	63	1420	17
10^3	8345	DICCG _{POD₅}	1357	67	1424	17

Table 15: Comparison between the ICCG and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the bottom boundary, domain 64×64 cells, capillary pressure included.

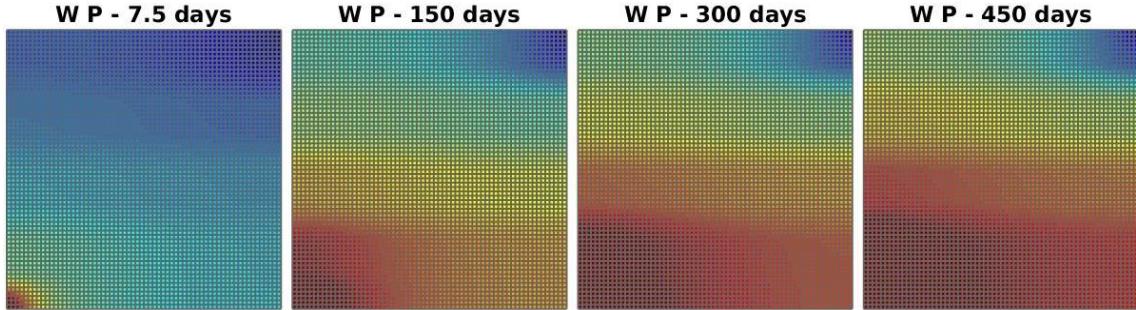


Figure 31: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells, capillary pressure included.

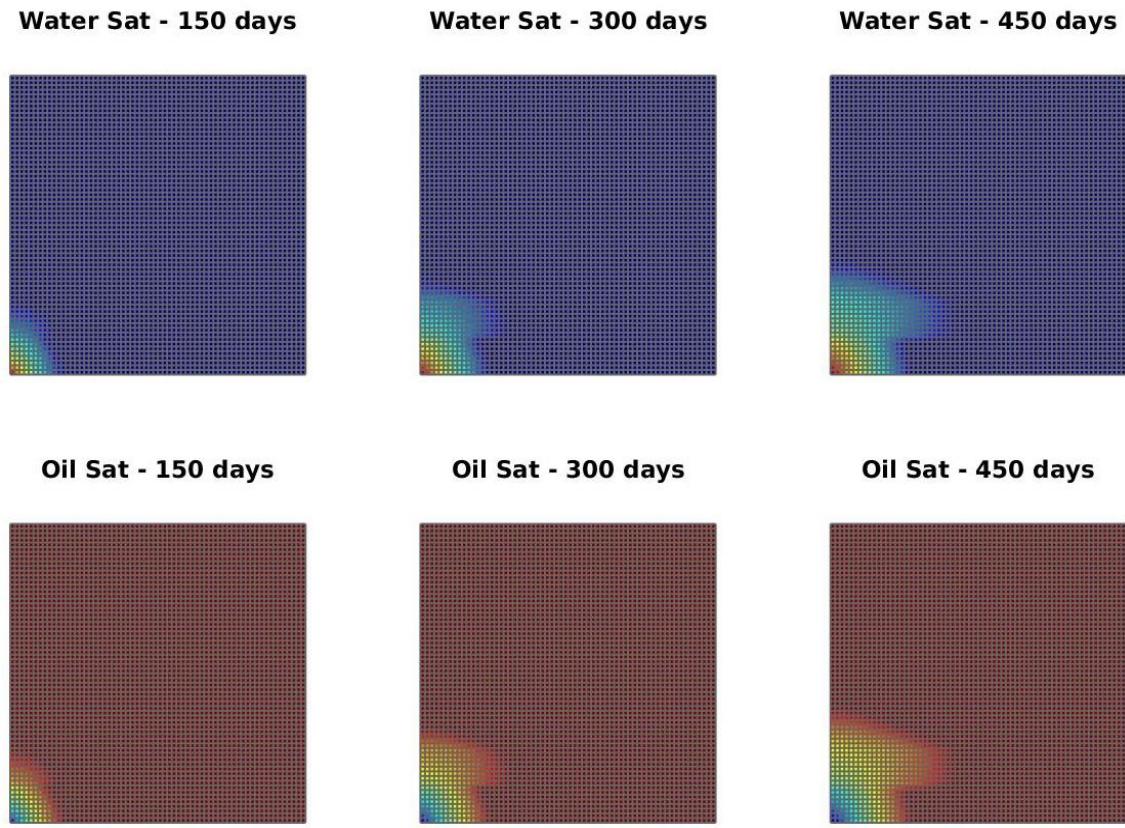


Figure 32: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells, capillary pressure included.

Case 3 B: Two wells with a pre-set pressure, no capillary pressure, gravity included (3D).

In Table 16 the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selec-

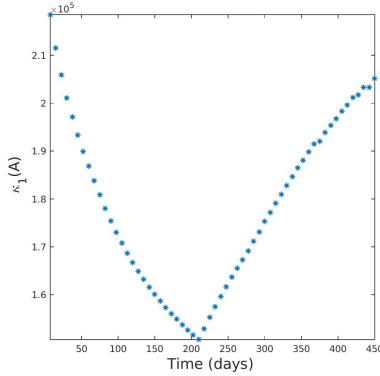


Figure 33: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , 64×64 grid cells.

tion 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 34 and 35 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	6923	DICCG ₁₀	1172	109	1281	19
10^1	6923	DICCG _{POD₁₀}	1172	109	1281	19
10^1	6923	DICCG _{POD₅}	1172	137	1309	19
10^2	10182	DICCG ₁₀	1706	67	1773	17
10^2	10182	DICCG _{POD₁₀}	1706	70	1776	17
10^2	10182	DICCG _{POD₅}	1706	76	1782	18
10^3	7920	DICCG ₁₀	1320	52	1372	17
10^3	7920	DICCG _{POD₁₀}	1320	52	1372	17
10^3	7920	DICCG _{POD₅}	1320	53	1373	17

Table 16: Comparison between the ICCG and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the bottom boundary, domain $64 \times 64 \times 10$ cells, no capillary pressure, gravity included.

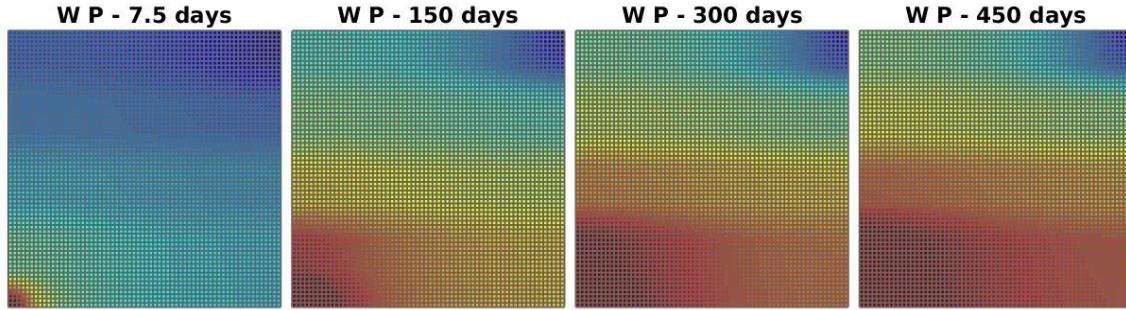


Figure 34: Pressure field for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, no capillary pressure, gravity included.

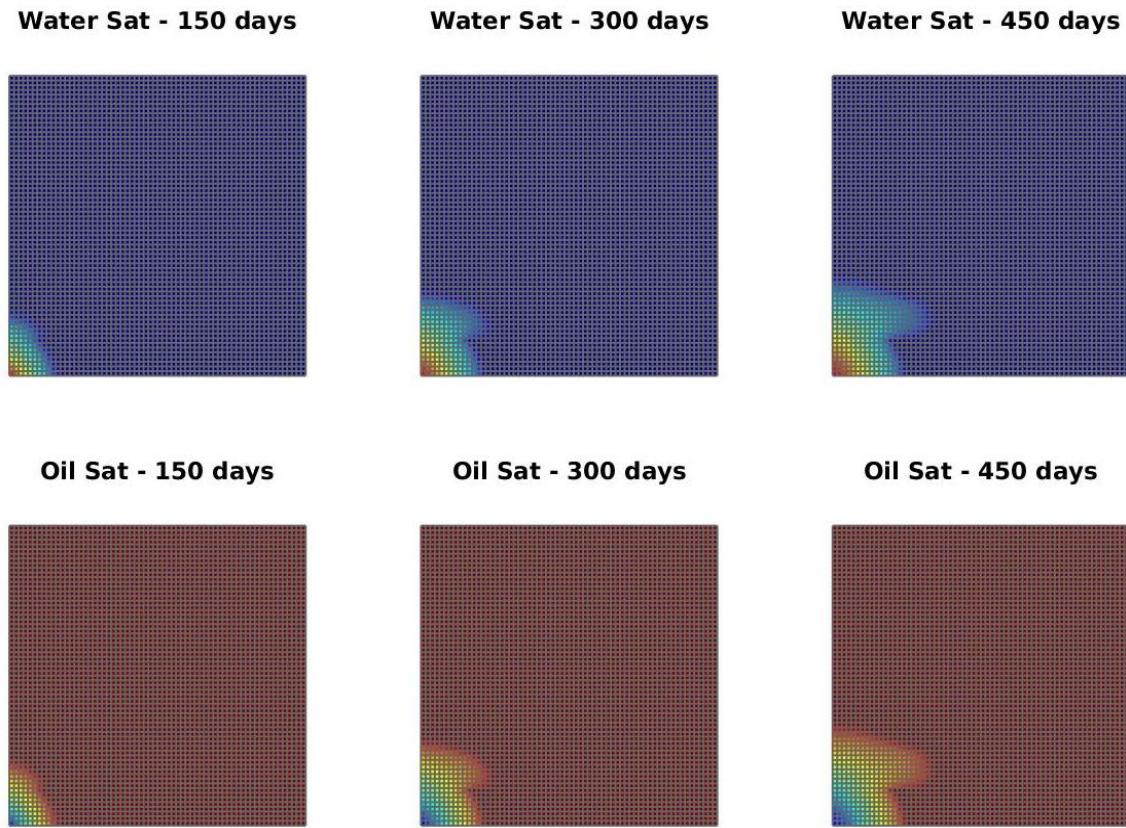


Figure 35: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, no capillary pressure, gravity included.

Case 3 C: Two wells with a pre-set pressure, capillary pressure and gravity included (3D).

In Table 17 the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selec-

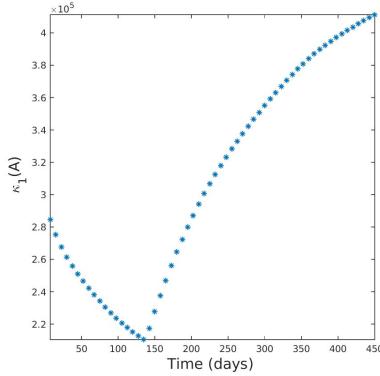


Figure 36: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells.

tion 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 46 and 47 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	6926	DICCG ₁₀	1173	132	1305	19
10^1	6926	DICCG _{POD₁₀}	1173	132	1305	19
10^1	6926	DICCG _{POD₅}	1173	150	1323	19
10^2	10200	DICCG ₁₀	1710	68	1778	17
10^2	10200	DICCG _{POD₁₀}	1710	68	1778	17
10^2	10200	DICCG _{POD₅}	1710	74	1784	17
10^3	11184	DICCG ₁₀	1847	2553	4400	39
10^3	11184	DICCG _{POD₁₀}	1847	65	1912	17
10^3	11184	DICCG _{POD₅}	1847	65	1912	17

Table 17: Comparison between the ICCG and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the bottom boundary, domain $64 \times 64 \times 10$ cells, capillary pressure and gravity included.

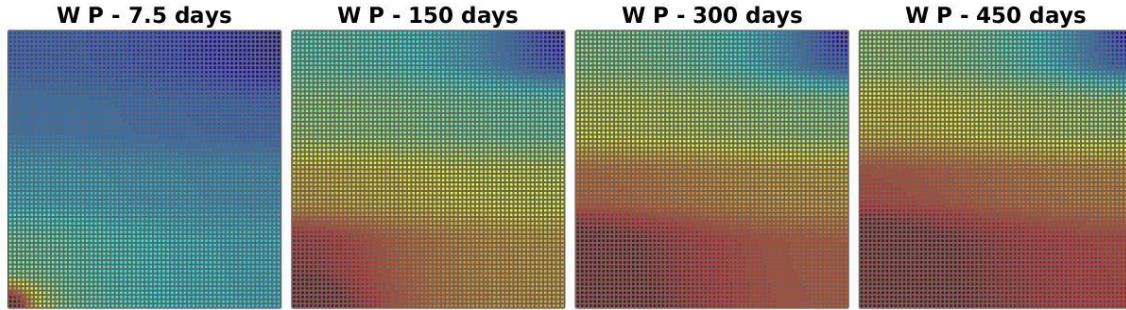


Figure 37: Pressure field for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, capillary pressure and gravity included.

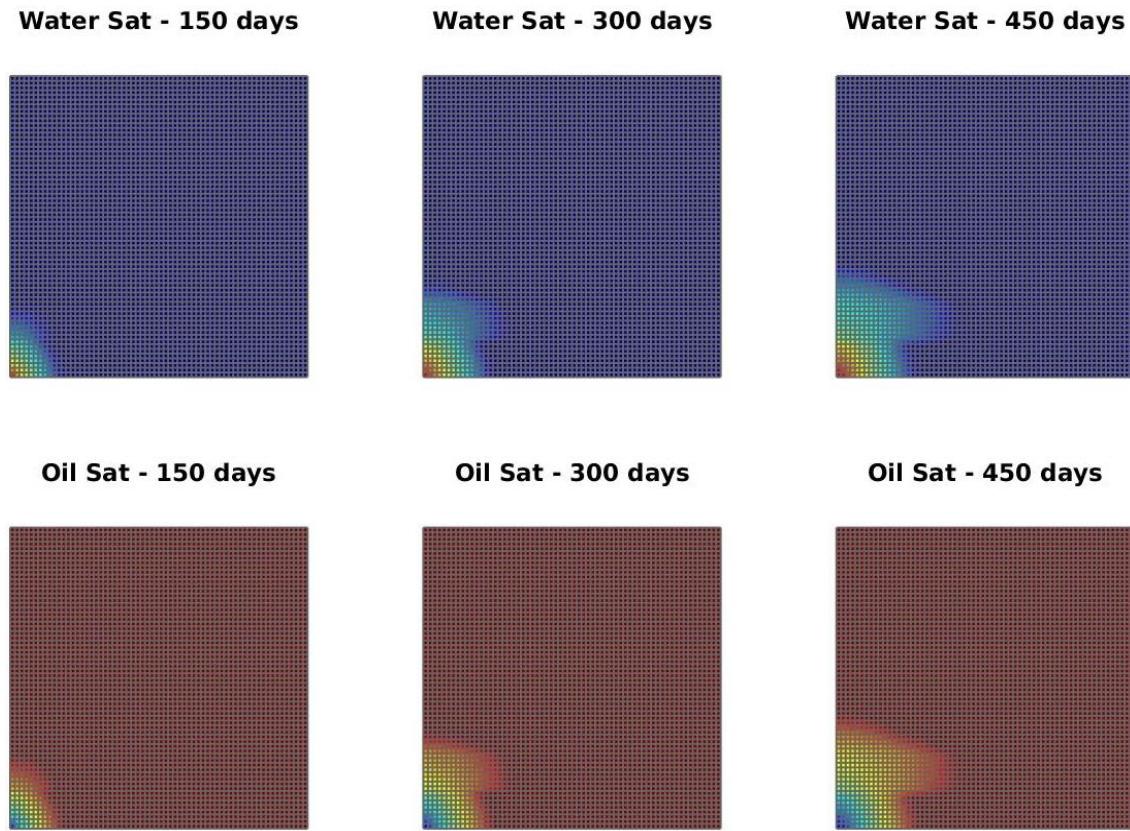


Figure 38: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, capillary pressure and gravity included.

Case 4: Two wells with a pre-set pressure, capillary pressure, gravity included (3D).

In this experiments we set Newman boundary conditions in all the boundaries, and we add two wells with prescribed pressure in two opposite corners of the reservoir. The values of

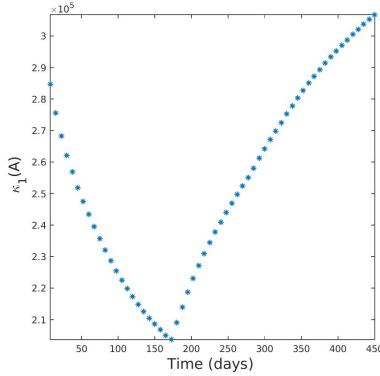


Figure 39: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , $64 \times 64 \times 10$ grid cells, capillary pressure and gravity included.

Property	Water	Oil	Units
μ	1	10	cp
ρ	1000	700	kg/m^3
k_r	$(S_w)^3$	$(1 - S_w)^{2.5}$	

Table 18: Fluids properties.

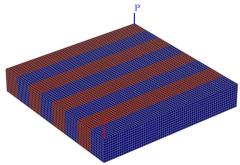


Figure 40: Rock permeability

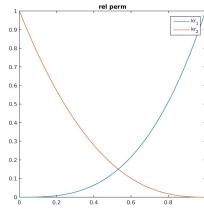


Figure 41: Fluid relative permeability

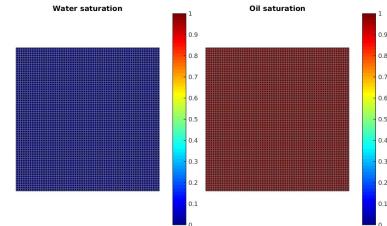


Figure 42: Initial saturation

the pressure are presented in Table 12.

Well	Water Sat	Oil Sat	Pressure
W_1	1	0	100 bars
W_2	0	1	0 bars

Table 19: Wells properties.

Property	Value	Units
T_{total}	60	days
dT	$T/60$	days

Table 20: Initial values of the system.

In Table 21 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 43 and 44 for various times.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
10^1	7164	DICCG ₁₀	1271	72	1343	19
10^1	7164	DICCG _{POD₁₀}	1271	72	1343	19
10^1	7164	DICCG _{POD₅}	1271	88	1359	19

Table 21: Comparison between the ICCC and DICCG methods of the average number of linear iterations for various contrast between permeability layers. Injection through the bottom boundary, domain 64 x 64 cells.

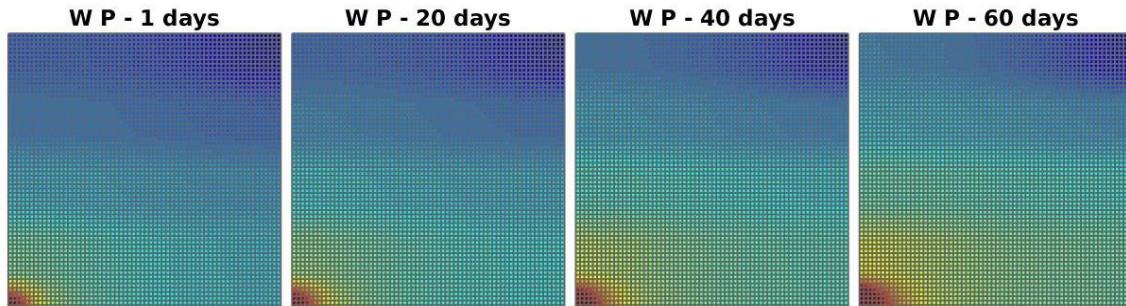


Figure 43: Pressure field for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

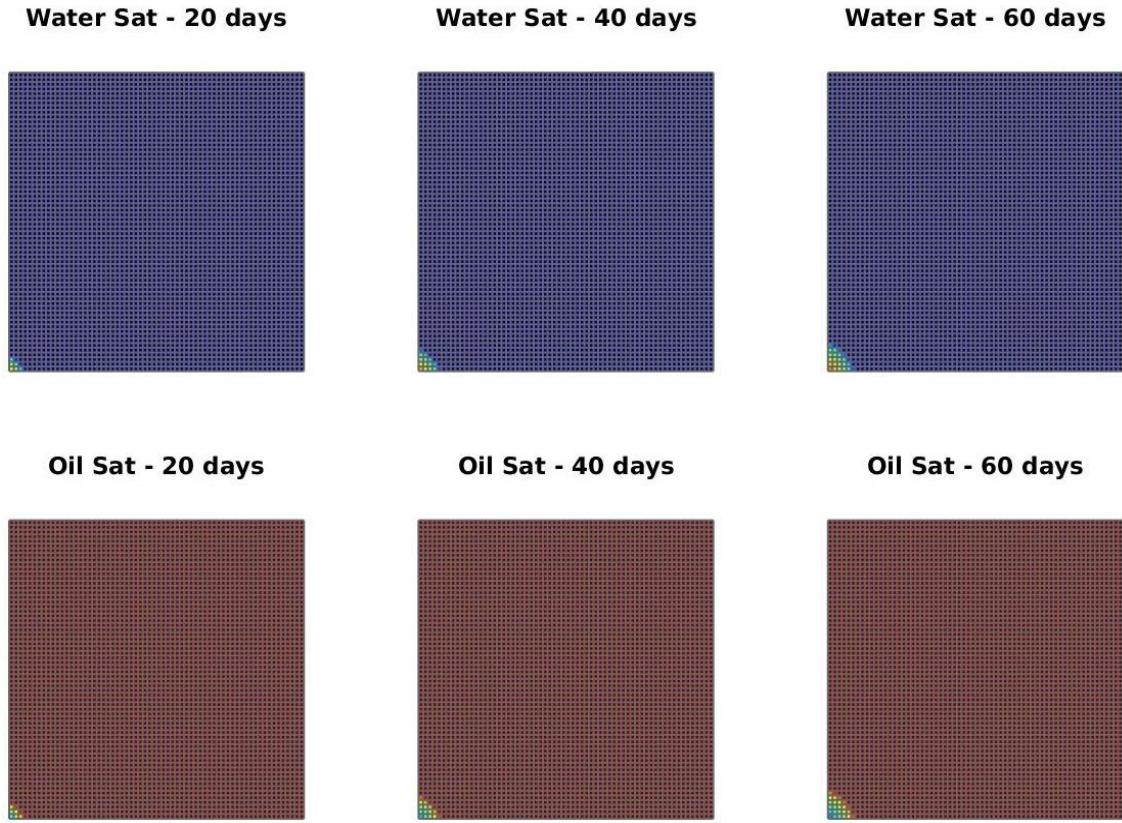


Figure 44: Saturations of oil and water for various times for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

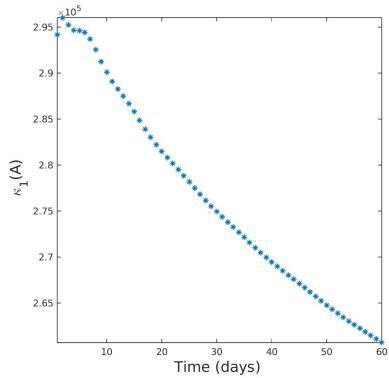


Figure 45: Estimated condition number of the matrix A for a contrast between permeability values of 10^1 , 64 x 64 grid cells.

SPE 10, 10 layers model (60 x 220 x 10 cells)

In Table 17 the number of iterations necessary to achieve convergence are presented for various contrast between permeability layers for the deflation method with different selec-

tion 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 of iterations of the DICCG methods with respect to the ICCG method is presented. The pressure field and the saturations for oil and water are presented in Figures 46 and 47 for various times.

Cp	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG +DICCG	% of total ICCG
No	15276	DICCG ₁₀	2631	149	2780	18
No	15276	DICCG _{POD₁₀}	2631	148	2779	18
No	15276	DICCG _{POD₅}	2631	171	2802	18
	15350	DICCG ₁₀	2642	188	2830	18
	15350	DICCG _{POD₁₀}	2642	190	2832	18
	15350	DICCG _{POD₅}	2642	211	2853	19

Table 22: Comparison between the ICCC and DICCG methods of the average number of linear iterations for the SPE 10 benchmark with 60 x 220 x 10 cells. Two wells, no capillary pressure, gravity included.

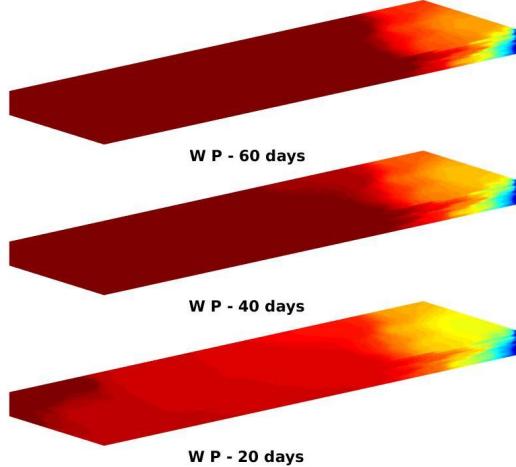


Figure 46: Pressure field for various times for the SPE 10 benchmark with 60 x 220 x 10 cells. Two wells, no capillary pressure, gravity included.

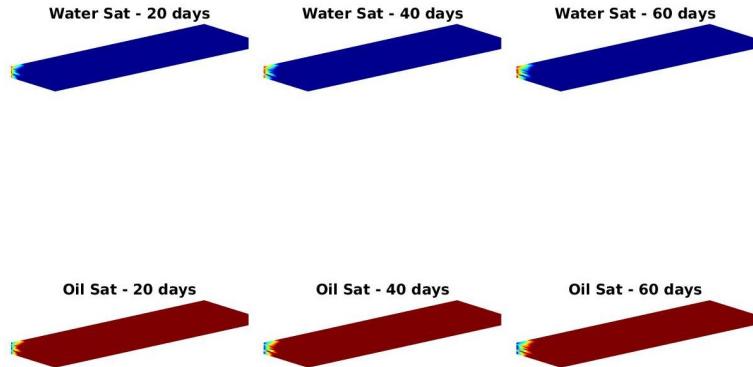


Figure 47: Saturations of oil and water for various times for the SPE 10 benchmark with $60 \times 220 \times 10$ cells. Two wells, no capillary pressure, gravity included.

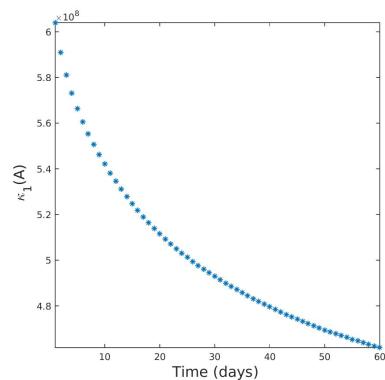


Figure 48: Estimated condition number of the matrix A for the SPE 10 benchmark with $60 \times 220 \times 10$ cells. Two wells, no capillary pressure, gravity included.

Eigenvalues analysis

For the eigenvalue analysis, we studied a grid containing 27 x 27 x 1 cells, with the same characteristics as in Case 1.

$\frac{\sigma_2}{\sigma_1}$	Total ICCG	Method	ICCG Snapshots	DICCG	Total ICCG	% of total ICCG
10^1	4980	DICCG ₁₀	374	465	839	17
10^1	4980	DICCG _{POD₁₀}	374	464	838	17
10^1	4980	DICCG _{POD₅}	374	565	939	19
10^3	3967	DICCG ₁₀	226	375	601	15
10^3	3967	DICCG _{POD₁₀}	226	375	601	15
10^3	3967	DICCG _{POD₅}	226	415	641	16
10^5	2918	DICCG ₁₀	175	397	572	20
10^5	2918	DICCG _{POD₁₀}	175	400	575	20
10^5	2918	DICCG _{POD₅}	175	475	650	22
10^7	2434	DICCG ₁₀	172	1105	1277	52
10^7	2434	DICCG _{POD₁₀}	172	472	644	26
10^7	2434	DICCG _{POD₅}	172	520	692	28
capillary pressure						
10^1	4819	DICCG ₁₀	373	467	840	17
10^1	4819	DICCG _{POD₁₀}	373	467	840	17
10^1	4819	DICCG _{POD₅}	373	480	853	18
10^3	3389	DICCG ₁₀	228	345	573	17
10^3	3389	DICCG _{POD₁₀}	228	345	573	17
10^3	3389	DICCG _{POD₅}	228	334	562	17
10^5	2908	DICCG ₁₀	172	404	576	20
10^5	2908	DICCG _{POD₁₀}	172	400	572	20
10^5	2908	DICCG _{POD₅}	172	389	561	19
10^7	2314	DICCG ₁₀	170	808	978	42
10^7	2314	DICCG _{POD₁₀}	170	603	773	33
10^7	2314	DICCG _{POD₅}	170	372	542	23

Table 23: Comparison between the ICCG and DICCG methods of the average number of linear iterations for the second NR iteration for various contrast between permeability layers.

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

Table 24: Fluids properties.

Permeability contrast 10^1 (27×27 cells)

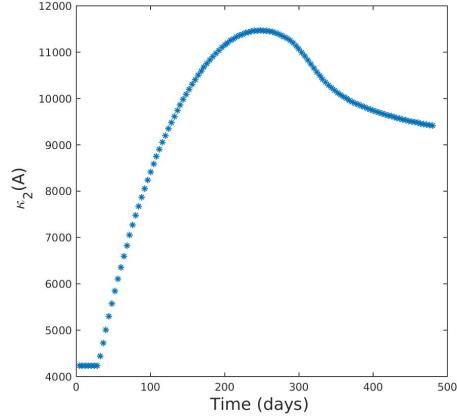


Figure 49: Eigs A

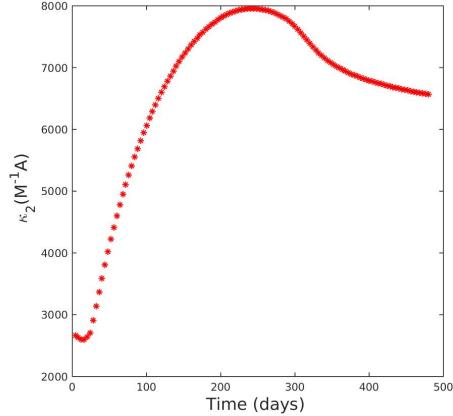


Figure 50: Eigs $M^{-1}(A)$

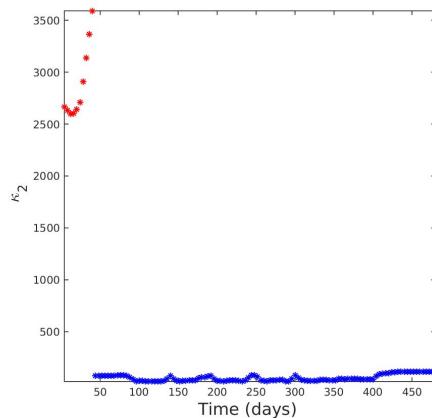


Figure 51: Eigs $PM^{-1}(A)$

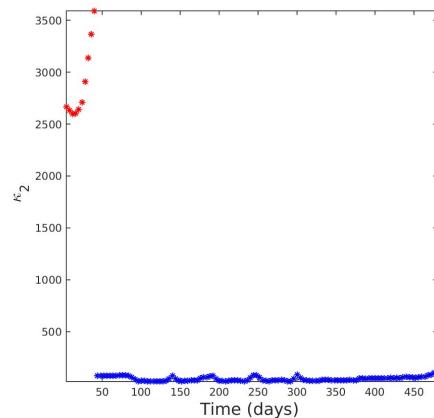


Figure 52: Eigs $PM^{-1}(A)$ POD

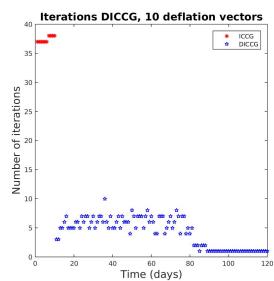


Figure 53: Iter DICCG

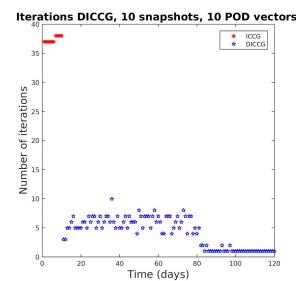


Figure 54: Iter $DICCG_{POD10}$

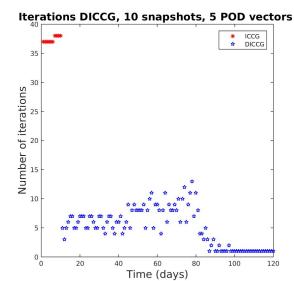


Figure 55: Iter $DICCG_{POD5}$

Permeability contrast 10^7 (27 x 27 cells)

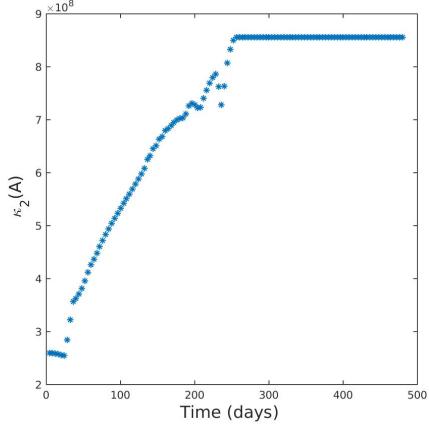


Figure 56: Eigs A

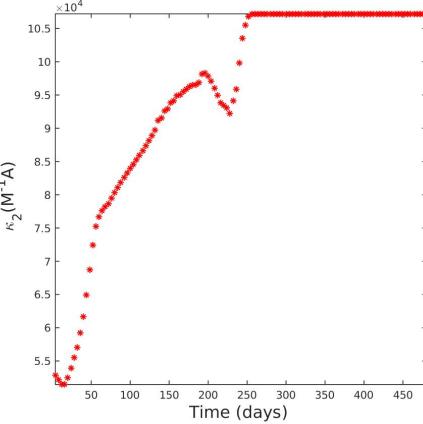


Figure 57: Eigs $M^{-1}(A)$

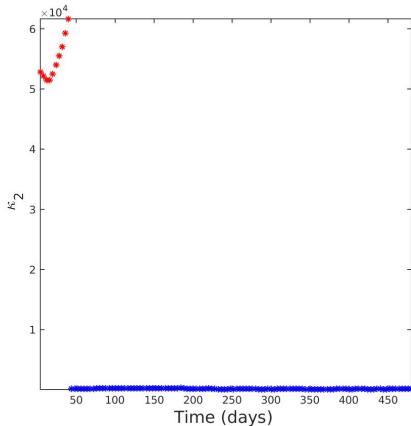


Figure 58: Eigs $PM^{-1}(A)$

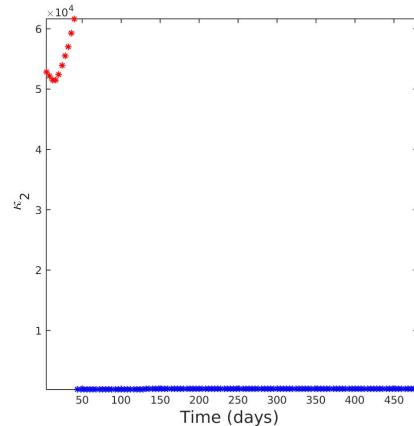


Figure 59: Eigs $PM^{-1}(A)$ POD

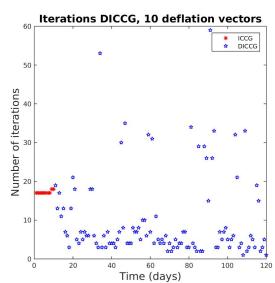


Figure 60: Iter DICCG

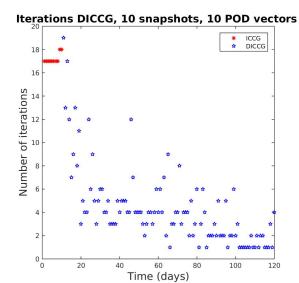


Figure 61: Iter DICCG_{POD10}

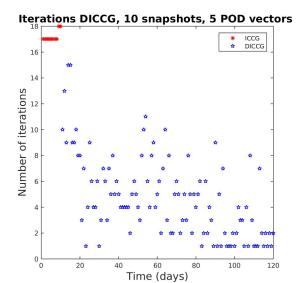


Figure 62: Iter DICCG_{POD5}

Permeability contrast 10^1 , capillary pressure included (27 x 27 cells)

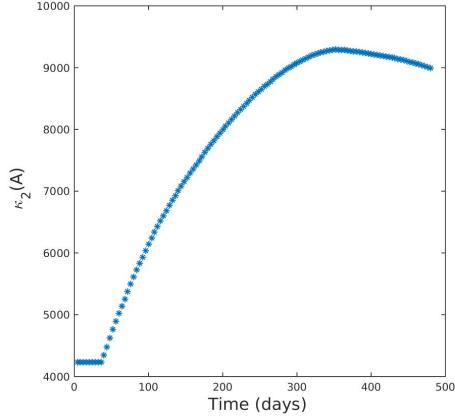


Figure 63: Eigs A

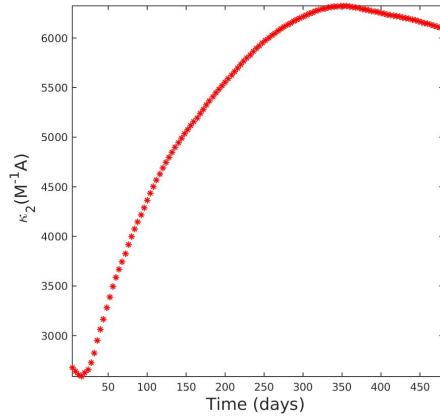


Figure 64: Eigs $M^{-1}(A)$

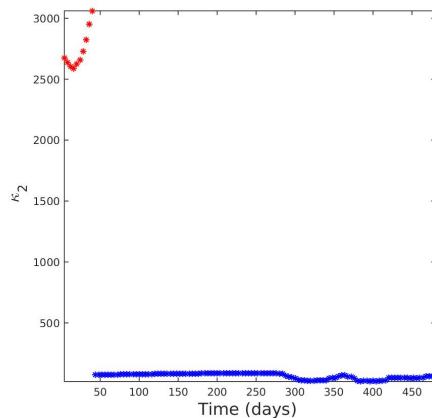


Figure 65: Eigs $PM^{-1}(A)$

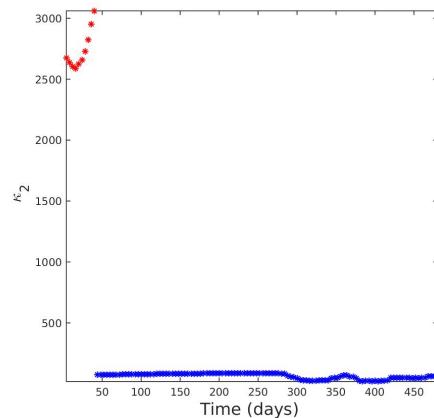


Figure 66: Eigs $PM^{-1}(A)$ POD

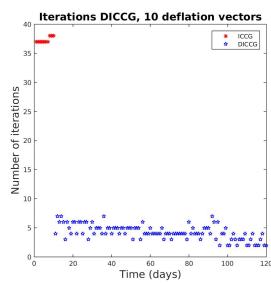


Figure 67: Iter DICCG

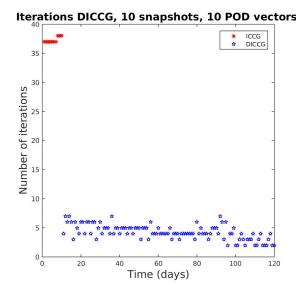


Figure 68: Iter $DICCG_{POD10}$

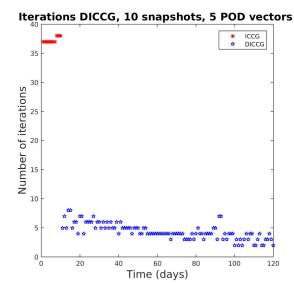


Figure 69: Iter $DICCG_{POD5}$

Permeability contrast 10^7 (27 x 27 cells)

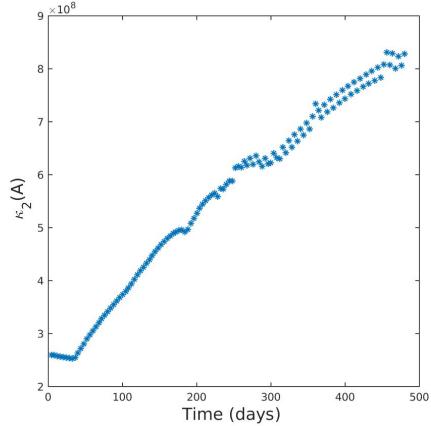


Figure 70: Eigs A

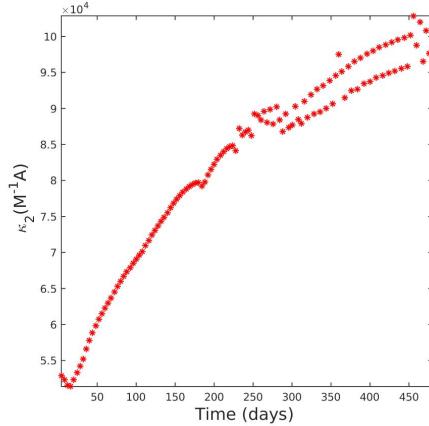


Figure 71: Eigs $M^{-1}(A)$

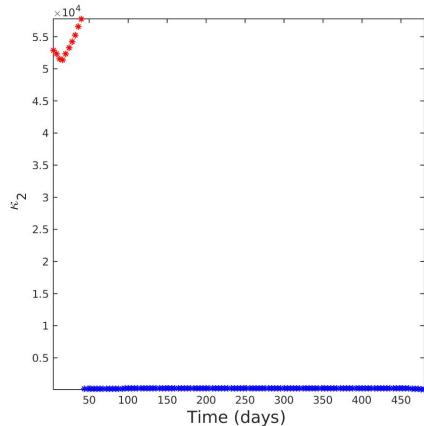


Figure 72: Eigs $PM^{-1}(A)$

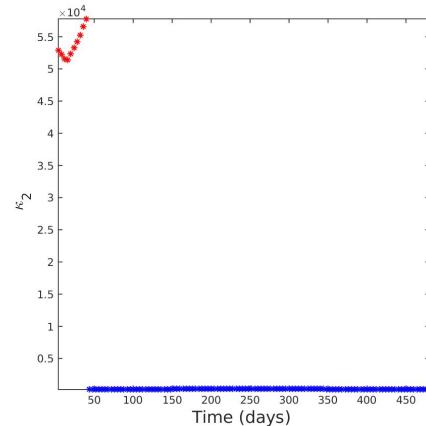


Figure 73: Eigs $PM^{-1}(A)$ POD

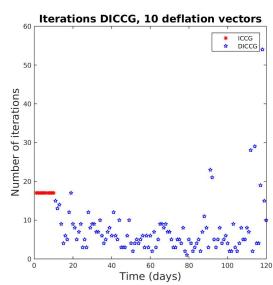


Figure 74: Iter DICCG

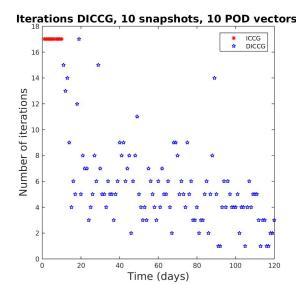


Figure 75: Iter DICCG_{POD10}

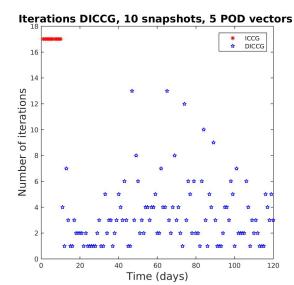


Figure 76: Iter DICCG_{POD5}

Permeability contrast 10^1 ($27 \times 27 \times 3$ cells)

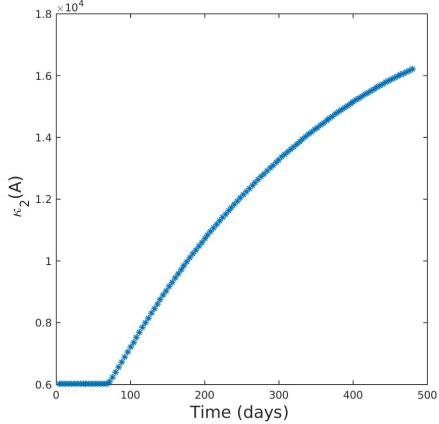


Figure 77: Eigs A

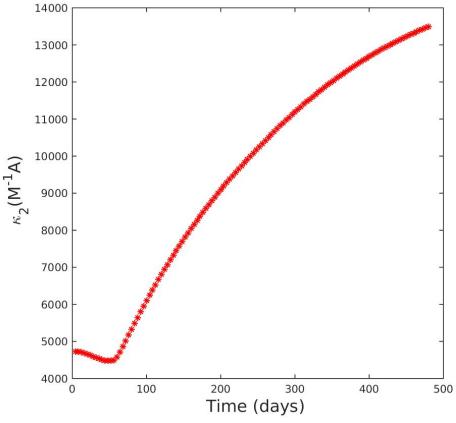


Figure 78: Eigs $M^{-1}(A)$

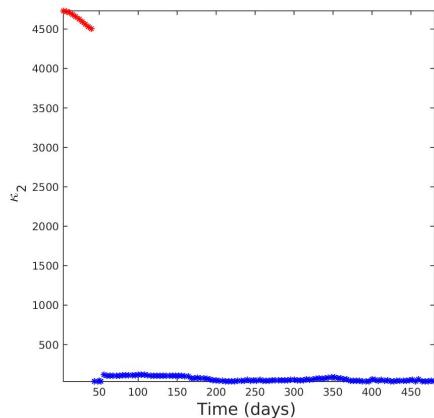


Figure 79: Eigs $PM^{-1}(A)$

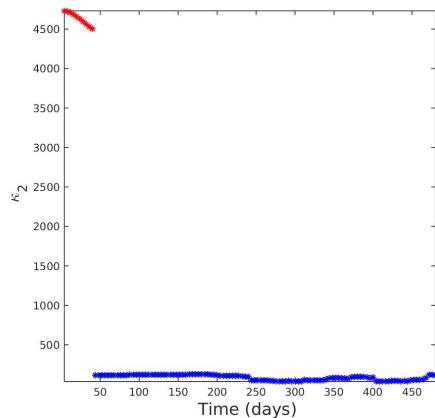


Figure 80: Eigs $PM^{-1}(A)$ POD

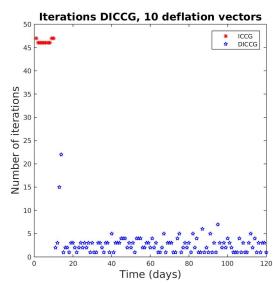


Figure 81: Iter DICCG

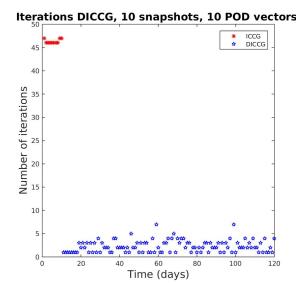


Figure 82: Iter DICCG_{POD10}

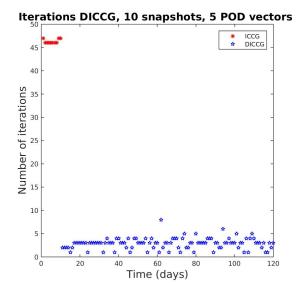


Figure 83: Iter DICCG_{POD5}

Permeability contrast 10^7 ($27 \times 27 \times 3$ cells)

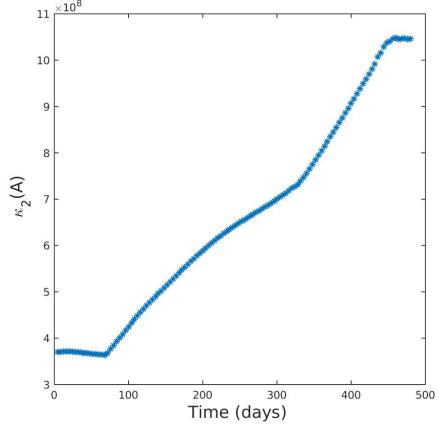


Figure 84: Eigs A

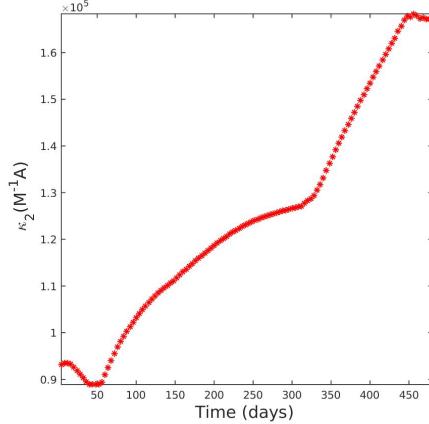


Figure 85: Eigs $M^{-1}(A)$

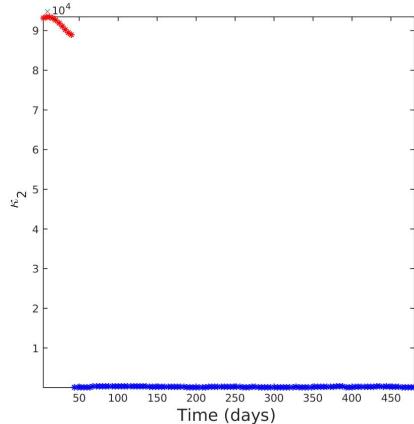


Figure 86: Eigs $PM^{-1}(A)$

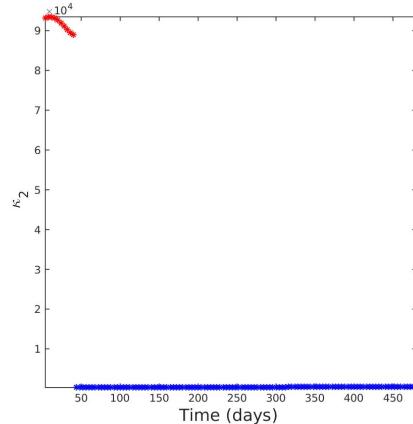


Figure 87: Eigs $PM^{-1}(A)$ POD

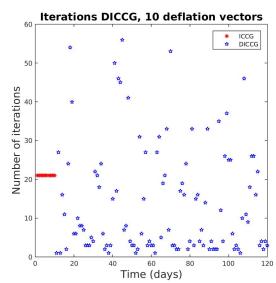


Figure 88: Iter DICCG

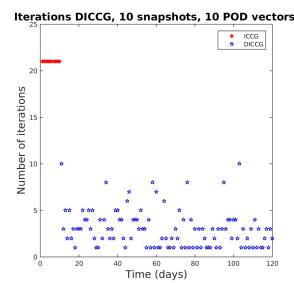


Figure 89: Iter DICCG_{POD10}

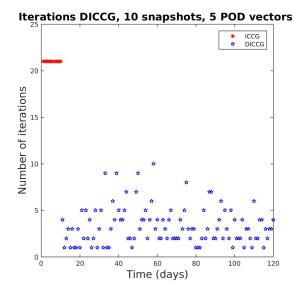


Figure 90: Iter DICCG_{POD5}

SPE 10 (60 x 220 x 10)

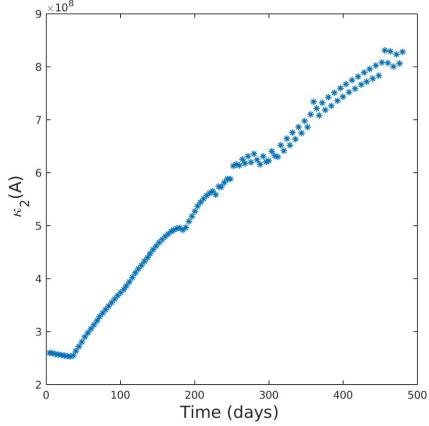


Figure 91: Eigs A

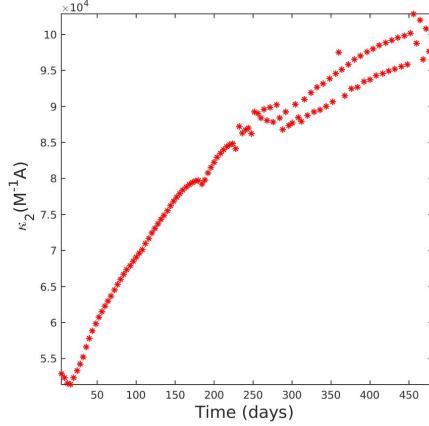


Figure 92: Eigs $M^{-1}(A)$

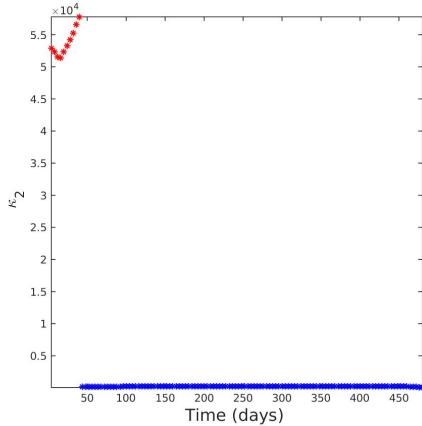


Figure 93: Eigs $PM^{-1}(A)$

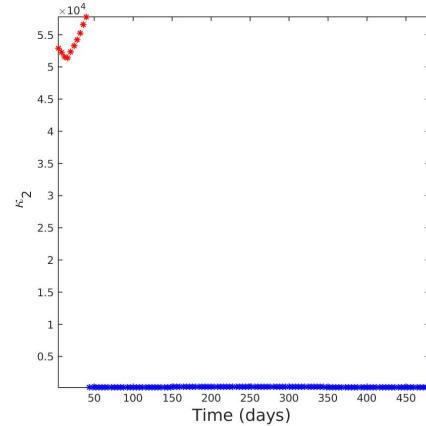


Figure 94: Eigs $PM^{-1}(A)$ POD

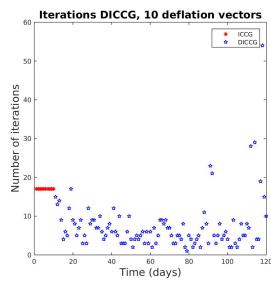


Figure 95: Iter DICCG

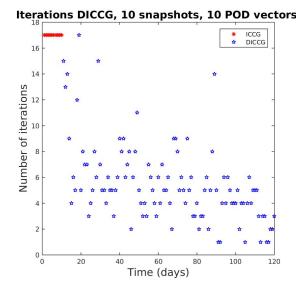


Figure 96: Iter $DICCG_{POD10}$

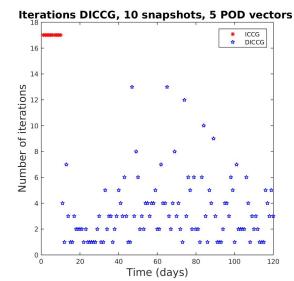


Figure 97: Iter $DICCG_{POD5}$

Conclusions

Acknowledgements

We like to thank the 'Consejo Nacional de Ciencia y Tecnología (CONACYT)', the 'Secretaría de Energía (SENER)' and the Mexican Institute of Petroleum (IMP) which, through the programs: 'Formación de Recursos Humanos Especializados para el Sector Hidrocarburos (CONACYT-SENER Hidrocarburos)' and 'Programa de Captación de Talento, Reclutamiento, Evaluación y Selección de Recursos Humanos (PCTRES)', have sponsored this work.

References

A List of notation

Symbol	Quantity	Unit
ϕ	Rock porosity	
\mathbf{K}	Rock permeability	<i>Darcy</i> (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 25: 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 [? , 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 it 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.

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

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:

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

C Singular Value Decomposition for POD

If we perform SVD in \mathbf{X} , we have

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T, \quad \mathbf{U} \in \mathbb{R}^{n \times n}, \quad \Sigma \in \mathbb{R}^{n \times m}, \quad \mathbf{V} \in \mathbb{R}^{m \times m}.$$

Then we have

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

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$$

$$\mathbf{U} = \mathbf{X}\mathbf{V}\Sigma^{-1}$$

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

If we compute Λ^T , we can compute \mathbf{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 [?]:

- 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:

$$\mathbf{x} = \mathbf{x} - \mathbf{P}^T\mathbf{x} + \mathbf{P}^T\mathbf{x} = (\mathbf{I} - \mathbf{P}^T)\mathbf{x} + \mathbf{P}^T\mathbf{x}. \quad (29)$$

Multiplying expression (29) by \mathbf{A} , using the properties of the deflation matrices, we have:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{A}(\mathbf{I} - \mathbf{P}^T)\mathbf{x} + \mathbf{AP}^T\mathbf{x}, && \text{Property :} \\ \mathbf{Ax} &= \mathbf{AQb} + \mathbf{AP}^T\mathbf{x}, && c) \\ \mathbf{b} &= \mathbf{AQb} + \mathbf{PAx}, && b), \end{aligned}$$

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

$$\begin{aligned} \mathbf{PAQb} + \mathbf{P}^2\mathbf{Ax} &= \mathbf{Pb}, \\ \mathbf{PAx} &= \mathbf{Pb}, \end{aligned}$$

where $\mathbf{PAx} = \mathbf{Pb}$ is the deflated system. Since \mathbf{PA} is singular, the solution of Equation (30) can contain components of the null space of \mathbf{PA} , ($\mathcal{N}(\mathbf{PA})$). A solution of this system, called the deflated solution, is denoted by $\hat{\mathbf{x}}$. Then, the linear system to solve is:

$$\mathbf{PA}\hat{\mathbf{x}} = \mathbf{Pb}. \quad (30)$$

As the solution of Equation (30) can contain components of $\mathcal{N}(\mathbf{PA})$, $\hat{\mathbf{x}}$ can be decomposed as:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{y}, \quad (31)$$

with $\mathbf{y} \in \mathcal{R}(\mathbf{Z}) \subset \mathcal{N}(\mathbf{PA})$, and \mathbf{x} the solution of Equation (??).

Note: If $\mathbf{y} \in \mathcal{R}(\mathbf{Z})$, then

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

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

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:

$$\mathbf{P}^T\mathbf{y} = \mathbf{0}. \quad (32)$$

Multiplying Equation (31) by \mathbf{P}^T we obtain:

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

substituting Equation (32) we arrive to:

$$\mathbf{P}^T\hat{\mathbf{x}} = \mathbf{P}^T\mathbf{x}. \quad (33)$$

Substitution of Equation (33) and property c) in Equation (29) leads to:

$$\mathbf{x} = \mathbf{Q}\mathbf{b} + \mathbf{P}^T\hat{\mathbf{x}}, \quad (34)$$

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

E Operation counts

The number of operations necessary to perform the deflation procedure is computed in this section for full matrices and sparse matrices.

First, we compute the number of operations between vectors and matrices necessary for ICCG method (see Table 26) and DICCG method (see Table 28).

With the numbers previously computed, we compute the number of operations necessary to perform the ICCG (see Table 27) and DICCG methods (see Table 29). In Table 30, we compute the number of operations necessary to perform the ICCG and DICCG methods for different sparsity of the matrix (m) and a diverse number of deflation vectors (p).

Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$.

Operation	Number of Operations	
	Full matrix	Sparse matrix (m non zero entries)
$\mathbf{x}^T \mathbf{y}$	$n(*) + n - 1(+) = 2n - 1$	$2n - 1$
$\mathbf{x}(+/-)\mathbf{y}$	n	n
$\alpha \mathbf{x}$	n	n
\mathbf{Ax}	$(n(*) + n - 1(+))n \text{ (r)} = 2n^2 - n$	$(m(*) + m - 1(+))n \text{ (r)} = 2mn - n$
\mathbf{AB}	$[(n(*) + n - 1(+))n \text{ (r)}]n \text{ (c)} = 2n^3 - n^2$	$[(m(*) + m - 1(+))n \text{ (r)}]m \text{ (c)} = 2m^2n - nm$
$\mathbf{A} \in \mathbb{R}^{m \times n} \mathbf{B} \in \mathbb{R}^{n \times p}$		
\mathbf{AB}	$mp(2n - 1)$	
$\mathbf{A} = \mathbf{LL}^T$	$1/3n^3$	
$\mathbf{Lx} = \mathbf{y}$	n^2	nm
$\mathbf{L}^T \mathbf{x} = \mathbf{y}$	n^2	nm

Table 26: Number of operations between matrices and vectors.

Algorithm 1 ICCG method, solving $\mathbf{Ax} = \mathbf{b}$.	Operations	
	Full matrix	Sparse matrix
Split preconditioner		
Give an initial guess \mathbf{x}^0 .		
Compute $\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0$.	$2n^2$	$2mn$
Compute $\hat{\mathbf{r}}^0 = \mathbf{L}^{-1}\mathbf{r}^0$.	n^2	nm
Compute $\hat{\mathbf{p}}^0 = \mathbf{L}^{-T}\hat{\mathbf{r}}^0$.	n^2	nm
for $k = 0, \dots$, until convergence		
$\mathbf{w}^k = \mathbf{Ap}^k$	$2n^2 - n$	$2nm - n$
$\mathbf{ry}^k = (\hat{\mathbf{r}}^k, \mathbf{y}^k)$	$2n - 1$	$2n - 1$
$\alpha^k = \frac{\mathbf{ry}^k}{(\mathbf{p}^k, \hat{\mathbf{w}}^k)}$	$2n$	$2n$
$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k$	$2n$	$2n$
$\hat{\mathbf{r}}^{k+1} = \hat{\mathbf{r}}^k - \alpha^k \mathbf{L}^{-1} \hat{\mathbf{w}}^k$	$n^2 + 2n$	$nm + 2n$
$\beta^k = \frac{(\hat{\mathbf{r}}^{k+1}, \hat{\mathbf{y}}^{k+1})}{\mathbf{ry}^k}$	$2n$	$2n$
$\mathbf{p}^{k+1} = \mathbf{L}^{-T} \hat{\mathbf{r}}^{k+1} + \beta^k \mathbf{p}^k$	$n^2 + 2n$	$nm + 2n$
end		
Total each k	$4n^2 + 11n - 1$	$4nm + 11n - 1 \sim (4m + 11)n$

Table 27: Number of operations needed to perform the ICCG method.

Algorithm 2 Deflation, solving $\mathbf{Ax} = \mathbf{b}$.	Operations	
	Full matrix	Sparse matrix
Let $\mathbf{Z} \in \mathbb{R}^{n \times p}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ $\mathbf{V} = \mathbf{AZ} \quad \in \mathbb{R}^{n \times p}$ $\mathbf{Z}^T \mathbf{V}$ $\mathbf{E} = \mathbf{Z}^T \mathbf{A} \mathbf{Z}$ \mathbf{E}^{-1} $\mathbf{B} = \mathbf{A} \mathbf{Z} \mathbf{E}^{-1} = \mathbf{V} \mathbf{E}^{-1} \quad \in \mathbb{R}^{n \times p}$	$np(2n - 1)$ $np(2n - 1)$ $2np(2n - 1)$ p^3 $np(2p - 1)$	$np(2m - 1)$ $np(2m - 1)$ $2np(2m - 1)$ p^3 $np(2p - 1)$
$\mathbf{y} = \mathbf{Z}^T \mathbf{x}$ $\mathbf{z} = \mathbf{By} \quad \in \mathbb{R}^n$ $\mathbf{w} = \mathbf{E}^{-1} \mathbf{y}$ $\mathbf{Q} = \mathbf{Z} \mathbf{w}$ $\mathbf{Qx} = \mathbf{ZE}^{-1} \mathbf{Z}^T \mathbf{x}$ $\mathbf{QE}^{-1} \mathbf{Z}^T \mathbf{x}$ (computing \mathbf{E} and \mathbf{E}^{-1})	$2np - p$ $n(2p - 1)$ $2p^2 - p$ $2pn - n$ $(4p - 1)n + p^2 - 2p$ $(4np + 2p - 1)n - 2p + p^3$	$2np - p$ $n(2p - 1)$ $2p^2 - p$ $2pn - n$ $(4p - 1)n + p^2 - 2p$ $(4mp + 2p - 1)n - 2p + p^3$
\mathbf{Vw} $\mathbf{AQx} = \mathbf{AZE}^{-1} \mathbf{Z}^T \mathbf{x} =$ $= [\mathbf{AZE}^{-1}] [\mathbf{Z}^T \mathbf{x}] = [\mathbf{B}] [\mathbf{Z}^T \mathbf{x}]$ (without computing \mathbf{B}) $\mathbf{Px} = (\mathbf{I} - \mathbf{AQ}) \mathbf{x} = \mathbf{x} - \mathbf{B} [\mathbf{Z}^T \mathbf{x}]$ (without computing \mathbf{B}) $\mathbf{P}_E \mathbf{x} = (\mathbf{I} - \mathbf{AQ}) \mathbf{x}$	$(2p - 1)n$ $[2np - p] + [n(2p - 1)]$ $= n(4p - 1) - p$ $4np - p$ $(2n + 4np + 2p - 1)n -$ $- 2p + p^3$	$(2p - 1)n$ $[2np - p] + [n(2p - 1)]$ $= n(4p - 1) - p$ $4np - p$ $(2m + 4mp + 2p - 1)n -$ $- 2p + p^3$

Table 28: Number of operations needed to compute some matrices and vectors necessary to perform the DICCG method.

Algorithm 3 DICCG method, solving $\mathbf{Ax} = \mathbf{b}$.	Operations	
	Full matrix	Sparse matrix
Split preconditioner		
Give an initial guess \mathbf{x}^0 .		
Compute:		
$\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0$	$2n^2$	$2mn$
$\mathbf{V} = \mathbf{AZ}$	$np(2n - 1)$	$np(2m - 1)$
$\hat{\mathbf{r}}^0 = \mathbf{Pr}^0$	$4np - p$	$4np - p$
$\mathbf{y}^0 = \mathbf{L}^{-1}\hat{\mathbf{r}}^0$	n^2	nm
$\mathbf{p}^0 = \mathbf{L}^{-T}\mathbf{y}^0$	n^2	nm
for $k = 0, \dots$, until convergence		
$\mathbf{w}^k = \mathbf{Ap}^k$	$2n^2 - n$	$2nm - n$
$\hat{\mathbf{w}}^k = \mathbf{Pw}^k$	$4np - p$	$4np - p$
$\mathbf{ry}^k = (\hat{\mathbf{r}}^k, \mathbf{y}^k)$	$2n - 1$	$2n - 1$
$\alpha^k = \frac{\mathbf{ry}^k}{(\mathbf{p}^k, \hat{\mathbf{w}}^k)}$	$2n$	$2n$
$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k$	$2n$	$2n$
$\hat{\mathbf{r}}^{k+1} = \hat{\mathbf{r}}^k - \alpha^k \mathbf{L}^{-1} \hat{\mathbf{w}}^k$	$n^2 + 2n$	$nm + 2n$
$\beta^k = \frac{(\hat{\mathbf{r}}^{k+1}, \hat{\mathbf{y}}^{k+1})}{\mathbf{ry}^k}$	$2n$	$2n$
$\mathbf{p}^{k+1} = \mathbf{L}^{-T} \mathbf{r}^{k+1} + \beta^k \mathbf{p}^k$	$n^2 + 2n$	$nm + 2n$
end		
Total each k	$4n^2 + 4pn + 11n - p - 1$	$4nm + 4pn + 11n - p - 1$ $\sim (4m + 4p + 11)n$

Table 29: Number of operations needed to perform the DICCG method.

			m	p=10	p=4
m=3	ICCG	$(4m+11)n$	23n	23n	23n
	DICCG	$(4m+11+4p)n$	$(23+4p)n$	63n	39n
	DICCG/ICCG			63/23=2.7	39/23=1.7
m=5	ICCG	$(4m+11)n$	31n	31n	31n
	DICCG	$(4m+11+4p)n$	$(31+4p)n$	71n	47n
	DICCG/ICCG			71/31=2.3	47/31=1.5
m=7	ICCG	$(4m+11)n$	39n	39n	39n
	DICCG	$(4m+11+4p)n$	$(39+4p)n$	79n	55n
	DICCG/ICCG			79/39=2	55/39=1.4

Table 30: Number of operations for the ICCG and DICCG methods for different sparsity of the matrices and different deflation vectors.